ADAPTIVE DIGITAL PROCESSING OF MULTIDIMENSIONAL SIGNALS WITH APPLICATIONS

Victor F. Kravchenko
Hector M. Perez-Meana
Volodymyr I. Ponomaryov

ADAPTIVE DIGITAL PROCESSING OF MULTIDIMENSIONAL SIGNALS WITH APPLICATIONS

Victor F. Kravchenko
Hector M. Perez-Meana
Volodymyr I. Ponomaryov

ADAPTIVE DIGITAL PROCESSING OF MULTIDIMENSIONAL SIGNALS WITH APPLICATIONS

MOSCOW
FIZMATLIT ®
2009

UDC 538.56:518.5:621.37+517.95:621.391.24:531
LBC 22.193
K 78

Kravchenko, V.F., Hector, M.P., and Ponomaryov, V.I. **Adaptive digital processing of multidimensional signals with applications.** — M.: FIZMATLIT, 2009. — 360 p. — ISBN 978-5-9221-1170-6.

In this monograph, the novel promising trends in adaptive digital processing of multidimensional 1D–3D signals with different applications to radio physics, radio engineering, and medicine are considered. The monograph consists of three parts. The first part (chapters 1-4) is devoted to the atomic functions (AF) and their applications, such as the novel wavelet systems (WA). This part includes the definition of the atomic functions, their properties, possible applications in signal and image processing, and the construction of novel wavelets based on the AF. The synthesis of novel weighting functions (windows) based on the AF and applications of the novel windows are discussed in the next chapters. In chapter 4, the basic principles of the wavelet analysis are considered in detail. Here, the Kotelnikov–Shannon and the Meyer wavelets as well as the wavelets based on the atomic functions are discussed. The second part of the book (chapters 5–9) is devoted to the multidimensional signal enhancement. Models of the image–and–noise and objective-and-subjective criteria are discussed in the fifth chapter. Chapter 6 introduces different types of statistical estimators (M, R, L, and RM) and their properties. Chapter 7 gives a review of the linear and nonlinear filtering techniques. Some commonly used models of multichannel (color) images are presented there. A novel approach of the vectorial order statistics to multichannel and video processing is presented in chapter 8. The vector median ordering and filtering, the adaptive multichannel non-filtering, the Vector Directional filter with a double window, etc. are explained. Elements of fuzzy logics theory and novel filtering techniques, such as 3D ultrasound, 3D vector, and fuzzy 3D vectorial filters, are discussed there. Chapter 9 exposes different implementations of processing techniques on the DSP and FPGA platforms. Some important problems are resolved: applications of the AF and wavelets based on the AF (WA) for compression-windowing in radar systems, compression algorithms for medical applications, and neural-network classification procedures in the mammography analysis. The analysis of the transversal FIR filter structure along with some of its most widely used adaptive algorithms is presented in the tenth chapter. In chapter 11, the fast Fourier transform is used for performing the convolution and correlation required in applications reducing the computational complexity. The adaptive infinite impulse response can provide the computational complexity with a much smaller number of filter coefficients. Some problems such as slow convergence, possible filter instability, and error function with multiple local minima, are discussed in chapter 12. The echo canceling procedures are described in chapter 13. The inter symbol interference reduction applying efficient equalizer algorithms are discussed in the final, fourteenth chapter of this book.

The monograph is recommended for scientists, engineers, students, and post-graduates specializing in radio physics, radio engineering, computational mathematics, computational physics, and medicine applications

Editor *V.F. Kravchenko*
Text design: *I.V. Shutov*
Cover design: *D.B. Belukha*

Printed in Russia

ISBN 978-5-9221-1170-6

ISBN 978-5-9221-1170-6

ООО Издательская фирма

ФИЗИКО-МАТЕМАТИЧЕСКАЯ ЛИТЕРАТУРА
(***ФИЗМАТЛИТ***)

113093,Москва, ул. Б. Серпуховская,д.8/7,стр.2 Телефон: (495) 334 7620
e-mail: fizmat@maik.ru Тел./Факс: (495) 334 7421

Dear authors:
Kravchenko, V.F., Perez-Meana, H.M., Ponomaryov, V.I.

Editorial **"FizMatLit" (Moscow, Russian Federation)** permits to you the freely distribution of the monograph "**Kravchenko, V.F., Perez-Meana, H.M., Ponomaryov, V.I. "Adaptive Digital Processing of Multidimentional Signals with Applications Moscow, Fizmatlit, 2009 360 p., ISBN 978-5-9221-1170-6."** that has been edited in accordance to agreement between the authors mentioned and FizMatLit in electronic variant as in paper one for different purposes such as, education, scientific and at the international conferences with citation of this book when it is being used.

General Director **M.N. Andreeva**

CONTENTS

Preface

The first part of this book (Chapters 1–4) is devoted to the atomic functions (AF) and their applications, such as novel wavelet (WA) systems. The theory of atomic functions goes back to 1971, when the function up (x) was first constructed and studies. Subsequently, the AF theory was considered in detail in different works initiated by the book by V. L. Rvachev and V. A. Rvachev «Non-classical Methods of Approximation Theory in Boundary-Value Problems», Kiev: Naukova Dumka, 1979. In the last years, several books concerning the AF theory have been published: V.F.Kravchenko «Lectures on the Theory of Atomic Functions and Their Some Applications», Moscow, Radiotekhnika, 2003; V. F. Kravchenko and M. A. Basarab «Boolean Algebra and Approximation Methods in Boundary Value Problems of Electrodynamics», Moscow, Fizmatlit, 2004; V. F. Kravchenko and V. L. Rvachev «Logic Algebra, Atomic Functions and Wavelets in Physical Applications», Moscow, Fizmatlit, 2006; «Digital Signal and Image Processing in Radio Physical Applications», Ed. by V. F. Kravchenko, Moscow, Fizmatlit, 2007.

The first part of this book presents an introduction to the atomic functions, their properties, possible applications in signal and image processing, and the design of novel wavelets based on the AF. The synthesis of novel weighting functions (windows) based on the AF and applications of the novel windows in the digital radar, electroencephalography, SAR, etc., are discussed in next chapters of the first part. In chapter 4, the basic principles of the wavelet analysis are considered in detail and different wavelets, such as the Kotelnikov–Shannon and Meyer wavelets, as well as the wavelets based on the atomic functions (WA), are discussed.

Chapters 5–9 are devoted to the multidimensional signal enhancement. Models of image and noise (additive, speckle, and impulsive) are discussed in chapter 5. Also, the objective and subjective criteria of evaluating the quality of filtering are presented there. Chapter 6 introduces different types of statistical estimators: the maximum likelihood and the M, R, and L estimators along with their properties. The theory of novel RM estimators is discussed there too. Chapter 7 gives a review of the linear and nonlinear filtering techniques. It offers explanation of the trimmed mean filters, the KNN, Lλ, LMS-L filters, the weighted median and order statistics filters, the vector median filter, the family of data-dependent filters, and different variants of the proposed RM filtering technique. Some commonly used models (RGB, YIQ, HIS, HSV, L*u*v* and L*a*b*) of multichannel (color) images are exposed there too. Finally, the applications of wavelet functions in multidimensional signal processing are discussed in connection with the theory presented in chapter 4. A novel approach of the vectorial order statistics to multichannel and video processing is presented in chapter 8. The vector median ordering and filtering are explained, and different classical and novel filtering techniques, such as the adaptive multichannel non-filtering, vector directional filter with double window, etc., are discussed. Also, the fuzzy logics definitions and properties, as well as the fuzzy generalization of classical filters are presented. Novel filtering techniques, such as 3D ultrasound, 3D vector, and fuzzy 3D vectorial filters, are studied and their properties are discussed.

Finally, chapter 9 exposes different implementations of multidimensional signal processing by the proposed algorithms on the DSP and FPGA platforms permitting

real-time filtering. Some important problems are resolved and presented there: applications of the AF and wavelets based on the AF (WA) for compression-windowing in radar systems, compression algorithms for medical applications, and neural-network based classification procedures in the mammography analysis.

The adaptive filters have been a subject of active research during last several decades due to their widespread use for solving different practical problems in communications, medicine, acoustics, security, etc. The transversal filter is the most widely used adaptive filter due to its unconditional stability. In order to deal with long impulse responses of such a filter, infinite impulse response (IIR) adaptive filter structures have been developed. In the third part of the book (chapters 10–14), a review of transversal, frequency domain, and infinite impulse response adaptive filter structures along with some successful applications, such as adaptive equalizers and echo canceling, is given. The analysis of the transversal FIR filter structure along with some of most widely used adaptive algorithms is presented in the chapter 10. In applications that require impulse responses with several hundreds or even thousands of FIR taps, their computational complexity becomes too high. In chapter 11, the fast Fourier transform is used for performing the convolution and correlation required in applications reducing the computational complexity.

The adaptive IIR can provide the computational complexity with a much smaller number of filter coefficients. There are a number of problems associated with adaptive IIR filters: slow convergence, possible filter instability, and error function with multiple local minima. These problems are discussed in chapter 12.

A fundamental problem in communication systems consisting of unidirectional and bidirectional communications links is the echo signal, which can be reduced using echo canceling. This problem is described in chapter 13.

The intersymbol interference reduction in most digital communication systems can be achieved by means of the efficient equalizer algorithms proposed and discussed in the final, fourteenth chapter of this book.

The first part of the book (chapters 1–4) was written by V. F. Kravchenko; the second part (chapters 5–9), by V. I. Ponomaryov; and the third part (chapters 10–14) by H. M. Perez-Meana.

Moscow–Mexico,
May 2009.

V. F. Kravchenko,
H. M. Perez-Meana,
and V. I. Ponomaryov

Chapter 1

THE THEORY OF ATOMIC FUNCTIONS

1.1. Introduction to the Theory of Atomic Functions

By definition, atomic functions (AF) are compactly supported infinitely differentiable solutions of differential equations with a shifted argument [1–5], i.e.,

$$Lf(x) = \lambda \sum_{k=1}^{M} c(k) f\,(a\,x - b(k)), \quad |a| > 1, \tag{1.1}$$

where L is a linear differential operator with constant coefficients. If $a=1$ and $b(k)=0$ $(k = 1,\ M)$, equation (1.1) becomes an ordinary differential equation.

The simplest and most important AFs are generated by infinite-to-one convolutions of rectangular impulses. To investigate such convolutions, we use the Fourier transform of this impulse:

$$\varphi(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{jux} \frac{\sin(u/2)}{u/2} du. \tag{1.2}$$

The N-to-one convolution of $(N+1)$ identical rectangle impulses $\varphi(x)$ gives us the compactly supported spline $\theta_N(x)$ which, analogously to (1.2), can be written as

$$\theta_N(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{jux} \left(\frac{\sin(u/2)}{u/2} \right)^{N+1} du. \tag{1.3}$$

Let us consider the convolution of impulses of variable length, $\varphi_n(x)$:

$$\varphi_n(x) = \begin{cases} 2^{n-1}, & |x| \leqslant 2^{n-1} \\ 0, & |x| > 2^{n-1} \end{cases}$$

Such convolution can be repeated infinitely, and the sum of the lengths of the convolved impulses forms a geometric progression, so that $\sum\limits_{n=1}^{\infty} 2^{-n+1} = 2$. Thus, the result of this operation is a new compactly supported function defined on the interval $[-1, 1]$. It can easily be shown that it satisfies equation (1.1) in the simplest form: $f'(x) = 2 \cdot f(2x + 1) - 2 \cdot f(2x - 1)$, where $f(0) = 1$, $\operatorname{supp} f(x) = [-1, 1]$. Its solution was denoted by $\operatorname{up}(x)$ («splash») (see [6–9]).

Analogously to (1.2) and (1.3), the function $\operatorname{up}(x)$ has the following representation in terms of the Fourier transform:

$$\operatorname{up}(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{jux} \prod_{k=1}^{\infty} \frac{\sin\left(u \cdot 2^{-k}\right)}{u \cdot 2^{-k}} du. \tag{1.4}$$

The most useful property of the AFs is the possibility to represent any polynomial by means of their translations. Also, the Fourier transforms for the AFs are known explicitly. Besides, simple expressions for moments and derivatives of the AFs take place [8].

1.2. The «Mother» Atomic Function up (x) and Its Main Properties

From the aforementioned definitions one can obtain immediately the main properties of the function up (x).

1. Function up (x) is even, i.e.,

$$\text{up}\,(x) = \text{up}\,(-x), \quad \text{up}\,(x) = 1 - \text{up}\,(1-x). \tag{1.5}$$

2. Its maximum value is up $(0) = 1$, and $\int\limits_{-1}^{1} \text{up}\,(x)dx = 1$.

Plots of the function up (x) and its Fourier spectrum are shown on Figure 1.1. The spectrum of up (x) is an even real-valued function of exponential type, rapidly damping and having zeros at points divisible by 2π. The first sidelobe level is equal to -23.5 dB.

3. The first derivative of the function up (x) has the simple expression $\text{up}'(x) = 2 \cdot \text{up}\,(2x+1) - 2 \cdot \text{up}\,(2x-1)$.

Analogously, we obtain expressions for higher-order derivatives. The nth order derivative is evaluated by the formula

4. $\text{up}^{(n)}(x) = 2^{C_n^2+1} \sum\limits_{k=1}^{2^n} \delta_k \,\text{up}\,(2^n x + 2^n + 1 - 2k)$, where δ_k are determined by the recurrent relations $\delta_1 = 1$, $\delta_{2k} = -\delta_k$, $\delta_{2k-1} = \delta_k$.

The function up (x) is a solution to the so-called «partition of unity» problem since its translations form the constant function equal to unity:

5.

$$\sum_{k=-\infty}^{+\infty} \text{up}\,(x-k) \equiv 1. \tag{1.6}$$

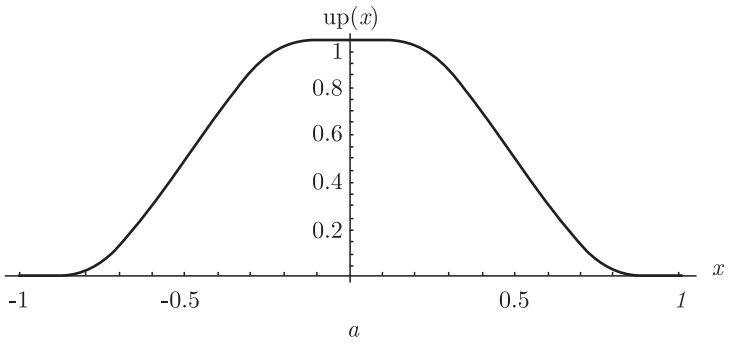
Translations with smaller steps yield polynomials of any degree, i.e.,

6.

$$\sum_{k=-\infty}^{+\infty} C(k)\,\text{up}\,\left(x - k \cdot 2^{-N}\right) \equiv x^N.$$

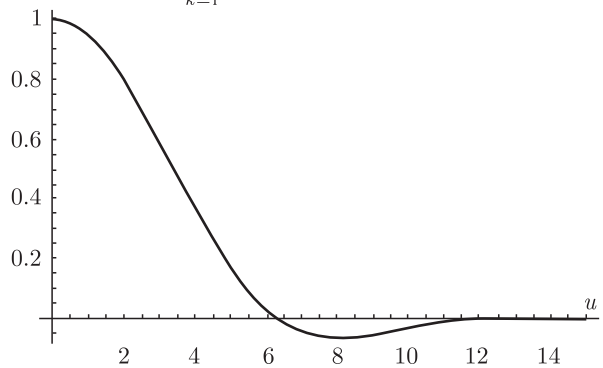
It is essential that, since up (x) is compactly supported, the previous sum contains only a finite number of nonzero terms for each value of x equal to $2^{N+1} = C(k)$. Let $\xi_1, \xi_2, \xi_3, \ldots, \xi_n, \ldots$ be a sequence of independent random variables uniformly distributed on the interval $[-1, 1]$. Then the value $\xi = \sum\limits_{k=1}^{\infty} \xi_k 2^{-k}$ has the probability density up (x).

7. For symmetric moments of up (x) we have $a_n = \int\limits_{-1}^{1} x^n \,\text{up}\,(x)dx$ and, due to evenness of the function, $a_0 = 1$ and $a_{2n+1} = 0$. Moments of even order are evaluated recurrently as $a_{2n} = \dfrac{(2n)!}{2^{2n}-1} \sum\limits_{k=1}^{n} \dfrac{a_{2n-2k}}{(2n-2k)!(2k+1)!}$.



a

$$\Im\{\mathrm{up}(x)\} = \prod_{k=1}^{x} \frac{\sin(u \cdot 2^{-k})}{u \cdot 2^{-k}}$$



b

Fig. 1.1. Atomic function up (x): (*a*) function; (*b*) spectrum.

The first four of them are as follows: $a_2 = \frac{1}{9}$, $a_4 = \frac{19}{3^3 5^2}$, $a_6 = \frac{583}{3^5 5 \cdot 7^2}$, and $a_8 = \frac{132809}{3^7 5^3 7 \cdot 17}$.

8. For nonsymmetric moments of up (x) we have $b_n = \int\limits_0^1 x^n \,\mathrm{up}\,(x)dx$, $b_{2n} = a_{2n}$, and $b_{2n+1} = \frac{1}{(n+1)2^{n+1}} \sum a_{2n-2k+2} C_{2n+2k}^{2k}$. For instance, $b_1 = \frac{5}{2^2 3^2}$, $b_3 = \frac{143}{2^3 3^3 5^2}$, $b_5 = \frac{1153}{2^6 3^6 7^2}$, and $b_7 = \frac{1616353}{2^4 3^7 5^4 7 \cdot 17}$.

9. The function up (x) possesses the following interesting and important property: its moments are related to its values at binary rational points as

$$\mathrm{up}\,(1 - 2^{-n}) = \frac{b_{n-1}}{(n-1)!\, 2^{n(n-1)/2}}. \tag{1.7}$$

This allows one to evaluate the function at these points by the formulas for the moments: $\mathrm{up}\,(-1/2) = \frac{1}{2}$, $\mathrm{up}\,(-3/4) = \frac{5}{2^3 3^2}$, $\mathrm{up}\,(-7/8) = \frac{1}{2^5 3^2}$, $\mathrm{up}\,(-15/16) = \frac{143}{2^{10} 3^4 5^2}$, and $\mathrm{up}\,(-31/32) = \frac{19}{2^{14} 3^4 5^2}$.

10. Property (8) gives the possibility to write the following specific series for effective evaluation of $\mathrm{up}\,(x)$ at an arbitrary point of the interval $[0, 1]$:

$$\mathrm{up}\,(x-1) = \sum_{n=1}^{\infty} (-1)^{S_n+1} a_n \sum_{k=0}^{n} A_{nk}\,(x - 0,\, a_1 \ldots a_n)^k, \tag{1.8}$$

where $S_n = \sum\limits_{i=1}^{n} a_i$, a_i are the digits of the binary representation of the argument, and $x = \sum\limits_{i=1}^{\infty} a_i 2^{-i}$, so that $(x - 0,\, a_1 \ldots a_n) = \sum\limits_{i=1}^{\infty} a_{i+n} 2^{-i}$. The coefficients of series (1.9) are defined by the values at binary rational points: $A_{nk} = \frac{2^{C_{k+1}^2} \cdot \mathrm{up}\,(1 - 2^{-(n-k)})}{k!}$.

Series (1.8) is rapidly convergent. To attain the accuracy of 2^{-64}, it is sufficient to take $n = 1, \ldots, 9$. If $n = 5$, the error will not exceed $2.06 \cdot 10^{-9}$.

The function $\mathrm{up}\,(x)$ is nonanalytic everywhere on its support: either the corresponding Taylor series has a zero radius of convergence or it converges to another function. That is why one cannot use ordinary power series to represent $\mathrm{up}\,(x)$. However, after extension with period 2π onto the whole real axis, the function $\mathrm{up}\,(x)$ has a rapidly convergent Fourier series expansion in even harmonics:

$$\mathrm{up}\,(x) = 0.5 + \sum_{k=1}^{\infty} Up\,(\pi k) \cos\,[\pi(2k-1)x].$$

11. To approximate functions of many (L) variables, the multidimensional function $\mathrm{up}\,(L, x) = \prod\limits_{i=1}^{L} \mathrm{up}\,(x_i)$ is used.

1.3. Functions $\mathrm{fup}_N(x)$ and Their Properties

The convolution of functions $\theta_n(x)$ and $\mathrm{up}\,(x)$ yields another important function defined on the interval $[-(N+2)/2, (N+2)/2]$ and denoted by $\mathrm{fup}_N(x)$:

$$\mathrm{fup}_N(x) = \theta_N(x) * \mathrm{up}\,(2x) = \theta_{N-1}(x) * \mathrm{up}\,(x). \tag{1.9}$$

Here, $\mathrm{fup}_0(x) \equiv \mathrm{up}\,(x)$.

The Fourier transform of $\mathrm{fup}_N(x)$ can be written as

$$\mathrm{fup}_N(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{jux} \left(\frac{\sin(u/2)}{u/2}\right)^N \prod_{k=1}^{\infty} \frac{\sin\,(u \cdot 2^{-k})}{u \cdot 2^{-k}} du. \tag{1.10}$$

All three considered functions, up (x), $\theta_N(x)$, and fup $_N(x)$, have common representation based on the identity $\sin(x)/x \equiv \prod_{k=1}^{\infty} \cos\left(2^{-k}x\right)$:

$$\varphi_{a,b}(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{jux} \prod_{k=1}^{\infty} \left(\cos\left(2^{-k}x\right)\right)^{ak+b} du. \tag{1.11}$$

Then, if $a = 0$ and $b = N + 1$, we have $\theta_N(x)$. If $a = 1$ and $b = -1$, we have up (x) and, if of $a = 1$ and $b = N$, (1.11) yields fup $_N(x)$.

As mentioned above, since the perfect splines are piecewise polynomials, they can be used in the problems of polynomial interpolation and the corresponding relations have the convolution form. However, one should take into account that the compactly supported spline $\theta_N(x)$ has at most N-1 continuous derivatives, whereas most practical problems require infinitely differentiable compactly supported functions. It is in such cases that the infinitely differentiable atomic functions fup $_N(x)$ are most useful.

Functions fup $_N(x)$ are the so-called *fractional components* of up (x) [6, 14–16, 19, 20]. This means that the function up (x) can be expanded into a finite interval convolution of functions fup $_N(x)$ for any N. This property makes formulas for the atomic functions to be flexible in numerical realization. In interpolation problems, the functions fup $_N(x)$ are of special interest, whereas the function up (x) is considered as a basic one providing infinite smoothness in interpolation.

Let us consider the main properties of functions fup $_N(x)$, necessary for their application.

1. The function fup $_N(x)$ is positive and even for all N, supp fup $_N(x) = [-(N+2)/2, (N+2)/2]$, and $\int_{-1}^{1}$ fup $_N(x)dx = 1$.

2. The derivative of fup $_N(x)$ is expressed via the function fup $_{N-1}(x)$:

$$\text{fup}'_N(x) = \text{fup}_{N-1}\left(x + \frac{1}{2}\right) - \text{fup}_{N-1}\left(x - \frac{1}{2}\right).$$

3. The functions fup $_N(x)$ and fup $_{N+1}(x)$ are recurrently related by the expression

$$\text{fup}_N(x) = 2^{-N} \sum_{k=0}^{N+1} C_{N+1}^k \, \text{fup}_{N+1}\,[2x - k + (N+2)/2].$$

This property means that each function fup $_N(x)$ is decomposed to the finite interval convolution of $(N+2)$ compressed functions fup $_{N+1}(x)$.

4. A recurrent use of property 3) yields the relation between up (x) and fup $_N(x)$:

$$\text{up}(x) = 2^{-C_N^2} \sum_{k_1=0}^{1} \sum_{k_2=0}^{2} \cdots \sum_{k_N=0}^{N} C_1^{k_1} C_2^{k_2} \ldots C_N^{k_N} \times \\ \times \text{fup}_N\left[2^N(x+1) - \sum_{i=1}^{N} \left(k_i 2^i - N\right)\right], \tag{1.11a}$$

where the number of expansion terms is equal to $2^{N+1} - (N+1)$. For example, up $(x) =$ $= \text{fup}_1\left(2x - \frac{1}{2}\right) + \text{fup}_1\left(2x + \frac{1}{2}\right)$.

5. From property 4) one can obtain the inverse expression for $\mathrm{fup}_N(x)$ via $\mathrm{up}(x)$. To do this, represent (1.11a) in the form

$$2^{C_N^2} up\left(\frac{x-(N+2)/2}{2^N}+1\right) = \\ = \frac{1}{2\pi}\int\limits_{-\infty}^{+\infty}\prod_{k=0}^{N-1}\left(1+e^{ju\,2^k}\right)^{N-k}\left(\frac{\sin(u/2)}{u/2}\right)^N\prod_{i=1}^{\infty}\frac{\sin\left(u\,2^{-i}\right)}{u\,2^{-i}}e^{ju}du,$$

from which, by a subsequent subtraction of functions $\mathrm{up}(x)$ shifted to the left, one can obtain the necessary relations for $\mathrm{fup}_N(x)$. For example, on the interval $[0,\ 3/2]$ we have

$$\mathrm{fup}_1(x) = \mathrm{up}\left(\frac{x+1/2}{4}\right) - \mathrm{up}\left(\frac{x+3/2}{4}\right). \tag{1.12}$$

Analogously, for the other cases we obtain

$$\begin{aligned} &N=2: && 2\cdot\mathrm{up}\left(\frac{x+2}{4}\right) = \mathrm{fup}_2(x) + 2\cdot\mathrm{fup}_2(x+1),\\ &[0,2]: && \mathrm{fup}_2(x) = 2\Big[\mathrm{up}\left(\frac{x+2}{4}\right) - 2\cdot\mathrm{up}\left(\frac{x+3}{4}\right)\Big]. \qquad (1.12a)\\ &N=3: && 8\cdot\mathrm{up}\left(\frac{x+11/2}{8}\right) =\\ &&& = \mathrm{fup}_3(x) + 3\cdot\mathrm{fup}_3(x+1) + 5\cdot\mathrm{fup}_3(x+2),\\ &[0,5/2]: && \mathrm{fup}_3(x) = 8\Big[\mathrm{up}\left(\frac{x+11/2}{8}\right) -\\ &&& -3\cdot\mathrm{up}\left(\frac{x+13/2}{8}\right) + 4\cdot\mathrm{up}\left(\frac{x+15/2}{8}\right)\Big]. \qquad (1.12b)\\ &N=4: && 64\cdot\mathrm{up}\left(\frac{x+13}{16}\right) =\\ &&& = \mathrm{fup}_4(x) + 4\cdot\mathrm{fup}_4(x+1) + 9\cdot\mathrm{fup}_4(x+2),\\ &[0,3]: && \mathrm{fup}_4(x) = 64\Big[\mathrm{up}\left(\frac{x+13}{16}\right) -\\ &&& -4\,\mathrm{up}\left(\frac{x+14}{16}\right) + 7\,\mathrm{up}\left(\frac{x+15}{16}\right)\Big]. \qquad (1.12c) \end{aligned}$$

These relations allow the use of the series (Property 4) for computing $\mathrm{fup}_N(x)$. For instance, one can find $\mathrm{fup}_1(-1) = 5/72$, $\mathrm{fup}_1(-1/2) = 1/2$, and $\mathrm{fup}_1(0) = 31/36$.

6. From property 5) it follows that, to within a constant factor, $\mathrm{fup}_N(x)$ coincides on the interval $[N/2,\ N/2{+}1]$ with a part of the shifted function $\mathrm{up}(x)\,\mathrm{fup}_N(x) = 2^{C_N^2}\,\mathrm{up}\left(\frac{x-(N+2)/2}{2^N}+1\right)$.

7. Analogously, on each jth interval $[j\text{-}(N{+}2)/2;\ j-N/2]$, for $j=1,\ldots,N+1$ the following relation takes place:

$$\mathrm{fup}_N(x) = 2^{C_N^2}(-1)^j C_{N+1}^j\,\mathrm{up}\left(\frac{x-j+(N+2)/2}{2^N}-1\right) + \\ + \sum_{k=0}^{N}\frac{\mathrm{fup}_N^{(k)}\left(j-(N+2)/2\right)}{k!}\cdot\left(x-j+(N+2)/2\right)^k,$$

where

$$\mathrm{fup}_N^{(k)}(j-(N+2)/2)=\frac{2^{-k-1}}{(n-k)!}\cdot\sum_{m=0}^{j-1}(-1)^m C_{N+1}^m\sum_{s=0}^{N-k}[2(j-m)-1]^s\,C_{n-k}^s a_{n-k-s}.$$

Here, at the points $x_j=-N/2+j$, equally spaced with the unit step from the left end of the support of function $\mathrm{fup}_N(x)$, we have

$$\mathrm{fup}_N(x_j)=2^{C_N^2}(-1)^j C_{N+1}^j\,\mathrm{up}\left(1-2^{-N}\right)+\sum_{k=0}^{N}\frac{\mathrm{fup}_N^{(k)}(j-(N+2)/2)}{k!}. \tag{1.13}$$

In particular, $\mathrm{fup}_N(-N/2)=2^{C_N^2}\,\mathrm{up}\left(1-2^{-N}\right)$.

8. Another way to evaluate functions $\mathrm{fup}_N(x)$ is to use rapidly convergent Fourier series after periodic expansion of the function onto the whole real axis with a period equal to the length of its support. For example, for the function $\mathrm{fup}_{N-1}(x)$, we have

$$\mathrm{fup}_{N-1}(x)=N^{-1}\left\{1+2\sum_{k=1}^{N/4-1}\left(\frac{\sin\frac{\pi k}{N}}{\frac{\pi k}{N}}\right)^N\prod_{i=1}^{\infty}\frac{\sin\frac{\pi k 2^{-i}}{N}}{\frac{\pi k 2^{-i}}{N}}\cos\frac{2\pi k x}{N}\right\}. \tag{1.14}$$

The number of terms in the sum must be restricted by the quarter order of the function. For a function of order N, the infinite product in (1.14) can be omitted, i.e., it can be replaced by a compactly supported spline without loss of accuracy.

Plots of the functions $\mathrm{fup}_N(x)$, the first and second derivatives of the function $\mathrm{fup}_2(x)$, and Fourier transforms of the functions $\mathrm{fup}_N(x)$ are shown in Figures 1.2–1.3.

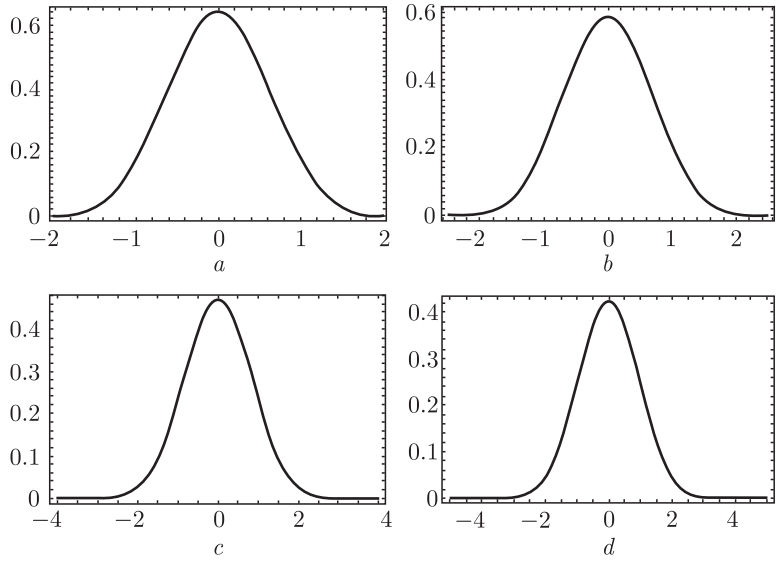


Fig. 1.2. Functions $\mathrm{fup}_N(x)$ for $N=2,3,6,8$.

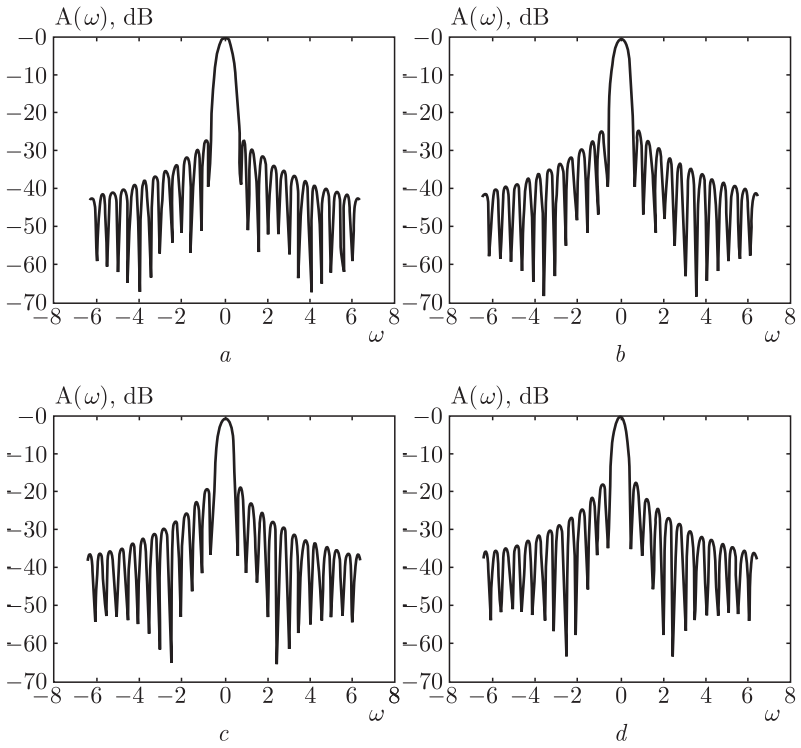


Fig. 1.3. Fourier transforms of $\operatorname{fup}_N(x)$ for $N = 2, 3, 6, 8$ in a logarithmic scale.

1.4. Atomic Functions $h_a(x)$ and Their Properties

Atomic functions $h_a(x)$ $(a > 1)$ are the compactly supported solutions of the functional-differential equation

$$y'(x) = \frac{a^2}{2}\left(y(ax+1) - y(ax-1)\right). \tag{1.15}$$

The function $h_a(x)$ is widely used for the synthesis of weighting windows in digital signal processing [13, 14]. The function $h_a(x)$ with $a = 2$ is designated by $\operatorname{up}(x)$. Let us itemize basic properties of the function $h_a(x)$ [3, 13]:

1) $h_a(x) = 0$ at $|x| > (a-1)^{-1}$.

2) $h_a(x) = a/2$ at $|x| \leqslant \dfrac{a-2}{a(a-1)}$, $a \geqslant 2$.

3) The Fourier transform of $h_a(x)$ is given by the formula

$$F_a(p) = \prod_{k=1}^{\infty} \operatorname{sinc}\left(p/a^k\right) \tag{1.16}$$

and vanishes at the points $a\pi n$, $n \neq 0$. Using formula (1.16), we can write the Fourier expansion of $h_a(x)$ on the interval $|x| \leqslant (a-1)^{-1}$:

$$h_a(x) = (a-1)\left(\frac{1}{2} + \sum_{k=1}^{\infty} F_a[(a-1)\pi k]\cos[(a-1)\pi k x]\right).$$

4) Expression (1.16) is the characteristic function of the random variable $\xi(a) = \sum_{j=1}^{\infty} a^{-j}\xi_j$, where $\{\xi_j\}$ is the sequence of independent random variables uniformly distributed over the segment [–1, 1]. The function h_a(x) is an infinite-fold convolution of the characteristic functions of intervals [–a–k, a–k] and represents the probability density of the random variable $\xi(a)$; therefore, $\int_{-\infty}^{\infty} h_a(x)dx = 1$. The lengths of the characteristic intervals form a geometric progression with the base $a - 1 < 1$.

5) Moments of function $h_a(x)$ and derivatives of expression (1.16) at zero points are related by the formula $\int_{-\infty}^{\infty} x^{2k} h_a(x)dx = (-1)^k F_a^{(2k)}(0)$.

Furthermore, $F_a^{(2k)}(0) = (2k)!c_{2k}(a)$, where the quantities $c_{2k}(a)$ are calculated from the simple recurrence formulas: $c_0(a) = 1$, $c_{2k}(a) = \dfrac{1}{a^{2k} - 1} \sum\limits_{j=0}^{k-1} \dfrac{(-1)^{k-j} c_{2j}(a)}{(2k - 2j + 1)!}$, $k = 1, 2, \ldots$

6) For $a > 2$, the function $h_a(x)$ is a polynomial on the full measure set and is a nonanalytic function on the remaining nowhere dense null set. (The latter means that the Taylor series either contains a finite number of terms and does not converge to $h_a(x)$ or has a zero radius of convergence). The functions $h_a(x)$ with $a > 2$ can be treated as splines of class C^{∞}.

7) Using (1.15), derivatives of the functions $h_a(x)$ can be recurrently expressed by the formula involving shifted and dilated functions $h_a(x)$:

$$h_a^{(n)}(x) = 2^{-n} a^{\frac{n(n+3)}{2}} \sum_{k=1}^{2^n} \delta_k h_a \left(a^n x + \sum_{j=1}^{n} a^{j-1} (-1)^{p_j(k-1)} \right),$$

where $\delta_1 = 1$, $\delta_{2k} = -\delta_k$, $\delta_{2k-1} = \delta_k$, $k = 1, 2, \ldots$, and $p_j(k)$ is the number standing in the jth position of the binary expansion of the number k, i.e., $p_j(k) = [k \cdot 2^j] \bmod 2$.

1.5. Interpolation of Signals by $h_a(x)$

It has been proposed to use the Fourier transforms of the functions $h_a(x)$ for the interpolation of signals [8]. In order to interpolate a function $f(x)$ at the points $2\pi n$, where n is an integer, the following series was proposed [6, 16]:

$$\widetilde{f}(x) = \sum_{k=-\infty}^{\infty} f(2\pi k) \prod_{n=1}^{\infty} \operatorname{sinc}\left[(x - 2\pi k)/2^n\right]. \tag{1.17}$$

However, the convergence conditions for this series were not investigated. Below, a more general result is proved.

Theorem 1. *The series*

$$\widetilde{f}(x) = \sum_{k=-\infty}^{\infty} f(k\Delta) F_a\left[\frac{a\pi}{\Delta}(x - k\Delta)\right], \tag{1.18}$$

where $a > 1$ and $\Delta > 0$, converges if the function $f(x)$ is absolutely integrable over the whole real axis, i.e., if $f(x) \in L_1[-\infty, \infty]$.

Proof. Evidently,

$$F_a\left[\frac{a\pi}{\Delta}(x - k\Delta)\right] = \int_{-\infty}^{\infty} h_a(z) \exp\left[iz\frac{a\pi}{\Delta}(x - k\Delta)\right] dz.$$

Then, $\widetilde{f}(x) = \sum\limits_{k=-\infty}^{\infty} f(k\Delta) \int\limits_{-\infty}^{\infty} \exp(ia\pi xz/\Delta) h_a(z) \exp(-ia\pi kz) dz$.

Let us expand the function $\exp(ia\pi xz/\Delta)$ into the Taylor series in powers of the variable z:

$$\exp(ia\pi xz/\Delta) = \sum_{m=0}^{\infty} \frac{(ia\pi x/\Delta)^m}{m!} z^m.$$

Then, the expression for $\widetilde{f}(x)$ takes the form

$$\widetilde{f}(x) = \sum_{k=-\infty}^{\infty} f(k\Delta) \sum_{m=0}^{\infty} \frac{(ia\pi x/\Delta)^m}{m!} \int\limits_{-\infty}^{\infty} z^m h_a(z) \exp(-ia\pi kz) dz.$$

Denote $b_{k,m} = \int_{-\infty}^{\infty} z^m h_a(z) \exp(-ia\pi kz) dz$. The following estimate is valid:

$$|b_{k,m}| \leqslant |\xi|^m h_a(0) \int\limits_{-1/(a-1)}^{1/(a-1)} |\exp(-ia\pi kz)| dz = \frac{2h_a(0)}{a-1} |\xi|^m = C|\xi|^m,$$

where $\xi \in [-(a-1)^{-1}, (a-1)^{-1}]$. Taking this fact into account, we obtain

$$|\widetilde{f}(x)| \leqslant \sum_{k=-\infty}^{\infty} |f(k\Delta)| \sum_{m=0}^{\infty} \frac{|ia\pi x|^m}{m!} |b_{k,m}| = C \sum_{m=0}^{\infty} \frac{|a\pi x|^m}{m!} |\xi|^m \sum_{k=-\infty}^{\infty} |f(k\Delta)|. \quad (1.19)$$

Since the Taylor series of the function e^x is absolutely convergent over the whole real axis, in order that $|\widetilde{f}(x)|$ be finite, the series $\sum_{k=-\infty}^{\infty} |f(k\Delta)|$ must converge. This convergence will take place if $f(x) \in L_1[-\infty, \infty]$. So, the theorem is proved.

One can see that, when a=2 and Δ=2π, series (1.18) coincides with the Zelkin–Kravchenko series (1.17).

The approximation with the functions of the general form (1.16) is provided by the following theorem.

Theorem 2. *Let the function f(x) have a finite spectrum* ($\operatorname{supp} \widehat{f}(p) = [-\Omega, \Omega]$). *Then the following exact expansion is valid:*

$$f(x) = \sum_{k=-\infty}^{\infty} f(k\Delta) F_a \left[\frac{a\pi}{\Delta} (x - k\Delta) \right], \quad (1.20)$$

where $F_a(x)$ is defined by expression (1.16) *and conditions*

$$a > 2, \ \Delta \leqslant \frac{\pi}{\Omega} \cdot \frac{a-2}{a-1}, \quad (1.20a)$$

$$\text{or} \quad \Delta < \frac{\pi}{\Omega}, \ a \geqslant \frac{2 - \Delta\Omega/\pi}{1 - \Delta\Omega/\pi} \quad (1.20b)$$

are fulfilled.

Proof. Define the auxiliary function

$$\varphi(x) \equiv f(x) \prod_{n=2}^{\infty} \operatorname{sinc} \left(\frac{\pi a}{\Delta a^n} (z - x) \right), \quad z \in R.$$

This function also has a finite spectrum and $\operatorname{supp} \widehat{\varphi}(p) = [-\alpha; \alpha]$, where $\alpha = \Omega + \frac{\pi}{\Delta} \sum\limits_{n=1}^{\infty} \frac{1}{a^n} = \pi \left(\frac{\Omega}{\pi} + \frac{1}{\Delta(a-1)} \right)$.

In order that $\varphi(x)$ be expandable into the Kotelnikov series, the condition $\alpha \leqslant \pi/\Delta$ must be fulfilled. Hence, inequalities (1.20a) and (1.20b) must also be fulfilled. In this case, we have

$$\varphi(x) = \sum_{k=-\infty}^{\infty} \varphi(k\Delta)\,\mathrm{sinc}\left[\frac{\pi}{\Delta}(x-k\Delta)\right] = \\ = \sum_{k=-\infty}^{\infty}\left\{f(k\Delta)\prod_{n=2}^{\infty}\mathrm{sinc}\left(\frac{\pi a}{\Delta a^n}(z-k\Delta)\right)\right\}\mathrm{sinc}\left[\frac{\pi}{\Delta}(x-k\Delta)\right].$$

Let $x = z$. Then, $\varphi(z) = f(z) = \sum\limits_{k=-\infty}^{\infty} f(k\Delta)\prod\limits_{n=1}^{\infty}\mathrm{sinc}\left(\frac{\pi a}{\Delta a^n}(z-k\Delta)\right)$.

The latter expression is the required expansion (1.20) to within the denotation of the independent variable. The theorem is proved.

Corollary. If $f(x) \equiv 1$, then $\Omega = 0$ ($\operatorname{supp}\widehat{f}(p) = \{0\}$), i.e., $\widehat{f}(p) = \delta(p)$, and, for all $\Delta > 0$ and $a \geqslant 2$, the following partition of unity takes place:

$$\sum_{k=-\infty}^{\infty}\prod_{n=1}^{\infty}\mathrm{sinc}\left[\frac{\pi}{\Delta a^{n-1}}(x-k\Delta)\right] = 1. \tag{1.21}$$

If $f(x)$ does not satisfy conditions of Theorem 2, series (1.20) can be treated as an approximate representation of this function. In this case, the approximation will be exact at the points $k\Delta$. The first problem in practical calculations is that we have to use a finite number of terms in the product on right of formula (1.16):

$$F_a(p) = \prod_{k=1}^{M}\mathrm{sinc}\,(p/a^k). \tag{1.22}$$

After reasoning similar to that used in the proof of Theorem 2, one may conclude that, in this case, we also obtain an *exact* expansion (1.20) if the conditions of the theorem are replaced with the weaker constraints: $a(1+a^{-M}) > 2$, $\Delta \leqslant \frac{\pi}{\Omega}\cdot\frac{a(1+a^{-M})-2}{a-1}$.

Table 1.1 summarizes the minimum possible values of parameter a for various numbers of terms M in the product obtained from the solution of the transcendental equation $a(1+a^{-M})-2=0$.

Table 1.1. The minimum possible values of parameter a for various numbers of terms M in the product in (1.16).

M	a	M	a	M	a	M	a
1	1	4	1.8405	7	1.9843	10	1.9980
2	1.0241	5	1.9277	8	1.9922	11	2.0000
3	1.6189	6	1.9660	9	1.9961	12	2.0000

Evidently, for $M = 1$, we obtain, as a special case, the Kotelnikov series; and, in the limit $M \to \infty$, the series (1.20) is correct.

The second problem is that we have to use a finite number of terms in series (1.20):

$$\widetilde{f}_N(x) = \sum_{k=-N}^{N} f(k\Delta)F_a\left[\frac{a\pi}{\Delta}(x-k\Delta)\right].$$

The effect of this truncation is not so important as the truncation of the Kotelnikov series, because the sidelobe levels of functions (1.16) and (1.22) are far lower than that of $\operatorname{sinc}(p)$.

1.6. An Example of Evaluating $h_a(x)$

1. The computational formula for evaluating $h_a(x)$ is written in the form

$$h_a(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{jux} \prod_{k=1}^{\infty} \frac{\sin\left(u \cdot a^{-k}\right)}{a \cdot 2^{-k}} du. \tag{1.23}$$

Let us restrict ourselves to a finite value of $k$in calculations:

$$\widetilde{h}_a(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{jux} \prod_{k=1}^{M} \frac{\sin\left(u \cdot a^{-k}\right)}{a \cdot 2^{-k}} du. \tag{1.24}$$

From property 3 it follows that, in practice (for small values of x), it is sufficient to take a small number of terms in the product in formula (1.16), $M = 4 \div 6$.

Using the aforementioned argumentation, we can construct plots and find maximal values of the absolute and relative error for $M = 4$ (as a standard function, we take $h_a(x)$ evaluated at $M = 8$ and $M = 12$, $a = 3$).

$$\begin{aligned} \delta_{\max} &= \max_{x \in [-1/(a-1), 1/(a-1)]} (|h_3(x) - \widetilde{h}_3(x)|), \\ \widetilde{\delta}_{\max} &= \max_{x \in [-1,1]} \left(\frac{|h_3(x) - \widetilde{h}_3(x)|}{\widetilde{h}_3(x)} \right) \cdot 100\%. \end{aligned} \tag{1.25}$$

The relative error is estimated as $\delta_{\max} \leqslant 1.7 \cdot 10^{-4}$ at $M = 4$ ($\widetilde{h}_a(x)$ is evaluated at $M = 8$), which does not exceed 0.024% at the endpoints of the interval $[-1/(a-1), 1/(a-1)]$; for $M = 4$ ($\widetilde{h}_a(x)$ is evaluated at $M = 12$), $\delta_{\max} \leqslant 1.5 \cdot 10^{-3}$ and $\widetilde{\delta}_{\max} \leqslant 1.5 \cdot 10^{-3}$ (0.15 %).

1.7. Atomic Functions $\Xi_n(x)$

Consider the atomic function $\Xi_n(x)$. By definition, $\Xi_n(x)$ is a compactly supported solution of the equation

$$y^n(x) = a \sum_{k=0}^{n} C_n^k (-1)^k y[(n+1)x + n - 2k] \tag{1.26}$$

on the interval [–1, 1], where $a = (n+1)^{n+1} \cdot 2^{-n}$ and $\int_{-1}^{1} \Xi_n(x) dx = 1$.

The functions $\Xi_n(x)$ are generalizations of the function $\operatorname{up}(x)$ (Fig. 1.4). As follows from Fig. 1.4 *a*, the width of $\Xi_n(x)$ decreases as n becomes greater, while its maximum value grows. With the increase of n, the sidelobe level of the Fourier transform $K_n(t)$ decreases and the main lobe width grows (Fig. 1.4 *b*).

Using transforms analogous to those made in [7, 8, 17], we obtain the following integral representation of $\Xi_n(x)$:

$$\Xi_n(x) = \frac{1}{2\pi} \int\limits_{-\infty}^{\infty} \exp\{ixt\} \prod_{k=1}^{\infty} \left(\frac{\sin t(n+1)^{-k}}{t(n+1)^{-k}} \right)^n dt. \tag{1.27}$$

If n=1, then $\Xi_1(x) = \text{up}\,(x)$. The results obtained for the function $\text{up}\,(x)$ can be derived for the function $\Xi_n(x)$ $(n>1)$ in the same way. The function $\Xi_n(x)$ asymptotically tends to a normalized compactly supported nth order spline $\theta_n(x)$ in the norm of the space L_1: $\left\| \Xi_n(x) - \dfrac{\theta_n(x)}{||\theta_n(x)||_{L_1}} \right\|_{C^k} \to 0$ as $n \to \infty$.

This function is infinitely fractured and its fracture components belong to the class of atomic functions.

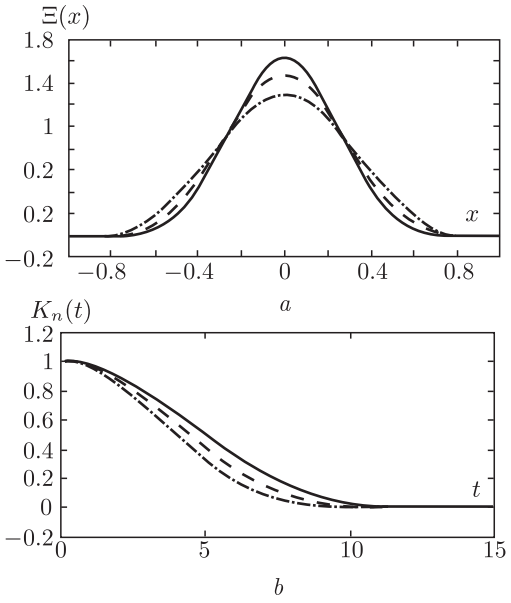


Fig. 1.4. Atomic function $\Xi_n(x)$: (a) plots of functions $\Xi_n(x)$; (b) its Fourier transform $K_n(t)$ at $n = 4$ (solid line), $n = 3$ (dashed line), $n = 2$ (dot-and-dashed line).

1.8. Atomic Functions $g_{k,h}(x)$

Consider the equation [11–13]

$$g''_{k,h}(x) + k^2 g_{k,h}(x) = a g_{k,h}(3x + 2h) - b g_{k,h}(3x) + a g_{k,h}(3x - 2h) \tag{1.28}$$

under the conditions that $\text{supp}\, g_{k,h}(x) = [-h, h]$, $\int\limits_{-h}^{h} g_{k,h}(x)dx = 1$.

The compactly supported solution of this equation is

$$g_{k,h}(x) = \frac{1}{2\pi} \int\limits_{-\infty}^{\infty} F_{k,h}(t) \exp\{-itx\} dt, \tag{1.29}$$

where

$$F_{k,h}(t) = \prod_{j=1}^{\infty} \frac{\dfrac{k^2}{1-\cos(2kh/3)} \left[\cos(2th3^{-j}) - \cos(2kh/3)\right]}{k^2 - t^2 9^{1-j}},$$

$$a = \frac{3}{2} \frac{k^2}{1-\cos(2kh/3)}, \quad b = 2a\cos(2kh/3).$$

Let analyze equation (1.28) at $h = 1$ (Fig. 1.5). It can be shown that functions $g_k(x)$ are infinitely fractured and their fracture components have the form (1.29), where

$$F_{k,m}(t) = \left| \frac{2k^2}{1-\cos(2k/3)} \right|^m \frac{\sin\dfrac{k+t}{3^{m+1}} \sin\dfrac{k-t}{3^{m+1}}}{(k+t)(k-t)} \times$$

$$\times \prod_{i=2}^{m} \frac{\sin\left(\dfrac{k}{3^i} + \dfrac{t}{3^{m+1}}\right) \sin\left(\dfrac{k}{3^i} - \dfrac{t}{3^{m+1}}\right)}{(k+t3^{1-i})(k-t3^{1-i})} \times$$

$$\times \prod_{j=m+1}^{\infty} \frac{2k^2}{1-\cos(2k/3)} \frac{\sin\left(\dfrac{k}{3} + t3^{-j}\right) \sin\left(\dfrac{k}{3} - t3^{-j}\right)}{(k+t3^{1-j})(k-t3^{1-j})}.$$

$$g_k''(x) + k^2 g_k(x) = a g_k(3x+2) - b g_k(3x) + a g_k(3x-2) \tag{1.30}$$

under the conditions $\operatorname{supp} g_k(x) = [-1, 1]$, $\int\limits_{-1}^{1} g_k(x) dx = 1$.

Applying the Fourier transform to both parts of (1.30) and designating $F_k(t) = \int\limits_{\infty}^{\infty} g_k(x) \exp\{itx\} dx$, we obtain the compactly supported solution $g_k(x) = \dfrac{1}{2\pi}$, where

$$F_k(t) = \prod_{j=1}^{\infty} \frac{\dfrac{k^2}{1-\cos(2k/3)} \left[\cos(2t3^{-j}) - \cos(2k/3)\right]}{k^2 - t^2 9^{1-j}}, \tag{1.31}$$

$$a = \frac{3}{2} \frac{k^2}{1-\cos(2k/3)}, \quad b = 2a\cos(2k/3).$$

The function $F_k(t)$ satisfies the functional equation

$$F_k(t) = \frac{k^2}{1-\cos(2k/3)} \frac{\cos(2t/3) - \cos(2k/3)}{k^2 - t^2} F_k\left(\frac{t}{3}\right).$$

Let us show that evaluation of the function $g_k(x)$ at the points $-1 + 2m/3^n$ $(m, n \in N)$, which form an everywhere dense set, can be reduced to calculating $F_k(k)$. Consider the differential equation

$$(D^2 + k^2) g_k(x) = f(x), \tag{1.32}$$

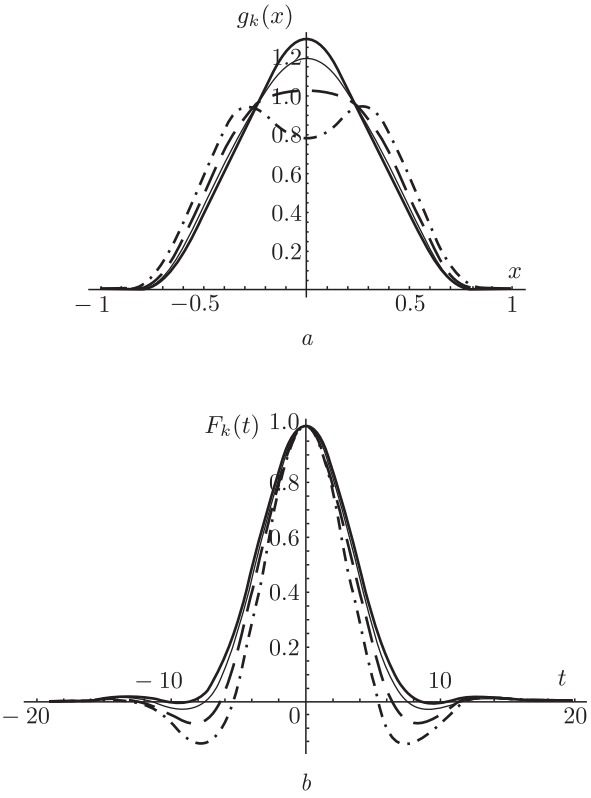


Fig. 1.5. Atomic function $g_k(x)$: (a) functions $g_k(x)$; (b) its Fourier transform $F_k(t)$ at $n = 1$ (solid line), $n = 2$ (thin line), $n = 3$ (dot-and-dashed line), and $n = 4$ (dashed line).

where $D^2 = \dfrac{d^2}{dx^2}$, $\quad f(x) = ag_k(3x+2) - bg_k(3x) + ag_k(3x-2)$.

Its general solution is $g_k(x) = \int\limits_{-1}^{x} G(x,t)f(t)dt + C_1 \cos kx + C_2 \sin kx$.

Here, $G(x,t) = \dfrac{1}{k}\sin(x-t)k$ is the Green function. To find the constants C_1 and C_2, we write

$$g_k'(x) = \int\limits_{-1}^{x} G'(x,t)f(t)dt + G(x,t)|_{x=t}\, f(t) - kC_1 \sin kx + kC_2 \cos kx.$$

Taking into account that the function is compactly supported and

$$g_k(-1) = g_k(1) = g_k'(-1) = g_k'(1) = 0, \tag{1.33}$$

we obtain $\begin{cases} C_1 \cos k - C_2 \sin k = 0, \\ kC_1 \sin k + kC_2 \cos k = 0. \end{cases}$

Since $G(x,x)=0$, we have $C_1=C_2=0$. Thus, the solution of equation (1.32) with boundary conditions (1.33) is

$$g_k(x)=\int\limits_{-1}^{x}G(x,t)f(t)dt. \tag{1.34}$$

Substituting the function $f(x)$ into equation (1.34), we obtain

$$g_k(x)=\int\limits_{-1}^{x}G(x,t)\left\{ag_k(3t+2)-bg_k(3t)+ag_k(3t-2)\right\}dt. \tag{1.35}$$

Making corresponding substitutions in (1.35) and taking into account that $G(x,t)=$ $=\frac{1}{2ki}\left\{\exp[ik(x-t)]-\exp[-ik(x-t)]\right\}$, we get

$$\begin{aligned}
g_k(x)=\frac{1}{2ki}\Bigg\{\frac{a}{3}\Bigg[&\exp\left\{ik\left(x+\frac{2}{3}\right)\right\}\int\limits_{-1}^{3x+2}g_k(u)\exp\left\{-\frac{iku}{3}\right\}du-\\
&-\exp\left\{-ik\left(x+\frac{2}{3}\right)\right\}\int\limits_{-1}^{3x+2}g_k(u)\exp\left\{\frac{iku}{3}\right\}du\Bigg]-\\
&-\frac{b}{3}\Bigg[\exp\{ikx\}\int\limits_{-3}^{3x}g_k(u)\exp\left\{-\frac{iku}{3}\right\}du-\\
&-\exp\{-ikx\}\int\limits_{-3}^{3x}g_k(u)\exp\left\{\frac{iku}{3}\right\}du\Bigg]+\\
+\frac{a}{3}\Bigg[&\exp\left\{ik\left(x-\frac{2}{3}\right)\right\}\int\limits_{-5}^{3x-2}g_k(u)\exp\left\{-\frac{iku}{3}\right\}du-\\
&-\exp\left\{-ik\left(x-\frac{2}{3}\right)\right\}\int\limits_{-5}^{3x-2}g_k(u)\exp\left\{\frac{iku}{3}\right\}du\Bigg]\Bigg\}.
\end{aligned} \tag{1.36}$$

Let us evaluate $g_k\left(-\frac{1}{3}\right)$. From (1.36) it follows that

$$g_k\left(-\frac{1}{3}\right)=\frac{a}{6ki}\left\{\exp\left\{\frac{ik}{3}\right\}F\left(-\frac{k}{3}\right)-\exp\left\{-\frac{ik}{3}\right\}F\left(\frac{k}{3}\right)\right\},$$

where $F_k(\alpha)$ is the Fourier transform of $g_k(x)$, defined as $F_k(\alpha)=\int\limits_{-\infty}^{\infty}g_k(x)\exp\{i\alpha x\}dx$. Since $F_k(\alpha)$ and $g_k(x)$ are even functions,

$$F_k\left(-\frac{k}{3}\right)=F_k\left(\frac{k}{3}\right)\quad\text{and}\quad g_k\left(\pm\frac{1}{3}\right)=\frac{a}{3k}\sin\frac{k}{3}F_k\left(\frac{k}{3}\right). \tag{1.37}$$

For the first derivative of $g_k(x)$, we obtain

$$g_k'\left(-\frac{1}{3}\right) = \frac{a}{3}\sin\frac{k}{3}F_k\left(\frac{k}{3}\right). \tag{1.38}$$

Since $g_k'\left(-\frac{1}{3}\right) = -g_k'\left(\frac{1}{3}\right)$, we have

$$g_k'\left(\frac{1}{3}\right) = \frac{a}{3}\cos\frac{k}{3}F_k\left(\frac{k}{3}\right). \tag{1.39}$$

Substituting the value $a = \frac{3}{2}\frac{k^2}{1-\cos 2k/3}$ into (1.37)–(1.39), we get

$$g_k\left(\pm\frac{1}{3}\right) = \frac{k}{4\sin\frac{k}{3}}F_k\left(\frac{k}{3}\right), \quad g_k'\left(\pm\frac{1}{3}\right) = \pm\frac{k^2\cos\frac{k}{3}}{4\sin^2\frac{k}{3}}F_k\left(\frac{k}{3}\right). \tag{1.40}$$

To evaluate the function $g_k(x)$ at the points $-7/9$, $-5/9$, apply the differential operator D^2+9k^2 to both parts of equality (1.40). Then we have

$$(D^2+9k^2)(D^2+9k^2)g_k(x) = f_1(x), \tag{1.41}$$

where

$$\begin{aligned} f_1(x) = 9\{&a^2g_k(9x+8) - abg_k(9x+6) + a^2g_k(9x+4) - \\ &- abg_k(9x+2) + b^2g_k(9) - abg_k(9x-2) + a^2g_k(9x-4) - \\ &- abg_k(9x-6) + a^2g_k(9x-8)\}. \end{aligned}$$

The general solution of equation (1.41) can be written as

$$\begin{aligned} g_k(x) = \frac{1}{8k^2}\left\{\int_{-1}^{x} G_1(x,t)f_1(t)dt - \int_{-1}^{x} G_2(x,t)f_1(t)dt\right\} + C_1\exp\{ikx\} + \\ + C_2\exp\{-ikx\} + C_3\exp\{3ikx\} - C_4\exp\{-3ikx\}, \end{aligned} \tag{1.42}$$

where $G_1(x,t) = \frac{1}{k}\sin(x-t)k$, $\quad G_2(x,t) = \frac{1}{3k}\sin 3(x-t)k$.

Taking into account that the function $g_k(x)$ and all its derivatives vanish at the points ±1 and $G_1(x,t)|_{x=t} = G_1''(x,t)|_{x=t} = 0$, $G_2(x,t)|_{x=t} = G_2''(x,t)|_{x=t} = 0$, we get $C_1 = C_2 = C_3 = C_4 = 0$ and

$$g_k(x) = \frac{1}{8k^2}\left\{\int_{-1}^{x} G_1(x,t)f_1(t)dt - \int_{-1}^{x} G_2(x,t)f_1(t)dt\right\}.$$

Let us evaluate $g_k\left(-7/9\right)$ and $g_k\left(-5/9\right)$. We have

$$g_k\left(-\frac{7}{9}\right) = \frac{a^2}{8ik^3}\left\{\sin\frac{k}{9}F_k\left(\frac{k}{9}\right) - \frac{1}{3}\sin\frac{k}{3}F_k\left(\frac{k}{3}\right)\right\}. \tag{1.43}$$

$$g_k\left(-\frac{5}{9}\right) = \frac{1}{8k^3}\left\{a^2\sin\frac{k}{3} - ab\sin\frac{k}{9}\right\}F_k\left(\frac{k}{9}\right) - \frac{1}{24k^3}\left\{a^2\sin k - ab\sin\frac{k}{3}\right\}F_k\left(\frac{k}{3}\right). \tag{1.44}$$

Since $a = \dfrac{3}{2}\dfrac{k^2}{1-\cos\frac{2}{3}k}$, $\quad b = 2a\cos\dfrac{2}{3}k$, from (1.29) it follows that

$$F_k\left(\frac{k}{3}\right) = \frac{9}{4}\frac{\sin^2\frac{2k}{9}\cos\frac{2k}{9}}{\sin^2\frac{k}{3}}.$$

Here,

$$g_k\left(-\frac{7}{9}\right) = \frac{9k\sin\frac{k}{9}}{128\sin^5\frac{k}{3}}\left\{\sin\frac{k}{3} - 3\sin\frac{k}{9}\cos^2\frac{k}{9}\cos\frac{2k}{9}\right\}F_k\left(\frac{k}{9}\right), \tag{1.45}$$

$$g_k\left(-\frac{5}{9}\right) = \frac{9k}{128\sin^6\frac{k}{3}}\left\{\left(1-\cos\frac{2k}{3}\sin\frac{k}{9}\right)\sin^2\frac{k}{3} - \right.$$
$$\left. - \frac{3}{4}\left(\sin k - 2\cos\frac{2k}{3}\sin\frac{k}{3}\right)\sin^2\frac{2k}{9}\cos\frac{2k}{9}\right\}F_k\left(\frac{k}{9}\right), \tag{1.46}$$

$$g_k\left(-\frac{1}{3}\right) = \frac{9k}{16}\frac{\sin^2\frac{2k}{9}\cos\frac{2k}{9}}{\sin^3\frac{k}{3}}F_k\left(\frac{k}{9}\right).$$

The values of the derivative of $g_k(x)$ at these points are as follows:

$$g_k'\left(-\frac{7}{9}\right) = \frac{9k^2}{128\sin^4\frac{k}{3}}\left\{\cos\frac{k}{9} - \frac{9\sin^2\frac{2k}{9}\cos\frac{2k}{9}\cos\frac{k}{3}}{4\sin^2\frac{k}{3}}\right\}F_k\left(\frac{k}{9}\right),$$

$$g_k'\left(-\frac{5}{9}\right) = \frac{9k^2}{128\sin^4\frac{k}{3}}\left\{\cos\frac{k}{9} - 2\cos\frac{2k}{3}\cos\frac{k}{9} + \frac{9\sin^2\frac{2k}{9}\cos\frac{2k}{9}\cos\frac{k}{3}}{4\sin^2\frac{k}{3}}\right\}F_k\left(\frac{k}{9}\right),$$

$$g_k'\left(-\frac{1}{3}\right) = \frac{9k^2\cos\frac{k}{3}\sin^2\frac{2k}{9}\cos\frac{2k}{9}}{16\sin^4\frac{k}{3}}F_k\left(\frac{k}{6}\right).$$

From (1.29) at $t = k$ it follows that

$$F_k(k) = \prod_{j=0}^{n-1}\left[\frac{\frac{k^2}{1-\cos 2k\cdot 3^{-1}}\left\{\cos\left(2k\cdot 3^{-j-1}\right) - \cos 2k\cdot 3^{-1}\right\}}{k^2 - k^2\cdot 9^{1-j}}\right]F_k\left(\frac{k}{3^n}\right). \tag{1.47}$$

Let estimate $F_k\left(\dfrac{k}{3^n}\right)$ for large n. Expanding the function $F_k\left(\dfrac{k}{3^n}\right)$ into the Taylor series and restricting ourselves to its first three terms, we obtain $F_k\left(\dfrac{k}{3^n}\right) \sim 1 + \dfrac{F_k''(0)}{2}\left(\dfrac{k}{3^n}\right)^2$, because $F_k\left(\dfrac{k}{3^n}\right) \sim F_k(0) = 1$ and $F_k'(0) = 0$.

It can easily be shown that $F_k''(0) = -\dfrac{2k^2 - 9\left(1 - \cos\frac{2}{3}k\right)}{4k^2\left(1 - \cos\frac{2}{3}k\right)}$. So, it is easy to get from the last equations that

$$F_k\left(\frac{k}{3^n}\right) \sim \frac{\left(8 \cdot 3^{2n} + 9\right)\sin^2\frac{k}{3} - k^2}{8 \cdot 3^{2n}\sin^2\frac{k}{3}}. \tag{1.48}$$

1.9. Definition and the Main Properties of Functions $\mathrm{up}_m(x)$

The function $\mathrm{up}_m(x)$ is a generalization of $\mathrm{up}(x)$. Consider the functional-differential equation

$$y'(x) = a\sum_{k=1}^{m}(y(2mx + 2m - 2k + 1) - y(2mx - 2k + 1)). \tag{1.49}$$

As noted above, we assume that $m = 2, 3, 4\ldots$

Theorem 1. *For any m, equation* (1.49) *has a unique infinite differentiable solution* $y_m(x)$*compactly supported on the interval* $[-1, 1]$ *and satisfying, with* $a = 2$*, the normalization condition*

$$\int_{-\infty}^{\infty} y_m(x)dx = 1.$$

Remark. Since the Fourier transforms of functions $\mathrm{up}_m(x)$ are represented by infinite products, $\mathrm{up}_m(x)$ is an infinite-to-one convolution of some functions. It is interesting to study these functions.

We can write the following equation:

$$F_m(t) = \frac{1}{imt}\sum_{k=1}^{m}\left(\exp\left(it\frac{2k-1}{2m}\right) - \exp\left(it\frac{-2m+2k-1}{2m}\right)\right)F_m\left(\frac{t}{2m}\right). \tag{1.50}$$

Rewriting the function $F\left(\frac{t}{2m}\right)$ according to formula (1.50) and continuing this process as made above, we get

$$F_m(t) = \prod_{s=1}^{\infty} 2\frac{(2m)^{s=2}}{it}\sum_{k=1}^{m}\left(\exp\left(it\frac{2k-1}{(2m)^s}\right) - \exp\left(it\frac{-2m+2k-1}{(2m)^s}\right)\right). \tag{1.51}$$

The function corresponding to the factor with the number s in representation (1.51) will be denoted by $\varphi_{s,m}(x)$, $(s = 1, 2, 3, \ldots)$. From (1.51), taking into account the basic properties of the Fourier transform, we obtain that, on the interval [–1, 0],

$$\begin{gathered}\varphi_{s,m}(x) = \frac{k}{m},\\ \left(x \in \left[-1 + \frac{2k-1}{2m}, -1 + \frac{2k+1}{2m}\right]\right), \quad k = 0, 1, 2, \ldots, m-1,\end{gathered} \tag{1.52}$$

and $\varphi_{1,m}(x) = 1, \quad \left(x \in \left[-\frac{1}{2m}, 0\right]\right)$.

Furthermore, it is obvious that $\varphi_{1,m}(x)$ is expanded onto the interval [0, 1] as an even function and vanishes outside the interval [−1, 1]. From representation (1.52) it is seen that

$$\varphi_{s+1,m}(x) = 2m\varphi_{s,m}(2mx), \quad (s = 1, 2, 3, \dots). \tag{1.53}$$

It can be proved immediately that $\int_{-1}^{1} \varphi_{1,m}(x)dx = 1$, and it follows from equality (1.53) that $\int_{-1}^{1} \varphi_{s,m}(x)dx = 1$, $(s = 1, 2, 3, \dots)$. One should emphasize the fact (used below) that all $\varphi_{s,m}$ are nonnegative and, since the convolution of nonnegative functions is nonnegative, the function is also nonnegative.

Let us point out and prove the main properties of the functions $\operatorname{up}_m(x)$.

1. $\operatorname{supp} \operatorname{up}_m = [-1,1]$.
2. $\int_{-1}^{1} \operatorname{up}_m(x)dx = 1$.
3. $\operatorname{up}_m(x) = \dfrac{1}{2\pi} \int\limits_{-\infty}^{\infty} e^{itx} \prod\limits_{k=1}^{\infty} \dfrac{\sin^2\left(\dfrac{mt}{(2m)^k}\right)}{\dfrac{mt}{(2m)^k} m \sin\left(\dfrac{t}{(2m)^k}\right)} dt.$
4. $\operatorname{up}_m \in C^\infty$.
5. $\operatorname{up}_m(0) = 1$.
6. $\operatorname{up}_m(-x) = \operatorname{up}_m(x)$.
7. $\operatorname{up}_m(x)$ increases monotonically at $x \in [-1, 0]$ and decreases at $x \in [0, 1]$.
8. Integral shifts of $\operatorname{up}_m(x)$ form the partition of unity, i.e., $\sum\limits_{k=-\infty}^{\infty} \operatorname{up}_m(x-k) \equiv 1$.

For $f(x) \in C^\infty$, denote by $N_l(f)$ the set of all $x \in [-1, 1]$ such that $f^{(l)}(x) = 0$. Also denote $N_l(\operatorname{up}_m) = N_{l,m}$. Then,

9. $N_{l,m} = \left\{ \dfrac{2s}{(2m)^l} \right\}_{l=1}^{\infty}, \quad \left(s \in \mathrm{Z},\ |s| \leqslant \dfrac{(2m)^l}{2} \right), \quad N_{0,m} = \{-1,1\}.$

10. $\|\operatorname{up}_m^{(n)}\|_{C[-1,1]} = 2^n (2m)^{\frac{(n-1)n}{2}}$. Henceforward, this value will be denoted by $B_n^{(m)}$.

Let $\Delta^2(f(x)) = f(x-h) - 2f(x) + f(x+h)$ be the second difference of the function $f(x)$ at the point x with the step h.

11. $\Delta^2(\operatorname{up}_m^{(l)}(x)) = 0$ at the points $x \in \left[\dfrac{s}{m(2m)^l} \right]$ with the step

$$h = \frac{1}{m(2m)^l},$$
$$(s \in \mathrm{Z},\ |s| < m(2m)^l,\ s \neq 0 \pmod m,\ l = 0, 1, 2 \dots).$$

12. For derivatives of the functions $\operatorname{up}_m(x)$, the following formula is valid:

$$\operatorname{up}_m^{(l)}(x) = B_l^{(m)} \sum_{k=1}^{(2m)^l} \delta_k^{(m)} \operatorname{up}_m\left((2m)^l x + (2m)^l - 2k + 1\right). \tag{1.54}$$

Here, $\delta_k^{(m)} = (-1)^{\sum\limits_i p_i}$, where p_i is the ith digit in the $2m$-nary representation of the number $k/m - 1$, if m divides k, and the number $[k/m]$, if m does not divide k.

13. Functions $\operatorname{up}_m(x)$ are nonanalytic everywhere on their supports.

14. $\lim\limits_{m\to\infty} \operatorname{up}_m(x) = B_1$ in the uniform metrics, where

$$B_1 = \begin{cases} 1-|x|, & x \in [-1,1], \\ 0, & x \notin [-1,1]. \end{cases}$$

B_1 is the B-spline [1, 2].

1.10. Moments and Values of Functions $\operatorname{up}_m(x)$

Let $\mu_n^{(m)}$ be the moment of order n for the function $\operatorname{up}_m(x)$, i.e., $\mu_h^{(m)} = \int\limits_{-1}^{1} x^n \operatorname{up}_m(x)dx$. Since $\operatorname{up}_m(x)$ is an even function, $\mu_{2n-1}^{(m)} = 0$, $(n \in \mathrm{N})$.

Theorem 2. *The moments $\mu_{2n}^{(m)}$ are rational numbers evaluated by the recurrent formula*

$$\mu_{2n}^{(m)} = \frac{(2n)!}{m^2\left((2m)^{2n}-1\right)} \times \sum_{k=1}^{n} \frac{\sum\limits_{l=1}^{m}(2l-1)^{2k+1}}{(2n-2k)!(2k+1)!}\mu_{2n-2k}^{(m)}, \quad (n = 1,2,3,\dots). \tag{1.55}$$

Suppose that

$$\nu_{2n-1}^{(m)} = \int\limits_{0}^{1} x^n \operatorname{up}_m(x)dx \quad \text{for} \quad n = 0,1,2,\dots. \tag{1.56}$$

Theorem 3. *The formula*

$$\nu_{2n-1}^{(m)} = \frac{1}{n(2m)^{2n+1}} \sum_{l=0}^{n} \binom{2n}{2l} \sum_{k=1}^{m} (2k-1)^{2l}\mu_{2n-2l}^{(m)} \tag{1.57}$$

takes place.

Theorem 4. *The following formula takes place:*

$$\operatorname{up}_m(x_{m,n,s}) = \frac{2^{n+1}}{n!(2m)^{\frac{(n+1)(n+2)}{2}}} \times \sum_{j=1}^{s} \delta_j^{(m)} \sum_{k=0}^{[n/2]} \binom{n}{2k} (2s-2j+1)^{n-2k}\mu_{2k}^{(m)}, \tag{1.58}$$

where $[a]$ is the integral part of a number a. At $s = 1$ these formulas are reduced to

$$\operatorname{up}_m(x_{m,n,1}) = \frac{1}{m!} \int\limits_{-1}^{x_{m,n,1}} \operatorname{up}_m^{(n+1)}. \tag{1.59}$$

1.11. Atomic Functions $\pi_m(x)$

Consider the functional-differential equation

$$\pi_m'(x) = a\Big[\pi_m\left(x_1(m)\right) + \sum_{k=2}^{2m-1} (-1)^k \pi_m\left(x_k(m)\right) - \pi_m\left(x_{2m}(m)\right)\Big], \tag{1.60}$$

where $x_k(m) = 2mx + 2m - 2k + 1$, $x \in R^1$, $k = \overline{1,2m}$, $m = 3,4,5,\dots$

Apply the Fourier transform to both sides of equation (1.60) and denote $\int_{-\infty}^{\infty} e^{ixt}\pi_m(x)dx$ by $F_m(t)$. Then, the latter equality will take the form

$$-itF_m(t) = \frac{a}{2m}\left[e^{-it\frac{2m-1}{2m}} + \sum_{k=2}^{2m-1}(-1)^k e^{it\frac{2k-1-2m}{2m}} - e^{it\frac{2m-1}{2m}}\right]F_m\left(t/2m\right).$$

According to the Euler formula:

$$F_m(t) = \frac{a}{tm}\left[\sin\frac{2m-1}{2m}t + \sum_{k=2}^{2m-1}(-1)^k \sin\frac{2m-2k+1}{2m}t\right]F_m\left(t/2m\right).$$

Passing to the integral in equality (1.60) as $t \to 0$ and taking into account that the integral of $\pi_m(x)$ equals unity, we get $a = 2m^2/(3m-2)$. From the previous formula, taking into account the calculated value a, we obtain

$$F_m(t) = \frac{1}{(3m-2)t/(2m)^2}\left[\sin(2m-1)t/(2m)^2 + \right.$$
$$\left. + \sum_{k=2}^{m}(-1)^k \sin(2m-2k+1)t/(2m)^2\right]F_m\left(t/(2m)^2\right).$$

Then, acting as above, we find

$$F_m(t) = \prod_{k=1}^{m}\frac{\left[\frac{\sin(2m-1)t}{(2m)^k} + \sum_{\nu=2}^{m}(-1)^\nu \frac{\sin(2m-2\nu+1)t}{(2m)^k}\right]}{(3m-2)t/(2m)^k}. \tag{1.61}$$

Thus, making the inverse Fourier transform of the characteristic function $F_m(t)$, we obtain the integral representation for $\pi_m(x)$: $\pi_m(t) = \frac{1}{2\pi}\int\limits_{-\infty}^{\infty} e^{ixt}F_m(t)dt$, where $F_m(t)$ is determined by (1.61).

The properties of $\pi_m(x)$ (Figs. 1.6 a–c) are as follows:

1. $\operatorname{supp}\pi_m(x) = [-1, 1]$;
2. $\pi_m(-x) = \pi_m(x)$;
3. $\pi_m(x) \in C^\infty[-1, 1]$;
4. $\int\limits_{-\infty}^{\infty}\pi_m(x)dx = 1$;
5. $\pi_m(0) = \begin{cases} m/(3m-2), & \text{if } m \text{ is odd,} \\ 2m/(3m-2), & \text{if } m \text{ is even;} \end{cases}$
6. $N_\nu(\pi_m) = N_\nu^{\mathrm{T}}, \quad \nu \geqslant 0,$

where the set N_ν^{T} has the form

$$N_\nu^m = \{x \in [-1, 1] : x = x_{n,s} = 2s/(2m)^n\},$$
$$\{s : -\frac{1}{2}(2m)^n, \ldots, \frac{1}{2}(2m)^n\}, \quad n \geqslant 1,$$
$$N_0^m = \{x \in [-1, 1] : x = x_{0,s} = s = \pm 1\},$$
$$\{s : -1, 0, 1\}, \quad n = 0.$$

7. $\Delta^2_{h_\nu^2}\pi_m^{(\nu)}(x) = 0$, $\forall x \in D_\nu^m$, $\nu \geqslant 0$, where D_v^m is the set of points of the interval [–1, 1] at which the second difference with the step $h_v^2 = 2/(2m)^v$, $v \geqslant 0$, of the vth derivative of $\pi_m(x)$ equals zero, and $N_\nu^m = \{x \in [-1, 1] := x_{v,p}^* = -1 + 2p/(2m)^{v+1}\}$,

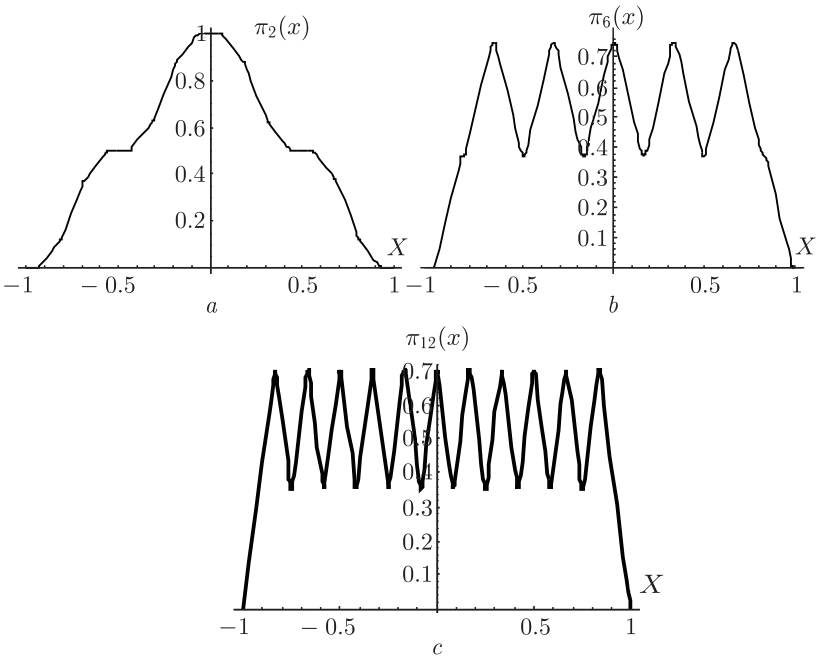


Fig. 1.6. Plots of $\pi_m(x)$ at $m = 2,\ 6,\ 12$.

where p is the running index $\{p : p \in Z^+,\ 0 < p < (2m)^{v+1},\ p \equiv \pm 1\ (\mathrm{mod} 2m)\}$ for $v \geqslant 0$.

8. $\Delta^1_{h^1_\nu} \pi^{(\nu)}_m(x) = 0 \quad \forall x \in T^m_\nu,\ \ \nu \geqslant 0$. The step h^1_ν is chosen by the definite rule [21]. T^m_v is the set of points from the interval [−1, 1] at which the first difference with the step $h^1_v = \pm 2/(2m)^v,\ \ v \geqslant 0$, of the νth derivative $\pi_m(x)$ vanishes. $T^m_v = \{x \in [-1, 1] : x = \widetilde{x}_{v,j} = -2j/(2m)^v\}$, where j is the running index $\{j : j \in Z^+,\ 0 < j < (2m)^{v+1},\ j \neq 0(\mathrm{mod} 2m),\ j \neq \pm 1(\mathrm{mod} 2m),\ j \neq \pm 2(\mathrm{mod} 2m),\}$ at $v \geqslant 0$.

9. $\left\| \pi^{(\nu)}_m(x) \right\|_{C[-1,1]} = K_{\nu,m},\ \nu \geqslant 0$;

10. Functions $\pi_m(x)$ are nonanalytic everywhere on their supports.

References

1. *Zelkin, Ye. G., Kravchenko, V. F., and Basarab M. A.* Interpolation of Signals with Compactly Supported Spectrums by Means of Fourier Transforms of Atomic Functions and its Use in Problems of Antenna Synthesis. Journal of Communications Technology and Electronics, 2002, vol. 47, no. 4, pp. 413–420.
2. *Kravchenko, V. F., Kolpakov, I. V., and Basarab M. A.* Atomic Functions and Their Application for Problems of Spline Interpolation. Elektromagnitnye volny i elektronnye sistemy, 2002, vol. 7., pp. 32–40.
3. *Kravchenko, V. F. and Golubin, M. V.* Spectral Properties of the New Weighting Functions Used in Digital Signal Processing. Elektromagnitnye volny i elektronnye sistemy, 2002, vol. 7., pp. 25–37
4. *Kravchenko, V. F. and Basarab, M. A.* Application of Atomic Functions for Restoring Signals with a Compactly Supported Spectrum. Doklady Mathematics, 2002, vol. 66. no. 1, pp. 152–156.

5. *Kravchenko, V.F. and Basarab, M.A.* Atomic Functions and Direct Methods for Solving Singular Integral Equations of Electromagnetics, Proc. of The Fourth International Symposium «Physics and Engineering of Millimeter and Sub-Millimeter Waves», June 4–9, 2001, Kharkov, Ukraine, vol. 1, pp. 238–240.
6. *Zelkin, E.G. and Kravchenko, V.F.* Approximation of Antenna Patterns by Atomic Functions. Antenny, 1999, no. 2(43), pp. 46–49.
7. *Kravchenko, V.F. and Zamyatin, A.A.* Application of Atomic Functions in Wavelet Matrix Transformation. Dokl. Akad. Nauk RAN, 1999, vol. 365, no. 3, pp. 340–342.
8. *Kravchenko, V.F., Rvachev, V.A., and Rvachev, V.L.* Mathematical Methods for Signal Processing Based on Atomic Functions. Journal of Communications Technology and Electronics, 1995, vol. 40(12), pp. 118–137.
9. *Kravchenko, V.F. and Rvachev, V.A.* Wavelet-Systems and Their Application in Signal Processing. Zarubezhnaya Radioelektronika. Uspekhi Sovremennoi Radioelektroniki, 1996, no. 4, pp. 3–20.
10. *Kravchenko, V.F. and Basarab, M.A.* New Methods of Signal Filtering Based on Atomic Functions, Proc. of The Fourth International Symposium «Physics and Engineering of Millimeter and Sub-Millimeter Waves», June 4–9, 2001, Kharkov, Ukraine, vol. 1, pp. 235–237.
11. *Kravchenko, V.F. and Basarab, M.A.* Novel Numerical Algorithms Based on Atomic Functions for Solving Ill-Posed Problems of Electrodynamics, Proc. of the URSI Int. Symposium on Electromagnetic Theory, 2001, May 13–17, Victoria, BC, Canada, pp. 716–718.
12. *Kravchenko, V.F. and Basarab, M.A.* Atomic Functions in Numerical Solution of Integral Equations, Proc. of the First International Workshop on Mathematical Modeling of Physical Processes in Inhomogeneous Media, March 20–22, 2001, Guanajuato, Gto. Mexico, pp. 8–10.
13. *Basarab, M.A., Kravchenko, V.F., and Masyuk, V.M.* R-Functions, Atomic Functions and their Applications. Zarubezhnaya Radioelektronika. Uspekhi Sovremennoi Radioelektroniki, 2001, no. 8, pp. 5–40.
14. *Kravchenko, V.F. and Basarab, M.A.* Approximations by Atomic Functions and Numerical Methods for Fredholm Integral Equations of the Second Kind. Differential Equations, 2001, vol. 37, no. 10, pp. 1480–1490.
15. *Basarab, M.A. and Kravchenko, V.F.* An Iterative Approach for Synthesis of an Interpolation Algebraic Polynomial with Respect to Atomic Functions. Elektromagnitnye Volny i Elektronnye Sistemy, 2000, vol. 5, no. 4, pp. 4–10.
16. *Basarab, M.A., Kravchenko, V.F., and Pustovoit, V.I.* Atomic Functions and Direct Methods for Solving Fredholm Integral Equations of the First Kind in the Problem of Synthesis of Linear Antennas. Doklady Physics, 2000, vol. 45, pp. 457–462.
17. *Kravchenko, V.F. and Zamyatin, A.A.* Application of Atomic Functions for Solving Integral Equations of Electrodynamics. Zarubezhnaya Radioelektronika. Uspekhi Sovremennoi Radioelektroniki, 1999. no. 3, pp. 58–62.
18. *Kravchenko, V.F. and Zamyatin, A.A.* Approximation Properties of Atomic Functions under Solving Electrodynamics Integral Equations by the Method of Moments. Dokl. Akad. Nauk, 1999, vol. 365, no. 4, pp. 475–477.
19. *Basarab, M.A. and Kravchenko, V.F.* The Construction of Equations for Boundaries of Fractal Domains with the Use of the R-function Method. Telecommunications and Radio Engineering, 2001, vol. 56, no. 12, pp. 2–13.
20. *Zelkin, E.G., Kravchenko, V.F., Pustovoit, V.I., and Timoshenko, V.V.* Approximation of Any Function by the Integer Functions of the Exponential Type. Dokl. Acad. Nauk RAN, 2000, vol. 371, no. 1, pp. 42–44.
21. *Starets, G.A.* One Class of Atomic Functions and Its Applications, Cand. Sc. (Phys.-Math.) Dissertation, Kharkov, Kharkov Gos. Univ., 1984.

Chapter 2

SPECTRAL PROPERTIES OF ATOMIC FUNCTIONS AND NOVEL WINDOWS

Let us consider the application of the AF to problems of digital signal processing and show their efficiency in comparison with processing by classical methods. Below, new weighting functions (windows) will be proposed and justified [1–4]. Introduction of such nonstandard windows offers an effective solution to problems that arose in the last years when a new class of ground-based and airborne radar capable of realizing simultaneous search and tracking of a large number of different targets appeared. The windows presented below are constructed by applying products, summations, and convolutions to simple windows as well as by composing separate parts of the known ones. As a rule, such windows do not have good performance and some of them are not required in modern practical needs. Therefore, the construction of new windows on the basis of the AF is of the great practical interest.

2.1. Synthesis of Novel Weighting Functions (Windows)

The short-term discrete Fourier transform (SDFT) of a signal $s(x)$ is written as $X_k(e^{j\omega}) = \sum\limits_{n=-\infty}^{\infty} w(k-n)s(n)e^{-j\omega n}$, where $w(k-n)$ is a weighting function used to isolate the input signal segment to be processed and corresponding to the discrete time moment k. Thus, a window $w(n)$ separates the necessary part of the signal by vanishing the latter outside the domain of interest. The window's form and width influence the signal frequency representation. An ideal frequency response must be characterized by a narrow main lobe providing good resolution, and the lack of sidelobes corresponding to the leak of energy. The latter is of great interest for signals possessing of non-uniform spectra with high amplitude and closely located peaks. In this case, the spectral peaks are extended and sidelobes with decreasing amplitudes are present. The overlapping of sidelobes corresponding to neighboring spectral peaks can cause their additional frequency shift, changing the main peaks' amplitudes and vanishing the small amplitude spectral components. It is not always possible to increase the length of the analyzed signal part, which improves the frequency resolution and weakens the effect of sidelobes overlapping, due to technical restrictions.

The use of the window in time domain influences the leak effect essentially. The absence of this function for a finite signal part of a signal analyzed is equivalent to the use of a rectangular window. This window is not optimal in analysis of stationary signals due to its discontinuity at the ends of the segment processed. A better weighting function should have zero values at both ends and vary monotonically inside the region of the processed signal part. The use of windows different from rectangular and smoothing discontinuities of the signal at the ends of the segment allows one to decrease the sidelobe level but, at the same time, the main lobe extends and resolution is degraded.

Recently a new class of V. F. Kravchenko–V. A. Rvachev windows based on the AF was used for solving signal processing problems. The following atomic weighting functions were applied [1]:

$$w_1(x) = \mathrm{up}\,(x),$$

$$w_2(x) = \mathrm{up}\,(x) + 0.01\,\mathrm{up}\,''(x),$$
$$w_3(x) = \mathrm{fup}_1(3x/2)/\,\mathrm{fup}_1(0),$$
$$w_4(x) = (\,\mathrm{fup}_1(3x/2) + 0.0036\,\mathrm{fup}_1''(3x/2))/(\,\mathrm{fup}_1(0) + 0.0036\,\mathrm{fup}_1''(0)),$$
$$w_5(x) = h_{3/2}(x),$$
$$w_6(x) = 1.0696(h_{3/2}(x) + h''_{3/2}(x)/121),$$
$$w_7(x) = \Xi_2(x)/\Xi_2(0).$$

The weighting windows are normalized as follows: $w(x)$=0 for $|x| > 1$, $w(0) = 1$, and $w(-x) = w(x)$. Figure 2.1 shows the atomic windows $w_1(x) - w_7(x)$ and their spectra.

Let us test the novel windows for the following properties: $w(nT) = 0$, $|n| > \dfrac{N}{2}$, N is even, $w(nT) = w(-nT)$.

To compare characteristics of different windows, the following system of physical parameters is used:

— *equivalent noise bandwidth* $k_1 = 2\dfrac{\int\limits_{-1}^{1} w^2(x)dx}{\left[\int\limits_{-1}^{1} w(x)dx\right]^2}$,

— *50% overlapping region correlation* $k_2 = \dfrac{\int\limits_{0}^{1} w(x)w(x-1)dx}{\int\limits_{-1}^{1} w^2(x)dx}100\%$,

— s*purious amplitude modulation (in dB)* $k_3 = -10\,\log\left|\dfrac{W(\pi/2)}{W(0)}\right|^2$,

where $W(p)$ is the Fourier transform of the window function;

— *maximum conversion losses (in dB)* $k_4 = 10\,\log(k_1) + k_3$;

— *maximum sidelobe level (in dB)* $k_5 = 10\,\log\max\limits_k\left|\dfrac{W(u_k)}{W(0)}\right|^2$,

where $\{u_k\}$ are the local maximum points (excluding u_0);

— *asymptotic decay rate of the sidelobes* (*in dB per octave*) $k_6 = 10\,\log\lim\limits_{u\to\infty}\left|\dfrac{W(2u)}{W(u)}\right|^2$;

— *window width at the 6 dB level* k_7=$2u$,

where u is the largest frequency such that $10\,\log\left|\dfrac{W(0)}{W(u)}\right|^2 = 6$,

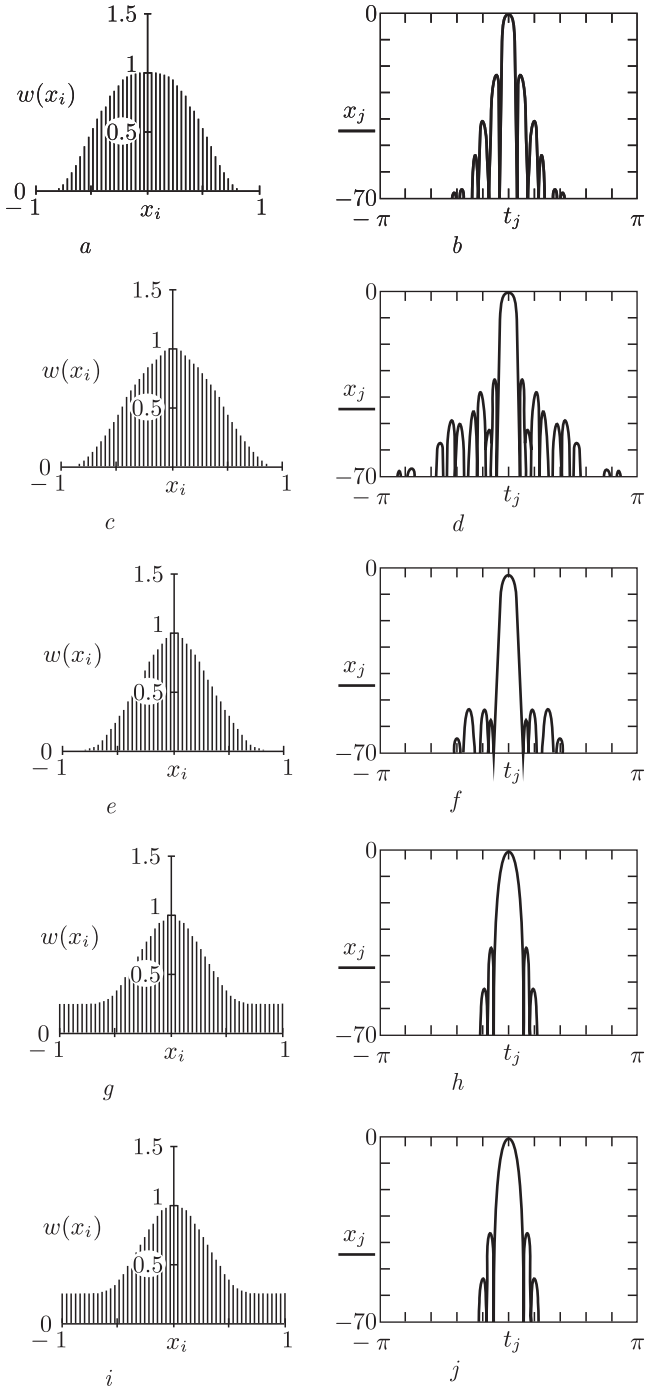
— *coherent gain* $k_8 = \dfrac{1}{2}\int\limits_{-1}^{1} w(x)dx$.

The following classical windows, widely used in practice, are introduced as comparative ones:

Kaiser–Bessel:

$$w(n) = \frac{I_0}{I_0(\alpha)}\alpha\sqrt{1 - \left(\frac{2n}{N-1} - 1\right)^2}, \quad 0 \leqslant n \leqslant N-1,$$

where I_0 is the zero-order Bessel function, and $\alpha = 2,\ 2.5,\ 3,\ 3.5$.


1.5
1
0.5
0
$w(x_i)$
-1
x_i
1
a
0
x_j
-70
$-\pi$
t_j
π
b
c
d
e
f
g
h
i
j

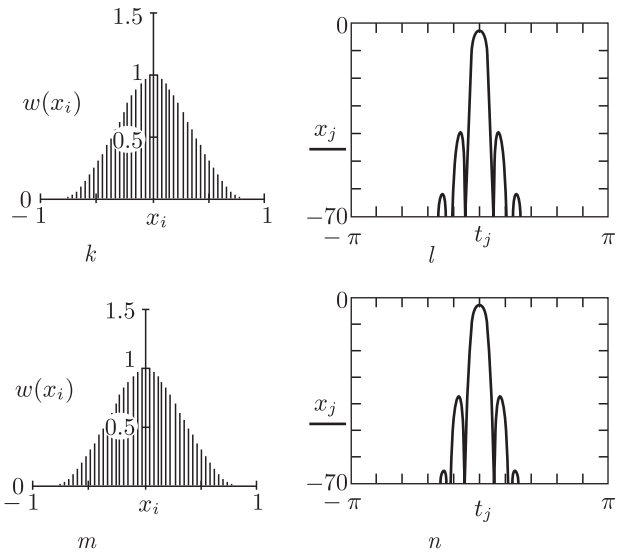


Fig. 2.1. Atomic weighting functions (windows) (a, c, e, g, i, k and m) and their Fourier transforms in a logarithmic scale (b, d, f, h, j, l and n).

Hamming:

$$w(n) = 0.54 - 0.46\cos\left(2\pi\frac{n}{N-1}\right), \quad 0 \leqslant n \leqslant N-1.$$

Blackman–Harris (four-termed):

$$w(n) = 0.35875 - 0.4883\cos\left(\frac{2\pi}{N}n\right) + 0.1413\cos\left(\frac{2\pi}{N}2n\right) - \\ - 0.0117\cos\left(\frac{2\pi}{N}3n\right), \quad 0 \leqslant n \leqslant N-1.$$

The comparative analysis of the main parameters of classical and new weighting functions is given in Tables 2.1 and 2.2.

2.2. Application of New Weighting Functions in Problems of Speech Synthesis

One of the main methods of speech processing is the short period spectral analysis (SSA) of speech, which provides a basis for numerous speech recognition systems, spectrographs, and voice coders. In its turn, the SSA involves the short period DPF (SDPF) of a speech segment weighted with a special window. Window functions are widely used in homomorphic speech processing, linear prediction, and encoding.

The detailed analysis of different algorithms of speech processing and prediction will be presented in the third part of this book.

As is known, speech signals can be categorized into voiced and unvoiced (fricative) speech. As compared with fricative speech, voiced speech has a higher energy level and a quasi-periodic structure. There exist transitional segments between voiced and unvoiced segments. One of the basic methods of speech processing is the SSA, which is based on the SDFT and enables one to reveal the signal features that cannot be detected in the time domain.

Table 2.1. Main physical parameters of classical and *Kravchenko–Rvachev* windows.

Windows	Parameters							
	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
Rectangular	1.0	50	3.9	3.9	−13.3	−6	1.2	1
Triangular (Bartlett)	1.3	25	1.8	3.1	−26.5	−12	1.7	0.5
Hamming	1.4	23	1.8	3.1	−43	−6	1.8	0.54
Hanning	1.5	17	1.4	3.2	−31.5	−18	1.9	0.5
Blackman	1.7	9	1.1	3.5	−58	−18	2.4	0.42
Kaiser–Bessel ($\beta = 3$)	1.8	7	1.0	3.6	−69	−6	2.4	0.4
Gauss ($\alpha = 6.25$)	1.5	19	1.6	3.2	−42	−6	1.9	0.49
Kravchenko–Rvachev:								
$w_1(x)$	1.6	12	1.2	3.3	−23.3	$-\infty$	2.1	0.5
$w_2(x)$	1.5	17	1.4	3.1	−32.4	$-\infty$	1.9	0.5
$w_3(x)$	1.9	6	0.9	3.6	−37.2	$-\infty$	2.4	0.39
$w_4(x)$	1.8	7	1.1	3.6	−51	$-\infty$	2.3	0.4
$w_5(x)$	1.3	30	0.7	1.7	−36	$-\infty$	2.9	0.52
$w_6(x)$	1.2	32	0.8	1.7	−51	$-\infty$	2.5	0.55
$w_7(x)$	1.9	5	0.9	3.7	−34	$-\infty$	2.4	0.38

Table 2.2. Main physical parameters of windows constructed on the basis of $\mathrm{fup}_N(x)$.

$\frac{\mathrm{fup}_N((N+2)x/2)}{\mathrm{fup}_N(0)}$	Parameters						
	b_1	b_2	b_3	b_4	b_5	b_7	b_8
$N = 0$	1.62	12	1.21	3.3	−23.3	2.08	0.5
$N = 1$	1.86	6	0.93	3.64	−37.2	2.4	0.39
$N = 2$	2.1	3	0.75	3.96	−50.8	2.7	0.35
$N = 3$	2.31	1	0.62	4.25	−64.2	2.97	0.31

To illustrate the aforesaid, let us consider a model of the speech signal [7–12] $s(t) = \sin(2\pi \cdot 500t) + 0.7\sin(2\pi \cdot 1350t) + 0.3\sin(2\pi \cdot 2300t) + 0.2\sin(2\pi \cdot 3400t)$ in the presence of additive noise with a zero mean and a unit amplitude (Fig. 2.2 *a*). Using the atomic window $w_1(x)$, let us process a 10ms signal segment (Fig. 2.2 *b*). The signal is sampled in frequency intervals of 10 kHz. Figures 2.2 *c*, *d* show the logarithmic absolute values of the SDFT of a signal weighted with the rectangular and atomic $w_1(x)$ windows. Since the Fourier transform of up(x) is known, the SDFT can easy be computed for the AF window. It is seen that the peaks of the frequency response that correspond to the harmonics of the original signal are narrower and sharper in the case

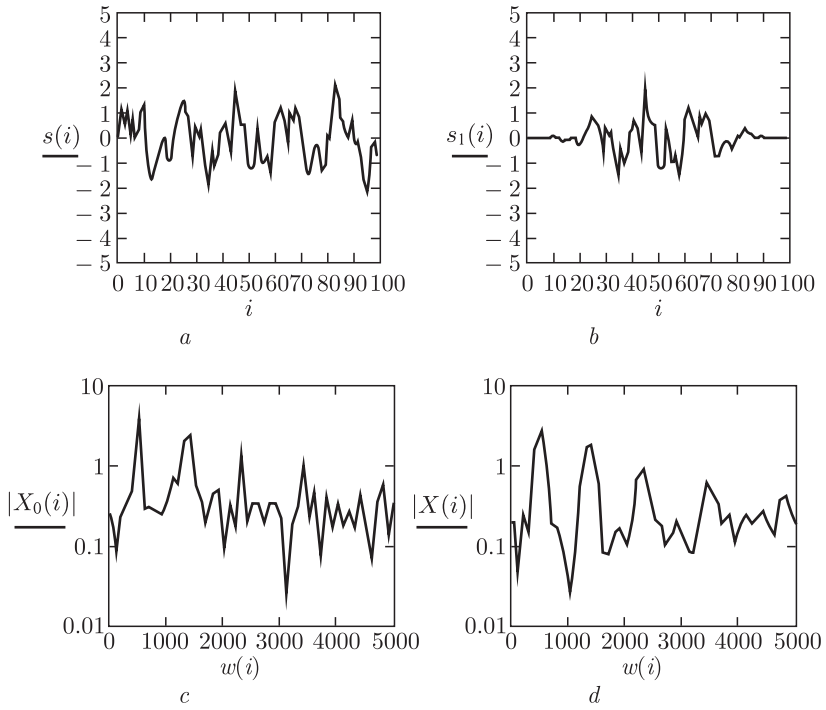


Fig. 2.2. Models of an original speech signal *(a)*, the signal treated with atomic window $w_1(x)$ (b), and the corresponding logarithmic absolute values of the SDFTs of a signal weighted with rectangular (c) and atomic $(w_1(x))$ (d) windows.

of a rectangular window (high-frequency resolution). At the same time, since high-level sidelobes result in energy loss in this case, the short-period spectrum looks noisier than the spectrum obtained in the case of the AF window. This circumstance hampers the identification of the original harmonics. Figure 2.3 demonstrates the short-period power spectrum of a signal weighted with the window $w_1(x)$. The peaks corresponding to the harmonics of the input sequence are clearly displayed.

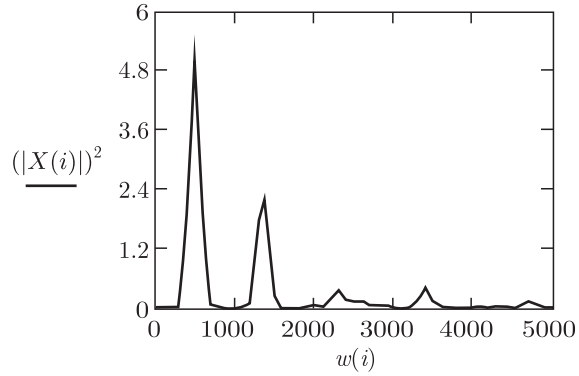


Fig. 2.3. Short period spectrum of an original speech signal (X) and the amplitude-frequency characteristic of a voice channel calculated using linear prediction (H).

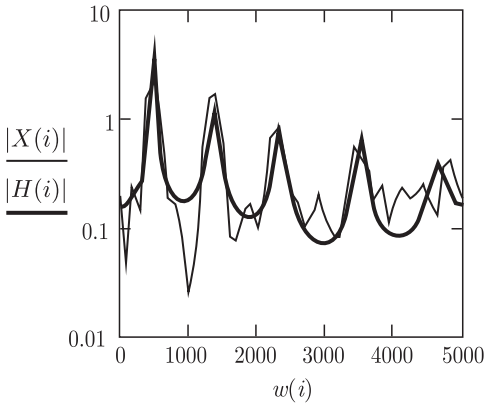


Fig. 2.4. Short period power spectrum of a signal segment treated with the atomic window $w_1(x)$.

Consider linear predictive coding applied to the determination of the frequency of the fundamental tone in the speech recognition, synthesis, and encoding. We suppose that discrete sequence $s[n]$ can be predicted from its preceding values $\widetilde{s}[n] = -\sum\limits_{k=1}^{P} a[k]s[n-k]$, where P is the order of a linear predictor and $a[k]$ are the coefficients of linear prediction. To minimize the prediction error, we use the least-squares method, which yields the system of equations:

$$\sum_{k=1}^{P} \widehat{a}[k] \sum_{n} s[n-k]s[n-m] = -\sum_{n} s[n]s[n-m], \quad 1 \leqslant m \leqslant P,$$

where $\widehat{a}[k]$ are the estimates of $a[k]$. In the general case, the summation must be performed over all n. However, actually, only a finite number of samples $s[n]$ are summed up, so as to make sequence $s[n]$ stationary. To this end, this sequence is truncated by the window $w[n]$:

$$s'[n] = \begin{cases} s[n]w[n], & 0 \leqslant n \leqslant N-1, \\ 0, & \text{otherwise.} \end{cases}$$

So, we obtain the system $r[m] = -\sum\limits_{k=1}^{P} \widehat{a}[k]r[m-k]$, $1 \leqslant m \leqslant P$, where $r[m] = \frac{1}{N-m} \sum\limits_{n=0}^{N-1-m} s'[n]s'[n+m]$ is the autocorrelation function of sequence $s'[n]$. Taking into account that the autocorrelation function is even, the above system of equations can be solved using the Levinson–Durbin recursive algorithm.

Linear speech prediction can be used for determining the frequency response of a voice channel:

$$H(e^{j\omega}) = \frac{G}{1 + \sum\limits_{k=1}^{P} \widehat{a}[k]e^{-jk\omega}},$$

where G is the gain. Figure 2.4 displays the short period spectrum of the original speech signal and the amplitude–frequency characteristic of the voice channel calculated using the linear prediction. The results are obtained with the atomic window $w_1(x)$ and the predictor of order P=10. The formants of the harmonic signal are clearly observed

in the plot. Using other atomic windows, one can efficiently solve more complicated problems of speech processing.

2.3. AF $\mathrm{up}\,(x)$, $\mathrm{fup}_N(x)$, $\Xi_n(x)$ and Their Combinations Used in Digital Signal Processing

The analysis of the results obtained above makes it possible to synthesize windows that are optimal from the viewpoint of sidelobe level minimization. Such windows should vanish at both ends and vary monotonically inside the interval. The use of the AFs, which satisfy such conditions, ensures decreasing the sidelobe level at the expense of the main lobe extension. The optimal choice of the window is determined by a compromise between the noise shift in the domain of sidelobes. So, if signal spectral components close with respect to the amplitudes, are situated both in the vicinity and far from the weak component, then windows with an equal level of sidelobes should be chosen. If we want to obtain high resolution between close signal components and distant components are absent, then windows with very narrow main lobe and minimum amplitude of neighboring sidelobes are required [2, 3]. Let us introduce the class of weighting windows based on the AFs $\mathrm{up}\,(x)$, $\mathrm{fup}_N(x)$, $\Xi_n(x)$, and their combinations with classical windows and consider the relation between their parameters and the function behavior.

2.4. Convolution Operation in Synthesis of New Windows

The synthesis of the weighting window $w(x)$ and the system of physical parameters were considered earlier. Here, we investigate new synthesized windows based on the AFs under the following assumptions: $w(nT) = 0$, $|n| > N/2$, N *is even*, and $w(nT) = w(-nT)$.

Let consider the window $K = \mathrm{up}\,(x) * \mathrm{up}\,(x)$ (Table 2.3) and determine its physical parameters.

Evaluations were performed for $N = 50$.

1. Equivalent noise bandwidth $k_1(K) = 50 \cdot 0.0468 = 2.34$.
2. Overlap correlation (50%) $k_2(K) = 0.8\%$.
3. Spurious amplitude modulation $k_3(K) = 0.6$.
4. Maximum conversion losses $k_4(K) = 4.3$ dB.
5. Maximum sidelobe level $k_5(K) = -47$ dB.
6. Sidelobe asymptotic decay rate $k_6(K) = -\infty$.
7. Window width at the six-decibel level $k_7(K) = 3.05$.
8. Coherent gain $k_8(K) = 0.31$.

For comparison, let present computational data for the synthesized window $K_2 = \mathrm{up}\,(x) * \mathrm{up}\,(x) * \mathrm{up}\,(x) * \mathrm{up}\,(x)$ (Table 2.2, Figs. 2.5 *c* and *d*).

1. Equivalent noise bandwidth $k_1(K_2) = 3.35$.
2. Overlap correlation (50%) $k_2(K_2) = 0.004\%$.
3. Spurious amplitude modulation $k_3(K_2) = 0.3$.
4. Maximum conversion losses $k_4(K_2) = 5.55$ dB.
5. Maximum sidelobe level $k_5(K_2) = -93.2$ dB.
6. Sidelobe asymptotic decay rate $k_6(K_2) = -\infty$.
7. Window width at the six-decibel level $k_7(K_2) = 4.45$.
8. Coherent gain $k_8(K) = 0.21$.

2.5. Numerical Simulations

The comparative numerical analysis of the new synthesized windows of Kravchenko (Table 2.3) and classical windows (Kaiser–Bessel, Hamming, four-termed Blackman–Harris) has demonstrated their advantages over the known windows with respect to some physical parameters. The new windows have low spurious amplitude modulation and sidelobe level, which, varying from -47 dB to -139.8 dB, is essentially dependent on the order of convolution. Correspondingly, the increased number of convolutions increases losses and mainlobe width. Another group of V. F. Kravchenko's windows is composed of convolutions of the AF $\Xi_n(x)$. The main parameters of the windows K_{Ξ_2} and K_{Ξ_4} are as follows (Table 2.3).

Table 2.3. Main physical parameters of the new windows synthesized by Kravchenko in comparison with classical ones.

Windows	Equivalent noise bandwidth, bin	Overlap correlation (50%)	Spurious amplitude modulation, dB	Maximum conversion losses, dB	Maximum sidelobe level, dB	Sidelobe asymptotic decay rate, dB per octave	Window width at the six-decibel level, bin	Coherent gain
	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8
K (Kravchenko)	2.34	0.8	0.6	4.3	-47	$-\infty$	3.05	0.31
K_1	2.9	0.06	0.4	5	-69.8	$-\infty$	3.82	0.25
K_2	3.35	0.004	0.3	5.55	-93.2	$-\infty$	4.45	0.21
K_3	3.75	$2.9\ 10^{-4}$	0.24	5.98	-116.4	$-\infty$	4.90	0.19
K_4	4.11	$2.1\ 10^{-5}$	0.2	6.34	-139.8	$-\infty$	5.41	0.17
K_{Ξ_2}	1.89	4.95	0.9	3.67	-34	$-\infty$	2.51	0.5
K_{Ξ_3}	2.14	2.1	1.35	4.66	-51	$-\infty$	2.05	0.5
K_{Ξ_4}	2.35	0.9	1.8	5.5	-68	$-\infty$	1.78	0.5
K_{Ξ_6}	2.73	0.7	2.7	7.1	-102	$-\infty$	1.5	0.5
KB, $a = 3,0$	1.8	7.4	1.02	3.56	-69	-6	2.39	0.4
KB, $a = 3,5$	1.93	4.8	0.89	3.74	-82	-6	2.57	0.37
Hamming	1.36	23.5	1.78	3.1	-43	-6	1.81	0.54
BH, four termed	2	3.8	0.83	3.85	-92	-6	2.72	0.36

1. Equivalent noise bandwidth

$$k_1(K_{\Xi_2}) = 50 \cdot 0.0178 = 1.89, \quad k_1(K_{\Xi_4}) = 50 \cdot 0.047 = 2.35.$$

2. Overlap correlation (50%)

$$k_2(K_{\Xi_2}) = 4.95\%, \quad k_2(K_{\Xi_4}) = 0.9\%.$$

3. Spurious amplitude modulation

$$k_3(K_{\Xi_2}) = -10 \log |W(\pi/2)/W(0)|^2 = 0.9, \quad k_3(K_{\Xi_4}) = 1.8.$$

4. Maximum conversion losses

$$k_4(K_{\Xi_2}) = 10 \log (1.89) + 0.9 = 3.67 \text{ dB}, \quad k_4(K_{\Xi_4}) = 5.5 \text{ dB}.$$

5. Maximum sidelobe level

$$k_5(K_{\Xi_2}) = -34 \text{ dB}, \quad k_5(K_{\Xi_4}) = -68 \text{ dB}.$$

6. Sidelobe asymptotic decay rate

$$k_6(K_{\Xi_2}) = -\infty, \quad k_6(K_{\Xi_4}) = -\infty$$

7. Window width at the six-decibel level
8. $k_7(K_{\Xi_2}) = 2.51, \quad k_7(K_{\Xi_4}) = 1.5.$
9. Coherent gain
10. $k_8(K_{\Xi_2}) = 0.5, \quad k_8(K_{\Xi_4}) = 0.5.$

Notice a high selectivity of this group of the Kravchenko windows: an increasing number of convolutions causes a decrease of the six-decibel bandwidth from 2.51 bin (windowK_{Ξ_2}) to 1.5 bin (window K_{Ξ_6}). The latter window (K_{Ξ_6}), apart from its good selectivity, possesses a low sidelobe level (−102 dB), low values of overlapping areas correlation (on the average, tenfold those for classical windows), and can be used for filtration of signals with a small distance between spectral peaks of equal intensity (for example, in digital radar, when it is necessary to detect a group of targets) and without strict requirements on the spurious amplitude modulation level.

Table 2.4. Main physical parameters of the new synthesized windows: Kravchenko–Hamming, Kravchenko–Kaiser–Bessel, Kravchenko–Blackman–Harris.

Windows	Equivalent noise bandwidth, bin	Overlap correlation (50 %)	Spurious amplitude modulation, dB	Maximum conversion losses, dB	Maximum sidelobe level, dB	Sidelobe asymptotic decay rate, dB per octave	Window width at the sixdecibel level, bin	Coherent gain
	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8
KH	2.14	2.06	0.74	4.05	−71.2	$-\infty$	2.78	0.35
KH_1	2.68	0.21	0.46	4.74	−96	$-\infty$	3.5	0.27
KH_2	3.17	0.016	0.17	5.1	−120	$-\infty$	4.1	0.23
KH_3	3.6	0.0012	0.026	5.8	−143	$-\infty$	4.77	0.2
KKB	2.44	0.58	0.56	4.43	−50.2	$-\infty$	3.2	0.29
KKB_1	2.97	0.042	0.38	5.1	−75.4	$-\infty$	3.9	0.24
KKB_2	3.4	0.003	0.29	5.6	−100	$-\infty$	4.45	0.21
KKB_3	3.81	$2.3\ 10^{-4}$	0.24	6	−123.6	$-\infty$	4.9	0.19
KBH	2.55	0.31	0.5	4.6	−44.2	$-\infty$	3.37	0.28
KBH_1	3.05	0.023	0.36	5.2	−68.7	$-\infty$	3.97	0.25
KBH_2	3.46	0.0026	0.233	5.6	−115.8	$-\infty$	5.09	0.21
KBH_3	3.85	$1.9\ 10^{-4}$	0.23	6	−116.2	$-\infty$	5.1	0.19

If weighting functions (windows) with small values of correlation, spurious amplitude modulation, and maximum sidelobe level (up to –143dB) are needed, one can use the sythesized Kravchenko–Hamming windows (Table 2.4), Kravchenko–Kaiser–Bessel windows (Table 2.4), and Kravchenko–Blackman–Harris windows (Table 2.4). All these windows have the infinite rate of sidelobe decay and small values of conversion losses. The main physical parameters of the new windows are presented in Tables 2.3 and 2.4

The diagram (Figure 2.5) illustrating the relation between conversion losses and maximum sidelobe level is important from the viewpoint of practical usage of weighting windows. A conclusion can be made that the new synthesized windows situated in the left lower corner of this diagram have the best quality. These windows have low sidelobe levels but relatively high values of conversion losses. The passage to translations and dilations of windows using the properties of the AF allows gives an essential improvement in the latter parameter.

Table 2.5 shows dependences of all main physical parameters of the new synthesized windows (Kravchenko K, Kravchenko–Hamming KH_1, and Kravchenko–Kaiser–Bessel KKB_2).

Table 2.5. Parameters of the new synthesized Kravchenko windows versus their time-domain dilation.

Windows	Relative time-domain dilation, %	Equivalent noise bandwidth, bin	Overlap correlation (50%)	Spurious amplitude modulation, dB	Maximum conversion losses, dB	Maximum sidelobe level, dB	Sidelobe asymptotic decay rate, dB per octave	Window width at the six-decibel level, bin	Coherent gain
		k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8
K	0	2.34	0.8	0.6	4.3	−46.66	$-\infty$	3.05	0.31
	10	2.1	2.35	0.74	3.96	−46.6	–	2.78	0.34
	20	1.87	5.64	0.94	3.66	−46	–	2.46	0.39
	40	1.42	20.5	1.67	3.2	−37.5	–	1.83	0.51
KH_1	0	2.68	0.21	0.46	4.74	–96	$-\infty$	3.5	0.27
	10	2.42	0.74	0.57	4.41	−84	–	3.2	0.3
	20	2.1	2.25	0.72	4.05	−75.1	–	2.85	0.33
	40	1.62	12.4	1.28	3.38	−57	–	2.15	0.44
KKB_2	0	3.4	0.003	0.29	5.6	−100	$-\infty$	4.45	0.21
	10	3.1	0.03	0.36	5.27	−100	–	4.07	0.23
	20	2.73	0.2	0.45	4.8	−99	–	3.6	0.26
	35	2.21	1.82	0.68	4.13	−96	–	2.93	0.32
	60	1.39	21.5	1.8	3.24	−60	–	1.78	0.52

If we dilate the weighting functions (windows) in the time domain, the following effects can be detected: the equivalent noise bandwidth and window width at the six-decibel level decrease for all windows and maximum conversion losses decrease significantly. Here, the following shortcomings should be noted: the correlation of overlapping regions increases and the parasitic amplitude modulation deteriorates. These effects are detected for the constant sidelobe level. Thus, the windows investigated are

not inferior to the known classical ones in their physical parameters, while some other parameters are sufficiently better. The obtained wide class of new weighting windows can be widely used in digital processing of signals with various spectral characteristics.

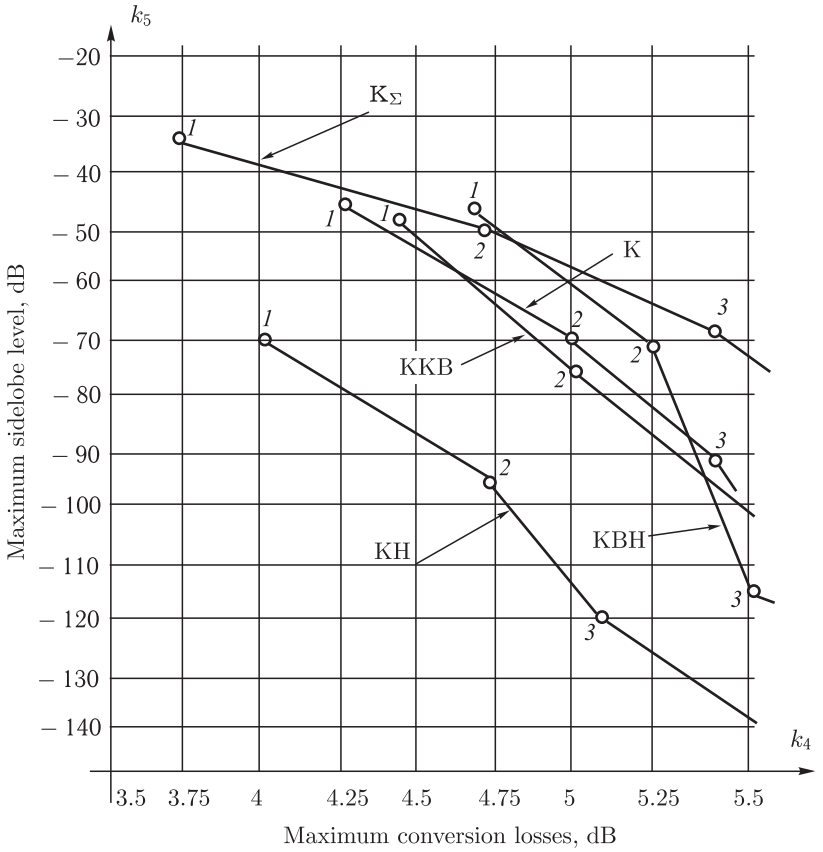


Fig. 2.5. Relation between conversion losses and maximum sidelobe level for some synthesized Kravchenko's windows.

2.6. Spectral Properties of New Weighting Functions Used in Digital Signal Processing

As is known, one of the main questions, common for all classic problems of signal spectral estimation, is the use of weighting functions (windows). Digital signal processing by means of windows is applied in practice for control of physical effects by spectral estimates in the presence of sidelobes. A new method for constructing weighting functions is developed and justified below. It is based on the combination (direct product) of the AFs $\mathrm{fup}_n(x)$ with the classical Gauss, Bernstein, and Dolph–Chebyshev functions. Characteristics of the new weighting functions as well as those of classical Hamming, Blackman–Harris, Natoll, and Kaiser–Bessel windows are presented [1–11].

2.7. Atomic Function $\mathrm{fup}_N(x)$ and Methods for Its Evaluation

The compactly supported functions $\mathrm{fup}_N(x)$ are the so-called fractional components of the function $\mathrm{up}(x)$.

The numerical realization requires computation of an infinite product and integration on the whole real axis. So, in practice, it is advisable to take the finite number of terms in the product

$$\widehat{\mathrm{fup}_N}(x) = \frac{1}{2\pi}\int\limits_{-\infty}^{\infty} e^{jux}\left(\frac{\sin(u/2)}{u/2}\right)^N \prod_{k=1}^{M}\frac{\sin\left(u\cdot 2^{-k}\right)}{u\cdot 2^{-k}}du.$$

For $M=5$, the relative error $\dfrac{\mathrm{fup}_N(x)-\widehat{\mathrm{fup}_N}(x)}{\mathrm{fup}_N(x)}$ does not exceed 0.1% at the ends of the support. For computations, it is more convenient to use the Fourier trigonometric expansion.

For $M=20$ and 10, the relative error at the boundaries of the interval $[-2,2]$ (for $\mathrm{fup}_2(x)$) does not exceed 0.05%. Here, computational efficiency increases sufficiently, because we do not evaluate improper integrals.

2.8. New Synthesized Windows

Window parameters. To estimate weighting functions, the following physical characteristics are used [6]: 1. Equivalent noise bandwidth (ENB); 2. Overlap correlation (50% overlap); 3. Spurious amplitude modulation (AM); 4. Maximum transformation loss; 5. Maximum sidelobe level; 6. 6-dB bandwidth; 7. Coherent gain; 8. Performance functional value.

The performance functional of new weighting functions. The performance functional $J(w)=\|w-w_{et}\|_{L^2[-1;1]}=\min$ is introduced for determining optimal weighting functions. This procedure consists of several stages. In the first stage, the aforementioned physical parameters of windows must be determined. In the second stage, values of the performance functional $J(w)$ for specific windows are evaluated. The analytical expression for the performance functional is

$$J(w)=J\left(k_4(w),k_5(w),k_7(w)\right)=\left(\frac{k_4(w_i)-k_4(w_e)}{k_4(w_e)}\right)^2+$$
$$+\left(\frac{k_5(w_i)-k_5(w_e)}{k_5(w_e)}\right)^2+\left(\frac{k_7(w_i)-k_7(w_e)}{k_7(w_e)}\right)^2, \quad (2.1)$$

where w_e is the standard window with the desired parameters $k_4=3$ dB, $k_5=-100$ dB, and $k_7=0.5$.

Definitions and designations of the new weighting functions are presented in Table 2.6.

The results of calculations according to equation (2.1) are shown in Table 2.7. The windows presented in Table 2.7 are normalized by $w(0)$.

Some known windows are also shown for comparison:

The Gauss function: $G_\alpha(t)=\exp(-(\alpha t)^2/2)$.

The Bernstein–Rogozinskii function: $B(t)=\cos(\pi t/2)$.

The Dolph–Chebyshev function: $D_\alpha(n)=\mathrm{F}^{-1}\left[W_\alpha(n)\right]$,

where

$$W_\alpha(n) = (-1)^n \frac{\cos\left[N \arccos\left(\beta_\alpha \cos\left[\pi\left(\frac{n}{N} - \frac{1}{2}\right)\right]\right)\right]}{\operatorname{ch}\left[N \operatorname{ch}^{-1}(\beta_\alpha)\right]},$$

and $\beta_\alpha = \operatorname{ch}\left[\frac{1}{N}\operatorname{ch}^{-1}(10^\alpha)\right]$. The domain is $n \in [-N/2, N/2]$.

Conversion from continuous time to discrete time is realized according to the following scheme (Fig. 2.6). First, the original window $w(x)$ is digitized with respect to the time variable $w[nT] = w(t)|_{t=nT}$.

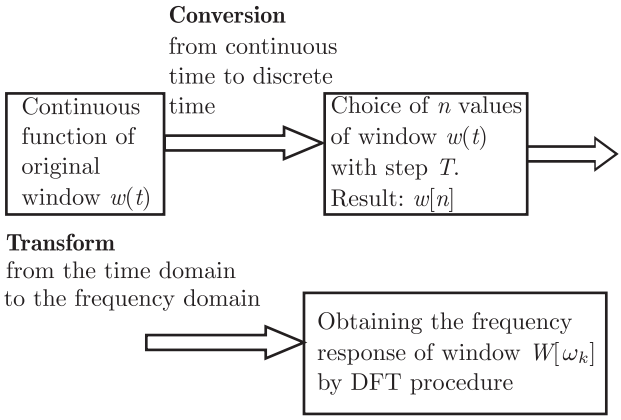


Fig. 2.6. Passage from a continuous window in time domain to amplitude-frequency response in frequency domain.

Since T is constant (the sampling period), window values can be denoted by $w[n]$. Then, we apply the DFT procedure to the obtained discrete window $w[n]$. If the number of samplings is multiple of 2^k, then we can use the fast Fourier transform (FFT):

$$W(\omega) = F[w[nT]] = \sum_{n=0}^{N-1} w[n] \cdot \exp(-j\omega nT),$$

where $\omega_k = \frac{2\pi}{N}$ is the sampling frequency.

Let illustrate this process with the window K_2^4. Pass from the continuous window $w(t) = \operatorname{fup}_2^4(t)$ to the discrete one $w[nT] = \operatorname{fup}_2^4(t)\big|_{t=nT}$. We get the Fourier transform as $W(\omega) = \sum\limits_{n=0}^{N-1} \operatorname{fup}_2^4[nT] \cdot \exp.(-j\omega nT)$. Here, we present values of physical parameters of the Kravchenko window K_2^4 ($N = 100$):

a. Equivalent noise bandwidth:

$$k_1(w(x)) = 100 \cdot 0.0199 \approx 1.99\,(bin).$$

b. 50% overlap correlation:

$$k_2(w(x)) = \frac{1.10}{25.88} \cdot 100\% = 4.25\,\%.$$

c. Spurious AM (dB):

$$k_3(w(x)) = -10 \cdot \lg\left|\frac{32.73}{36.10}\right|^2 = 0.85\,(dB),$$

where $W(\theta)$ is the Fourier transform of the window.

d. Maximum transformation losses:

$$k_4(w(x)) = 10 \cdot \lg(1.99) + 0.85 = 3.83\,(dB).$$

e. Maximum sidelobe level:

$$k_5(w(x)) = -51.6(dB),$$

where $\{\theta_k\}$ are the points of local maximums (excluding θ_0).

f. Window width at the six-decibel level $k_6(w(x)) = 2\theta = 2.63$, where θ is the highest frequency such that $10 \log \left|\frac{W(0)}{W(\theta)}\right|^2 = 6$.

g. Coherent gain:

$$k_7(w(x)) = 0.36.$$

Performance functional value:

$$J(w) = \left(\frac{3.83-3}{3}\right)^2 + \left(\frac{-51.6+100}{-100}\right)^2 + \left(\frac{0.36-0.5}{0.5}\right)^2 = 0.38.$$

Table 2.6. The new synthesized Kravchenko windows.

№	Windows	Discrete-time function $w(n)$
1	Kravchenko (K_2^4)	$w(n) = \text{fup}_2^4(2n/N)$
2	Kravchenko–Gauss (K_2G_2)	$w(n) = \text{fup}_2(2n/N) \cdot G_2(2n/N)$
3	Kravchenko–Gauss ($K_2^2G_2$)	$w(n) = \text{fup}_2^2(2n/N) \cdot G_2(2n/N)$
4	Kravchenko–Gauss (K_2G_3)	$w(n) = \text{fup}_2(2n/N) \cdot G_3(2n/N)$
5	Kravchenko–Bernstein–Rogozinskii (K_2BR^2)	$w(n) = \text{fup}_2(2n/N) \cdot B^2(2n/N)$
6	Kravchenko–Bernstein–Rogozinskii (K_2^2BR)	$w(n) = \text{fup}_2^2(2n/N) \cdot B\,(2n/N)$
7	Kravchenko–Bernstein-Rogozinskii ($K_2^2BR^2$)	$w(n) = \text{fup}_2(2n/N) \cdot B^2(x)$
8	Kravchenko (K_4^4)	$w(n) = \text{fup}_4^4(2n/N)$
9	Kravchenko–Gauss ($K_4^2G_2^2$)	$w(n) = \text{fup}_4^2(2n/N) \cdot G_2^2(2n/N)$
10	Kravchenko–Gauss ($K_4^4G_2$)	$w(n) = \text{fup}_4^4(2n/N) \cdot G_2(2n/N)$
11	Kravchenko–Gauss (K_4G_3)	$w(n) = \text{fup}_4(2n/N) \cdot G_3(2n/N)$
12	Kravchenko–Gauss ($K_4^2G_3$)	$w(n) = \text{fup}_4^2(2n/N) \cdot G_3(2n/N)$
13	Kravchenko–Bernstein–Rogozinskii (K_4^2BR)	$w(n) = \text{fup}_4^2(2n/N) \cdot B\,(2n/N)$
14	Kravchenko–Bernstein–Rogozinskii ($K_4^2BR^2$)	$w(n) = \text{fup}_4^2(2n/N) \cdot B^2(2n/N)$
15	Kravchenko–Dolph-Chebyshev (K_4C_3)	$w(n) = \text{fup}_4(2n/N) \cdot D_3(2n/N)$
16	Kravchenko–Dolph-Chebyshev ($K_4C_{3.5}$)	$w(n) = \text{fup}_4(2n/N) \cdot D_{3,5}\,(2n/N)$
17	Kravchenko–Gauss ($K_6^2G_2^2$)	$w(n) = \text{fup}_6^2(2n/N) \cdot G_2^2(2n/N)$
18	Kravchenko–Gauss (K_6G_3)	$w(n) = \text{fup}_6(2n/N) \cdot G_3(2n/N)$
19	Kravchenko–Gauss ($K_6^2G_3$)	$w(n) = \text{fup}_6^2(2n/N) \cdot G_3(2n/N)$
20	Kravchenko–Bernstein–Rogozinskii ($K_6^2BR^2$)	$w(n) = \text{fup}_6^2(2n/N) \cdot B^2(2n/N)$

Table 2.7. Main physical parameters of the new Kravchenko windows and the classical windows.

№	Windows	Equivalent noise bandwidth, bin	Overlap correlation (for the 50% overlap), %	Spurious amplitude modulation, dB	Maximum transformation loss, dB	Maximum sidelobe level, dB	6-dB bandwidth, bin	Coherent gain	Functional value
1	Kravchenko (K_2^4)	1.9861	4.2498	0.8518	3.8318	−51.6112	2.6276	0.3610	0.3883
2	Kravchenko–Gauss (K_2G_2)	1.5327	15.6455	1.4128	3.2673	−46.2344	2.0213	0.4675	0.3012
3	Kravchenko–Gauss ($K_2^2G_2$)	1.8105	7.4054	1.0259	3.6038	−53.7964	2.4255	0.3944	0.2986
4	Kravchenko–Gauss (K_2G_3)	1.9643	4.7297	0.8781	3.8101	−68.8390	2.6276	0.3614	0.2469
5	Kravchenko–Bernstein–Rogozinskii (K_2BR^2)	1.7393	8.5152	1.0775	3.4812	−45.7927	2.2234	0.4203	0.3450
6	Kravchenko–Bernstein–Rogozinskii (K_2^2BR)	1.7411	8.6540	1.0856	3.4939	−55.1020	2.2234	0.4166	0.2565
7	Kravchenko–Bernstein–Rogozinskii ($K_2^2BR^2$)	1.9674	4.1803	0.8528	3.7917	−54.5747	2.6276	0.3676	0.3461
8	Kravchenko (K_4^4)	1.6295	12.2556	1.2596	3.3801	−52.1313	2.2234	0.4371	0.2610
9	Kravchenko–Gauss ($K_4^2G_2^2$)	1.9631	4.7869	0.8809	3.8103	−70.6203	2.6276	0.3607	0.2369
10	Kravchenko–Gauss ($K_4^4G_2$)	1.9696	4.6700	0.8742	3.8180	−71.2806	2.6276	0.3598	0.2355
11	Kravchenko–Gauss (K_4G_3)	1.8782	6.1922	0.9609	3.6984	−64.4774	2.4255	0.3770	0.2409
12	Kravchenko–Gauss ($K_4^2G_3$)	2.0415	3.7429	0.8156	3.9152	−74.8054	2.6276	0.3467	0.2505
13	Kravchenko–Bernstein–Rogozinskii (K_4^2BR)	1.5642	14.1583	1.3313	3.2743	−48.9544	2.0213	0.4661	0.2735
14	Kravchenko–Bernstein–Rogozinskii ($K_4^2BR^2$)	1.8126	6.8755	0.9986	3.5816	−62.1117	2.4255	0.3998	0.2213

Continuation of Table 2.7.

№	Windows	Equivalent noise bandwidth, bin	Overlap correlation (for the 50% overlap), %	Spurious amplitude modulation, dB	Maximum transformation loss, dB	Maximum sidelobe level, dB	6-dB bandwidth, bin	Coherent gain	Functional value
15	Kravchenko–Dolph-Chebyshev (K_4C_3)	1.6932	10.1129	1.1569	3.4441	−65.4653	2.2234	0.4248	0.1638
16	Kravchenko–Dolph-Chebyshev ($K_4C_{3.5}$)	1.8007	7.3910	1.0249	3.5793	−74.9523	2.4255	0.3988	0.1410
17	Kravchenko–Gauss ($K_6^2G_2^2$)	1.8782	6.1959	0.9611	3.6986	−64.2160	2.4255	0.3769	0.2429
18	Kravchenko–Gauss (K_6G_3)	1.8344	7.0420	1.0064	3.6414	−62.2970	2.4255	0.3860	0.2398
19	Kravchenko–Gauss ($K_6^2G_3$)	1.9598	4.8492	0.8844	3.8066	−70.2968	2.6276	0.3611	0.2377
20	Kravchenko–Bernstein–Rogozinskii ($K_6^2BR^2$)	1.7336	8.6977	1.0859	3.4753	−51.1199	2.2234	0.4201	0.2896
21	Rectangular	1.0000	50.0000	3.9224	3.9224	−13.2799	1.2128	1.0000	1.8466
22	*Triangular*	1.3333	25.0001	1.8242	3.0736	−26.5077	1.8191	0.5000	0.5407
23	*Gauss* $\alpha = 2$	1.2327	31.1469	2.1279	3.0366	−36.9155	1.6170	0.5981	0.4366
24	*Gauss* $\alpha = 2.5$	1.4456	19.3529	1.5802	3.1807	−43.2656	2.0213	0.4951	0.3256
25	*Gauss* $\alpha = 3$	1.7017	10.1829	1.1632	3.4721	−56.0922	2.2234	0.4166	0.2454
26	*Gauss* $\alpha = 3.5$	1.9765	4.6147	0.8702	3.8292	−71.0006	2.6276	0.3579	0.2413
27	*Hamming*	1.3638	23.3241	1.7492	3.0967	−45.9347	1.8191	0.5395	0.2996
28	*Blackman-Harris (four-termed)*	2.0044	3.7602	0.8256	3.8453	−92.0271	2.6276	0.3587	0.1656
29	*Natoll (four-termed)*	1.9761	4.1760	0.8506	3.8087	−97.8587	2.6276	0.3636	0.1475
30	*Dolph-Chebyshev* (α=*3.5*)	1.6328	11.8490	1.2344	3.3636	−70.0161	2.2234	0.4434	0.1174
31	*Bernstein–Rogozinskii*	1.2337	31.8309	2.0982	3.0103	−23.0101	1.6170	0.6366	0.6674
32	*Kaiser* $\alpha = 3$	1.7952	7.3534	1.0226	3.5639	−69.6568	2.4255	0.4025	0.1654

The performance functional $J(w)$ (Fig. 2.7) takes into account three the most important parameters: maximum transformation losses, maximum sidelobe level, and coherent gain. As is seen from Table 2, the Kravchenko–Dolph–Chebyshev (K_4C_3, $K_4C_{3.5}$) and Kravchenko–Bernstein–Rogozinskii ($K_4^2BR^2$) windows have the lowest levels of $J(w)$, because they possess low sidelobe levels, small transformation losses, and good coherent gain.

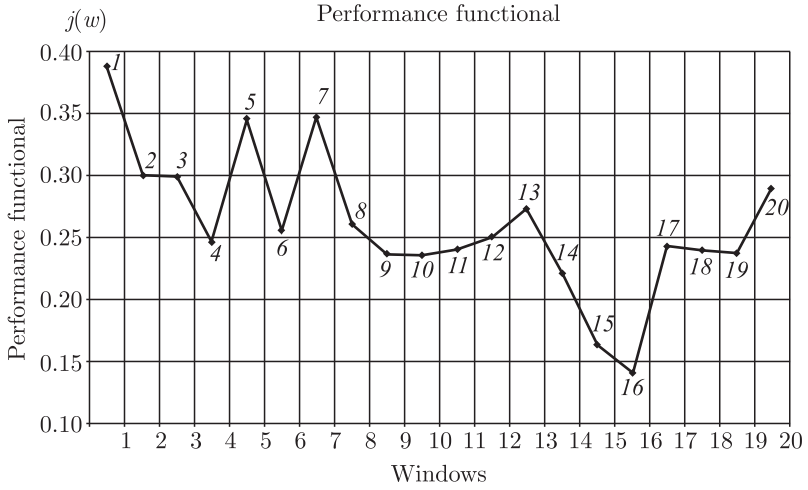


Fig. 2.7. Performance functional $J(w)$ for new synthesized Kravchenko windows. Minimum values correspond to the best windows. The windows are 1. K_2^4, 2. K_2G_2, 3. $K_2^2G_2$, 4. K_2G_3, 5. K_2BR^2, 6. K_2^2BR, 7. $K_2^2BR^2$, 8. K_4^4, 9. $K_4^2G_2^2$, 10. $K_4^4G_2$, 11. K_4G_3, 12. $K_4^2G_3$, 13. K_4^2BR, 14. $K_4^2BR^2$, 15. K_4C_3, 16. $K_4C_{3.5}$, 17. $K_6^2G_2^2$, 18. K_6G_3, 19. $K_6^2G_3$, 20. $K_6^2BR^2$.

2.9. Signal Filtration Using the New Windows

The following example shows the efficiency of using windows for detecting a small signal in the presence of an intensive closely located line. Let us study the amplitude–frequency response of the signal generated by the sum of two sinusoids: $x_1 = \sin(10 \cdot 2\pi \cdot x)$ and $x_2 = 0{,}01 \cdot \sin(16 \cdot 2\pi \cdot x)$. Since the frequency of each signal is multiplied by the number of bins in the DFT, both lines can easily be detected even when the rectangular window is used (Fig. 2.8 *a*). In this and next figures, axis x corresponds to the frequency in bins and axis y corresponds to the rate of decrease of signal's logarithmic amplitudes. Let us change the frequency of the more powerful signal, so that it will be placed between two neighboring bins. In this case, the picture sharply varies: the weak signal is hidden by the high sidelobes of the powerful signal (Fig. 2.8 *b*). Let us study changes in the spectra if the new synthesized Kravchenko windows are used. We will consider the behavior of the classical Hamming window (Fig. 2.9). As is seen from the LAFR, at the distance of 5.5 bins, the sidelobe level is -43 dB, which exceeds the powerful signal's sidelobe by 3 dB at the same frequency. Here, mutual signal suppression due to the phase opposition is observed along with a leakage of spectral components at positive and negative frequencies. Signals with the level lower than that of the powerful signal by 50 dB cannot be detected.

Consider the results of using Kravchenko windows K_2^4 and K_4^4 (Fig. 2.10). For the first of them (Fig. 2.10 *a*), the minimum of ~ 14 dB between two kernels is observed. However, an artifact appears at low frequencies (the right sidelobe of the window), which is similar to a signal with the amplitude of -55 dB at bin 7 of the DFT. As is seen, window K_4^4 (Fig. 2.10 *b*) has a minimum of ~ 16 dB and a relatively weak diffusion of the main lobe.

Consider the filtering properties of the Kravchenko–Gauss windows K_2G_2, $K_2^2G_2$, and K_2G_3 (Fig. 2.11). In this case, the first of them (Fig. 2.11 *a*) contains a false artifact with a level -60 dB at bin 7, although it provides the required level of signal recognition (~ 7 dB). The other windows (Figs. 2.11 *b*–*c*) ensure a very effective recognition (the minimum between signals is 11–20 dB). The results for the family of Kravchenko–Bernstein–Rogozinskii's windows (K_2BR^2, K_2^2BR, and $K_2^2BR^2$) are presented in Figs. 2.12 *a*–*c*. Here, the minimum between two peaks is within $\sim 12-20$ dB, which is a very good result. The window $K_2БР^2$ demonstrates the best quality of detection. It has a relatively narrow leakage band, and an artifact caused by a unity sidelobe has a level of -60 dB, which does not influence essentially the quality of recognition. Let us analyze the results of using the Kravchenko–Gauss windows ($K_4^2G_2^2$, $K_4^4G_2$, K_4G_3, and $K_4^2G_3$) in processing the signal under study (Fig. 2.13). Analysis of physical results shows that they provide detection of a weak signal with a minimum of 5–13 dB between peaks. It should be noted that these windows possess a small diffusion in the limits of the frequency band considered. Figure 2.14 shows the spectrum behavior after applying the new family of the Kravchenko–Bernstein–Rogozinskii windows (K_4^2BR and $K_4^2BR^2$). The second signal is clearly distinguishable from the first one on these plots. For the window K_4^2BR (Fig. 2.14 *a*), an artifact exists. It is caused by the presence of sidelobes at the level of -53 dB. Figure 2.15 presents the results of applying the Kravchenko–Chebyshev windows. Note that, in both cases, the second signal is clearly distinguishable and the minimum between two peaks is equal to 15–20 dB. At the same time, on Fig. 2.15 *a*, an artifact caused by a high sidelobe level is observed to the left from the first kernel. The results for the Kravchenko–Bernstein–Rogozinskii window $K_6^2BR^2$ are presented in Fig. 2.16. The window ensures recognition of a signal with a minimum between peaks of ~ 16 dB. The Kravchenko–Gauss windows (Fig. 2.17) also provide a weak signal recognition, and window K_6G_3 has small diffusion of spectral components, its minimum being ~ 15 dB.

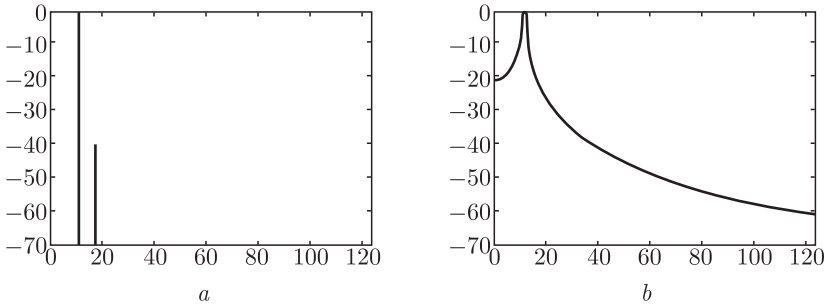


Fig. 2.8. Powerful signals with frequencies (*a*) 10 bin and (*b*) 10.5 bin.

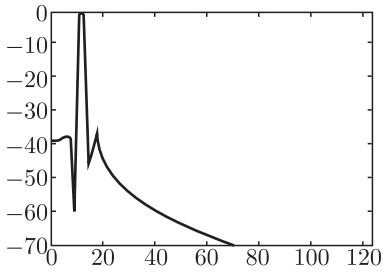


Fig. 2.9. The Hamming window.

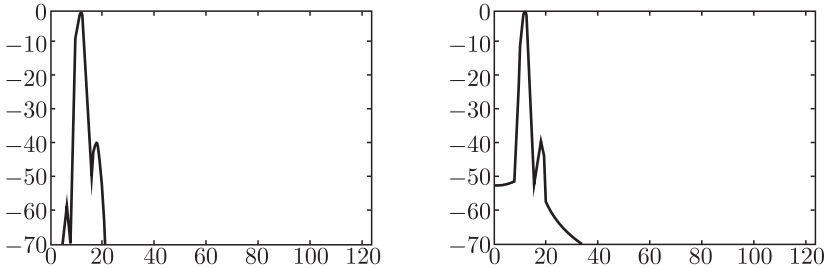


Fig. 2.10. The Kravchenko windows (*a*) K_2^4 and (*b*) K_4^4.

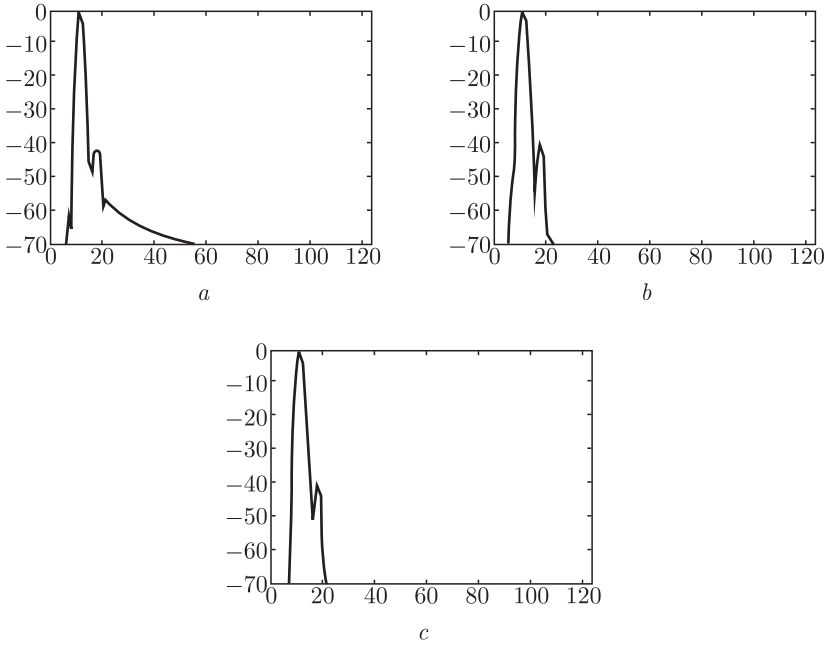


Fig. 2.11. The Kravchenko–Gauss windows (*a*) K_2G_2, (*b*) $K_2^2G_2$, and (*c*) K_2G_3.

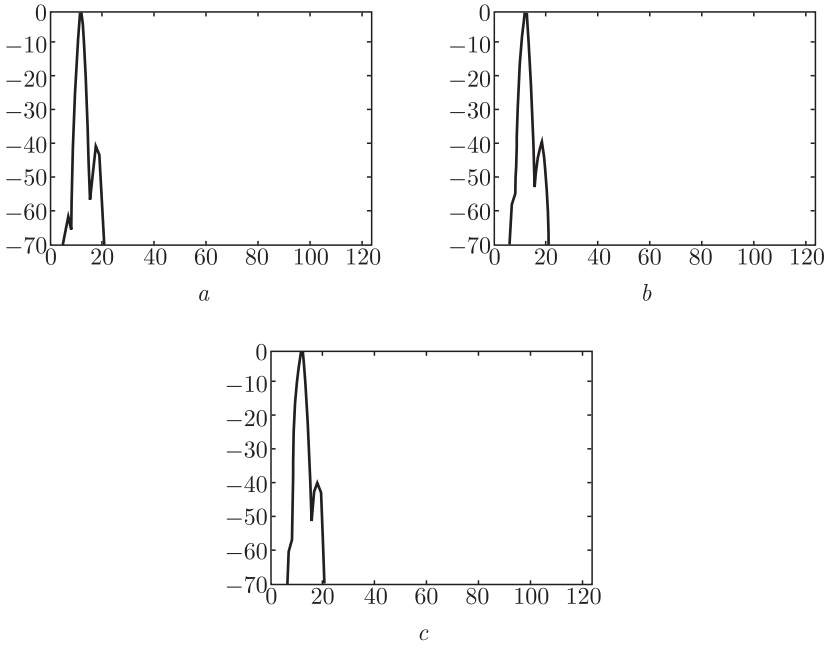


Fig. 2.12. The Kravchenko–Bernstein–Rogozinskii windows (*a*) K_2BR^2, (*b*) K_2^2BR, and (*c*) $K_2^2BR^2$.

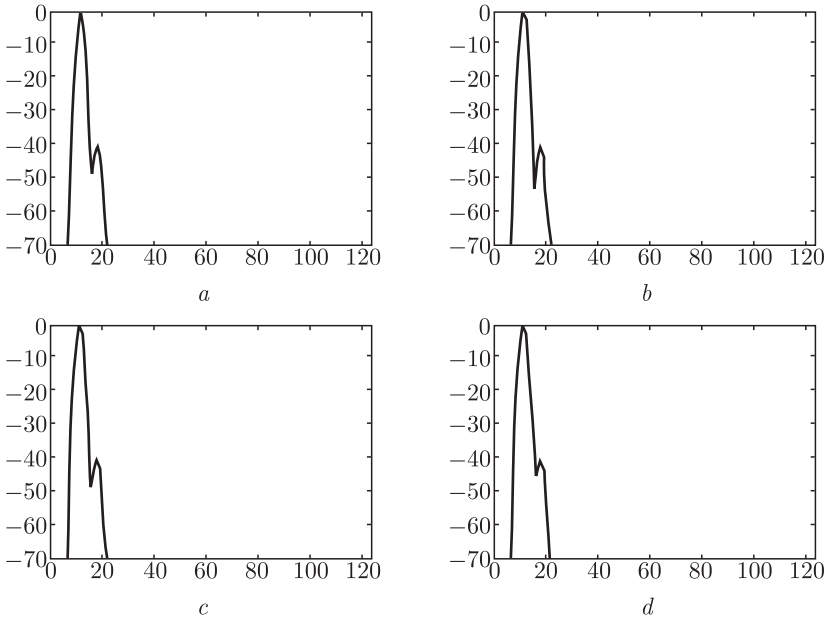


Fig. 2.13. The Kravchenko–Gauss windows (*a*) $K_4^2G_2^2$, (*b*) $K_4^4G_2$, (*c*) K_4G_3, and (*d*) $K_4^2G_3$.

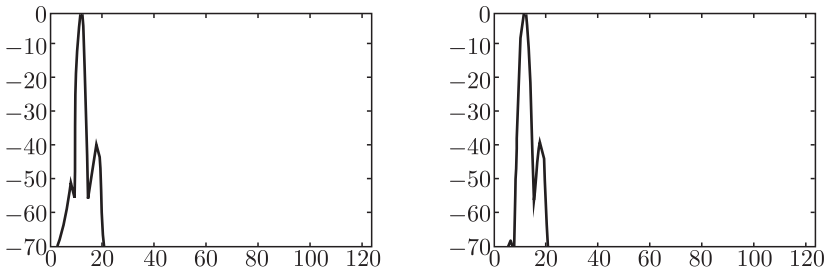


Fig. 2.14. The Kravchenko–Bernstein–Rogozinskii windows (a) K_4^2BR and (b) $K_4^2BR^2$.

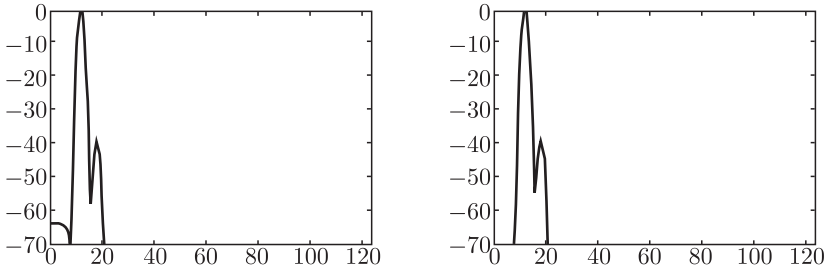


Fig. 2.15. The Kravchenko–Dolph–Chebyshev windows (a) K_4C_3 and (b) $K_4C_{3.5}$.

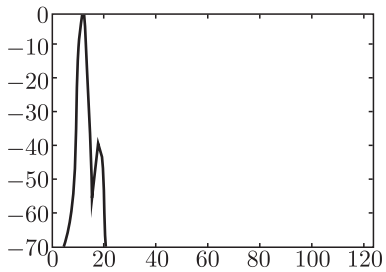


Fig. 2.16. The Kravchenko–Bernstein–Rogozinskii window $K_6^2BR^2$.

2.10. Time-Domain Dilation of the New Synthesized Kravchenko Windows

Figure 2.18 demonstrates the influence of the window's time dilation $w^*(t) = w(k \cdot t)$ on sidelobe levels and maximum transformation losses. The time dilation corresponds to shortening the interval of definition for the window $[a^*; b^*] = \frac{1}{k}[a; b]$, where k is the scale factor and $[a; b]$ is the interval of definition for window $w(t)$. Here, the sidelobe level increases and the transformation loss decreases proportionally. The same situation is observed for other windows.

Numerical experiments and physical analysis of results have shown that the parameters of the new synthesized Kravchenko, Kravchenko–Gauss, Kravchenko–Bernstein–Rogozinskii, and Kravchenko–Dolph–Chebyshev windows are comparable with those of the classical windows and some of them are even better. These results are fundamental

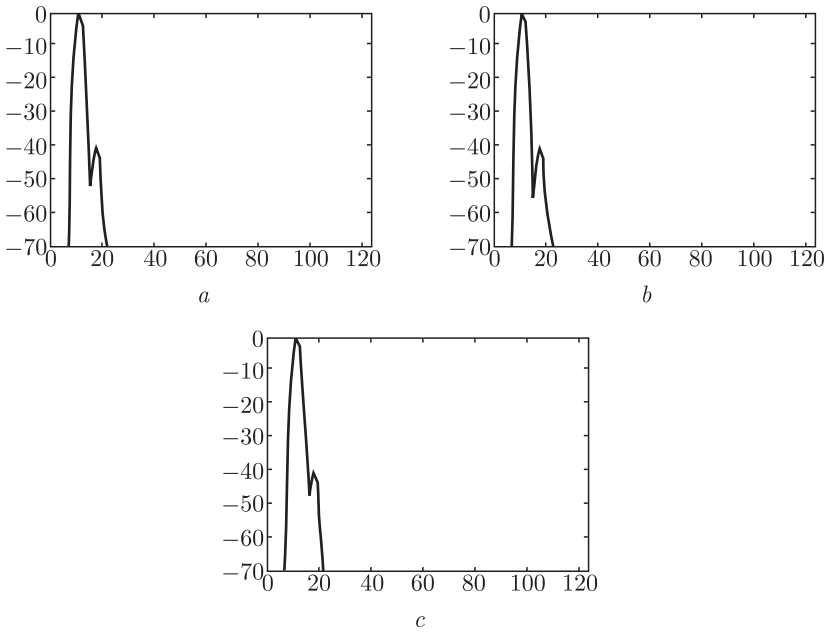


Fig. 2.17. The Kravchenko–Gauss windows (*a*) $K_6^2G_2^2$, (*b*) K_6G_3, and (*c*) $K_6^2G_3$.

for the realization of digital spectral processing of multivariate signals in the Doppler radar, synthetic-aperture radar, in problems of signal resolution and compression, computer tomography and termography, and medical diagnostics.

Other examples of application of the new Kravchenko windows for detection of numerous targets by radar are presented in the second part of this book.

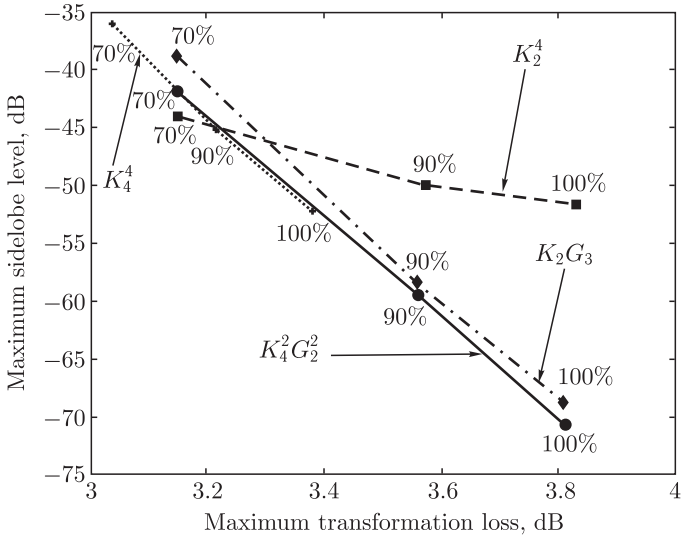


Fig. 2.18. Maximum sidelobe level vs. maximum transformation loss for the K_2^4, K_2G_3, K_4^4, and $K_4^2G_2^2$ windows with the support length reduced by 10 and 30%.

References

1. *Kravchenko*, *V.F.* New Synthesized Windows. DAN RAN. 2002, vol. 382, no. 2, pp. 190–198.
2. *Zelkin*, *Ye.G. and Kravchenko*, *V.F.* Atomic Functions in Antenna Synthesis Problems and New Synthesized Windows. J. Commun. Technol. El. 2001, vol. 46, no. 8, pp. 829–857.
3. *Basarab*, *M.A.*, *Kravchenko*, *V.F.*, *and Masyuk*, *V.M.* R-Functions, Atomic Functions and Their Applications. Zarubezhnaya Radioelektronika. Uspekhi Sovremennoi Radioelektroniki, 2001, no. 8, pp. 5–40.
4. *Basarab*, *M.A. and Kravchenko*, *V.F.* R-Functions and Atomic Functions in Problems of Digital Signal Processing of Multivariate Signals. Telecommunications and Radio Engineering, 2001, vol. 56, no. 4–5, pp. 4–44.
5. *Marple*, *Jr.*, *S.L.* Tsifrovoi spektralnyi analiz i ego prilozheniya (Digital Spectral Analysis and Its Applications). Moscow; Mir, 1990.
6. *Harris*, *F.J.* On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. Proc. of the IEEE, 1978, vol. 66, no. 1, pp. 51–83.
7. *Kravchenko*, *V.F.*, *Basarab*, *M.A.*, *Pustovoit*, *V.I.*, *and Perez-Meana*, *H.* New Constructions of Weighting Windows on the Base of Atomic Functions in Problems of Speech Synthesis. DAN RAN, 2001, vol. 377, no. 2, pp. 183–189.
8. *Kravchenko*, *V.F.*, *Basarab*, *M.A.*, *and Perez-Meana*, *H.* Atomic Functions and the Construction of New Windows in Problems of Digital Speech Analysis and Modeling. Telecommunications and Radio Engineering, 2001, vol. 56, no. 1, pp. 3–17.
9. *Zelkin*, *Ye.G. and Kravchenko*, *V.F.* Atomic Functions in Antenna Synthesis Problems and New Synthesized Windows. J. Commun. Technol. El., 2001, vol. 46, no. 8, pp. 829–857.
10. *Marple*, *Jr.*, *S.L.* Digital Spectral Analysis and Its Applications. Moscow: Mir, 1990.
11. *Harris*, *F.J.* On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. Proc. of the IEEE. 1978, vol. 66, no. 1, pp. 51–83.

Chapter 3

SYNTHESIS OF DIGITAL FILTERS ON THE BASIS OF THE ATOMIC FUNCTIONS AND ITS APPLICATIONS

3.1. Synthesis of Digital Filters

3.1.1. Finite Impulse Response (FIR) filters. The atomic windows enable one to improve the frequency response of digital FIR. Without loss of generality, let us consider the synthesis of a low-pass filter (LPF) [1–3] whose ideal frequency response is

$$H_0(\omega) = \begin{cases} 1, & |\omega| < \omega_c, \\ 0, & \omega_c < |\omega| < \pi, \end{cases}$$

where ω_c is the cutoff frequency. This response is unrealizable. The frequency response of the FIR synthesized using the rectangular window has considerable ripples in the vicinity of the cutoff frequency, the so-called Gibbs oscillations. This frequency response can be smoothed using a window different from a rectangular one. In this case, the FIR of the ideal filter is multiplied by a window weighting function. We compare FIRs based on different windows using the following parameters: the *passband ripple* R (in dB) and the *stopband attenuation* A (in dB). Figure 3.1 and Table 3.1 [4, 5] summarize the results obtained for the classical and atomic windows. The cutoff, passband, and stopband frequencies are assumed to be equal to $\pi/2$, 1.27, and 1.87, respectively.

Only the Hamming window exhibits a characteristic better than that of the atomic windows under consideration. A FIR based on the window $w_2(x)$ has the characteristics comparable with the filters based on the Hamming and Gaussian windows.

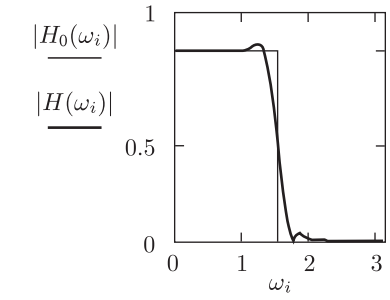
3.1.2. Infinite Impulse Response (IIR) filters. Consider the use of the AFs for the synthesis of IIR filters. Here, we briefly describe this algorithm [2]. As is known, the frequency response of an analog RF is

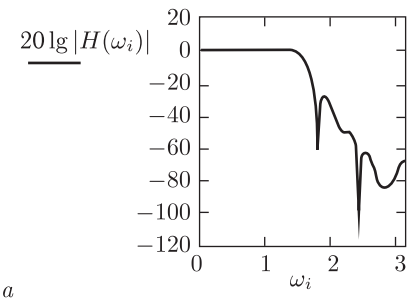
$$H(p) = \frac{A(p)}{B(p)} = \frac{\sum\limits_{k=0}^{K-1} a_k p^k}{\sum\limits_{k=0}^{M-1} b_k p^k}, \tag{3.1}$$

where $p = j\omega$ and ω is the circular frequency. Expression (3.1) corresponds to a linear continuous structure described by a linear differential equation with constant coefficients

$$\sum_{k=0}^{M-1} b_k y^{(k)}(t) = \sum_{k=0}^{K-1} a_k x^{(k)}(t). \tag{3.2}$$

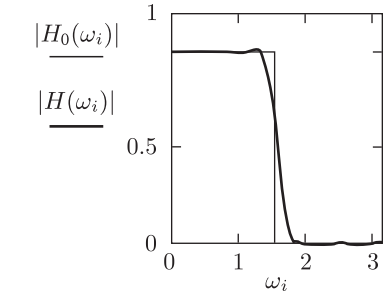
$w(x)$=up(x)

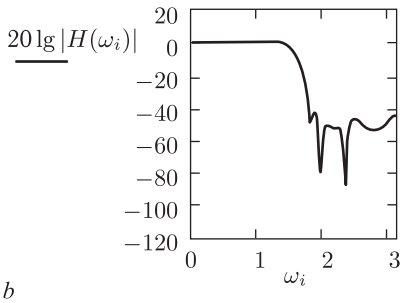




a

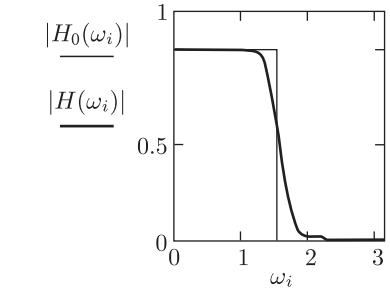
$w(x)$=up(x)+0.01up$''(x)$

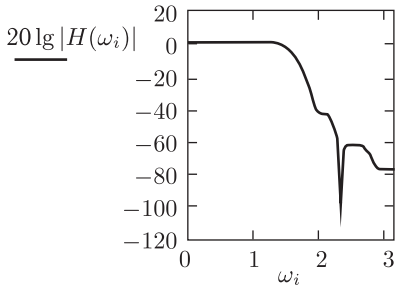




b

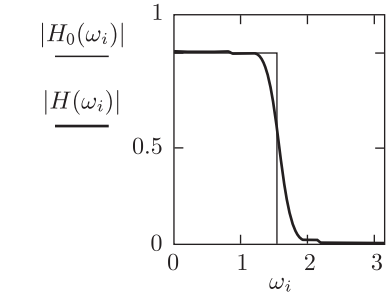
$w(x) = \text{fup}_1(3x/2)/\text{fup}_1(0)$

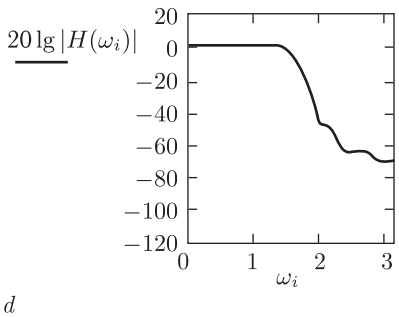




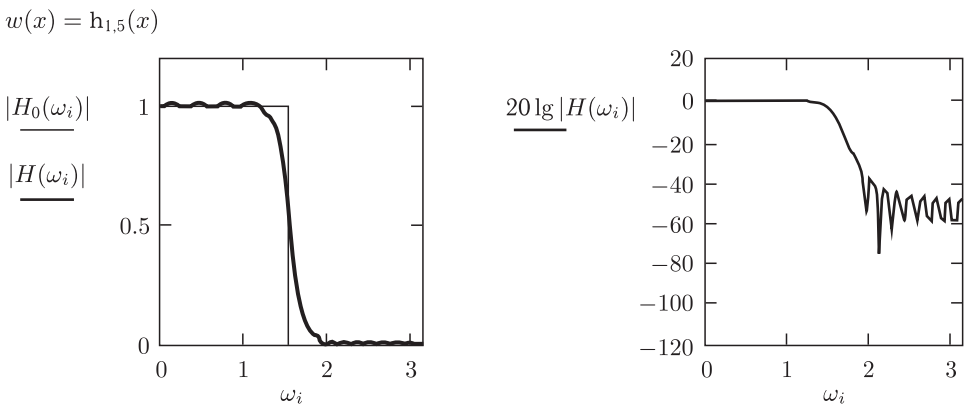
c

$w(x) = (\text{fup}_1(3x/2) + 0.0036\text{fup}_1''(3x/2))/(\text{fup}_1(0) + 0.0036\text{fup}_1''(0))$

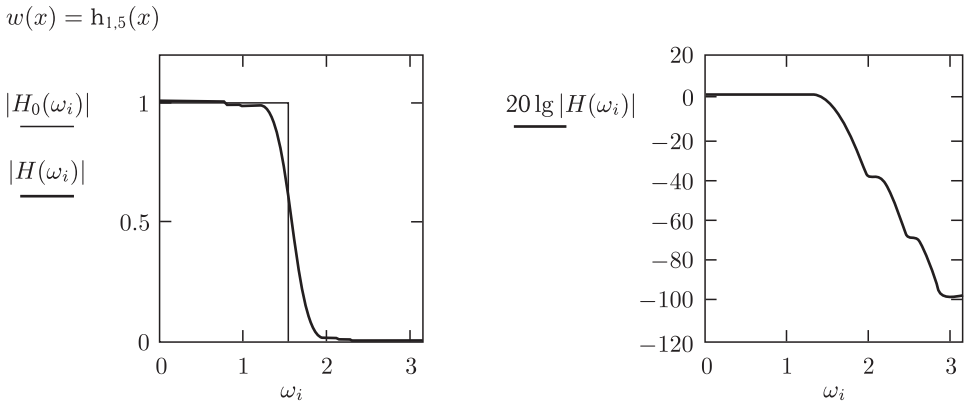




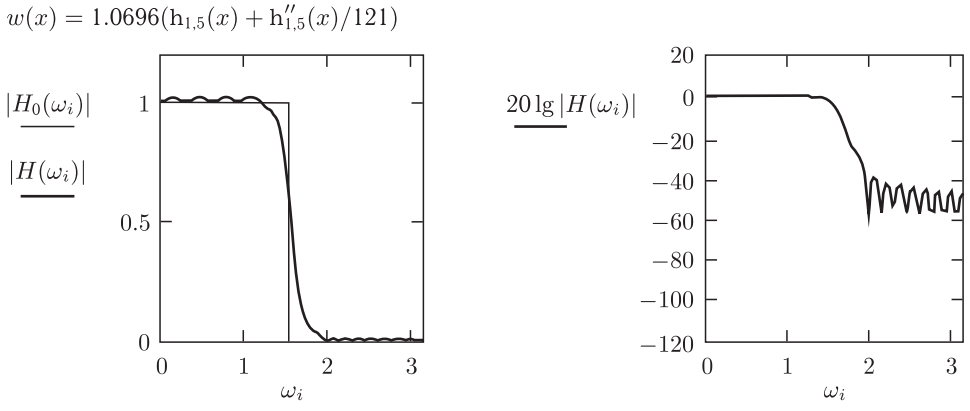
d



e



f



g

Fig. 3.1. Gain characteristics (left) and logarithmic gain characteristics (right) of the ideal FIR ($H_0(\omega)$ and the LPFs based on the Kravchenko–Rvachev windows $w_1 - w_7$ ($H(\omega)$).

Table 3.1. Parameters of FIRs based on the known and Kravchenko–Rvachev windows.

Windows	Parameters, dB	
	R	A
Well-known		
Rectangular	−0.42	−26.55
Bartlett	−0.44	−26.11
Hamming	0.016	−56.24
Hanning	0.05	−44.54
Blackman	−0.13	−36.41
Kaiser–Bessel ($\beta = 3$)	−0.18	−33.68
Gauss ($\alpha = 6.25$)	−0.08	−41.04
Kravchenko–Rvachev		
$w_1(x)$	0.32	−28.51
$w_2(x)$	0.05	−44.45
$w_3(x)$	−0.23	−31.48
$w_4(x)$	−0.19	−32.95
$w_5(x)$	−0.32	−28.61
$w_6(x)$	−0.27	−30.3
$w_7(x)$	−0.28	−29.78

This equation relates the output, $y(t)$, and input, $x(t)$, signals. The synthesis of a digital IIR consists of constructing the transfer function

$$H(z) = \frac{C(z)}{D(z)} = \frac{\sum\limits_{k=0}^{R-1} c_k z^{-k}}{1 + \sum\limits_{k=1}^{Q-1} d_k z^{-k}} \tag{3.3}$$

corresponding to the linear difference equation

$$y[n] + \sum_{k=1}^{Q-1} d_k y[n-k] = \sum_{k=0}^{R-1} c_k x[n-k]. \tag{3.4}$$

Here, $z = \exp(j2\pi f/f_0)$, f is the frequency, f_0 is the sampling frequency, and $x[n]$ and $y[n]$ are the samples of the input and output sequences. The algorithm involves a discrete approximation of the derivatives providing a passage from (3.2) to (3.4). Within the framework of the theory of finite difference schemes, the kth derivative of a discrete

function is approximately estimated from the $(k+1)$-th sample of the function. For example,

$$\begin{aligned} \mathrm{dif}_1(x) &= x[n] - x[n-1], \\ \mathrm{dif}_2(x) &= x[n+1] - 2x[n] + x[n-1], \end{aligned} \tag{3.4a}$$

In other words, the estimation is performed using a $(k+1)$-th order IIR with the frequency response

$$H_k(j\omega) \sim \sin^k(\omega/f_0).$$

Hence, difference filters are formed by multiplying the ideal frequency response $j\omega$ and the Dirichlet frequency window

$$W_k(j\omega) = \left[\frac{\sin(\omega/f_0)}{\omega/f_0}\right]^k.$$

The rectangular window is not the best one, because it provides reasonable characteristics only for the estimation of high-order derivatives. Therefore, it is necessary to use the approximations of derivatives alternative to (3.4a).

Taking into account the properties of the Dirac function, we can represent the left-hand side of (3.2) as the convolution

$$\sum_{k=0}^{M-1} b_k y^{(k)}(t) = y(t) * \sum_{k=0}^{M-1} b_k \delta^{(k)}(t), \tag{3.5}$$

where q is the sign of convolution. Since the derivatives of the Dirac function do not form a realizable sequence of functions, this function should be replaced by a suitable finite function $h(t)$ with the spectrum well localized within the interval $[-f_0/2,\ f_0/2]$. These requirements are satisfied by the AFs whose derivatives can easily be estimated using the equation $Ly(x) = \sum\limits_{m=1}^{M} c_m y\ (ax - b_m)$. Instead of (3.5), we obtain

$$\sum_{k=0}^{M-1} b_k y^{(k)}(t) \approx y(t) * \sum_{k=0}^{M-1} b_k h^{(k)}(t),$$

or, in the discrete form,

$$\sum_{k=0}^{M-1} b_k y^{(k)}(t) \approx \sum_{i=0}^{N-1} \left\{ \sum_{k=0}^{M-1} b_k h^{(k)}[i] \right\} y[n-i] = \sum_{i=0}^{N-1} \left\{ \sum_{k=0}^{M-1} b_k h^{(k)}[i] \right\} z^{-i}. \tag{3.6}$$

In a similar manner, one can modify the right-hand side of (3.2). The problem is to choose an optimum digitization interval $\Delta \leqslant f_0^{-1}$ so as to provide the approximation of derivatives with a sufficient accuracy when the order of the filter is not very high. We illustrate the algorithm by the following example.

Suppose that it is necessary to determine the coefficients of a digital recursive LPF with periodic passband ripple and the following parameters: the maximum passband nonuniformity of $R = 0.1$ dB, the maximum stopband attenuation of $A = 30$ dB, the sampling frequency $f_0 = 12$ kHz, the limiting passband frequency of $f_1 = 1$ kHz, and the limiting stopband frequency of $f_2 = 4$ kHz. We choose the Chebyshev filter as an analog prototype filter. According to the standard technique, we find the necessary order $(n = 3)$ and the transfer function of the analog filter

$$H(p) = 1.638 \cdot \frac{1}{(0.969 + p)(1.690 + 0.969p + p^2)}.$$

Thus, we obtain the following coefficients in (3.5): $b_0 = 1.638$, $a_0 = 1.638$, $a_1 = 2.629$, $a_2 = 1.938$, and $a_3 = 1$. Applying a bilinear transformation to replace p by z, we find the transfer function of the recursive LPF:

$$H_1(z) = 0.018 \cdot \frac{(1+z^{-1})^3}{(1-0.588z^{-1})(1-1.273z^{-1}+0.624z^{-2})}.$$

Now, let us synthesize a filter using the AFs. We set $h(t) = \mathrm{fup}_3(t)$ and use a four-point approximation of derivatives. According to (3.6), we obtain the following transfer function of the third-order recursive Chebyshev filter:

$$H_2(z) = \frac{\sum\limits_{k=0}^{3}\left\{\sum\limits_{i=0}^{3} b_i 2^i fup_3^{(i)}\left(-\frac{3}{2}+k\right)\right\}z^{-k}}{\sum\limits_{k=1}^{3}\left\{\sum\limits_{i=0}^{3} a_i 2^i fup_3^{(i)}\left(-\frac{3}{2}+k\right)\right\}z^{-k}} = \\ = \frac{5.686\cdot 10^{-3} + 0.097z^{-1} + 0.097z^{-2} + 5.686\cdot 10^{-3}z^{-3}}{1.581 - 3.004z^{-1} + 2.229z^{-2} - 0.601z^{-3}}$$

with $b_1 = b_2 = b_3 = 0$. Figure 3.2 demonstrates the frequency responses obtained for $H_1(z)$ and $H_2(z)$.

As compared to known frequency-conversion techniques, the AF method proposed for the synthesis of digital filters provides a simple computational algorithm, which is easy to implement.

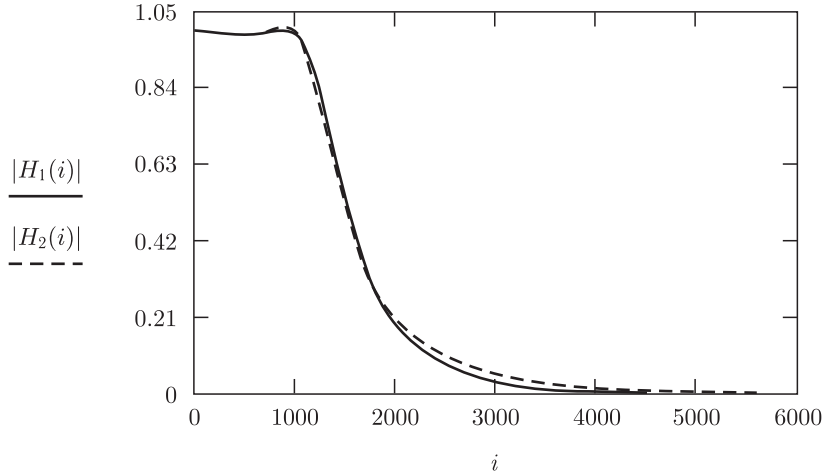


Fig. 3.2. Frequency responses of the digital Chebyshev LPF synthesized using (solid line) the bilinear transformation (H_1) and (dashed line) AFs (H_2).

3.2. Kravchenko–Rvachev Windows Used in Digital Radar

Frequency discriminators (FDs) refer to the basic components of digital range and speed meters. After every probing session, the information about the lag of the received signal and the Doppler frequency is stored in the memory unit. After that, the fast Fourier transform (FFT) is applied to the data corresponding to a single range interval.

Then, this procedure is performed for other range intervals, resulting in the spectrum of the received signal. As applied to the automatic tracking system, the M-point FFT can be treated as an algorithm of a multifilter spectrum analyzer with M frequency channels called FFT filters [3, 5, 6].

For the FD algorithms, the initial data consist of complex amplitudes $U\overset{*}{F}_p$ formed by the FFT operation. Here, U is the amplitude of a complex digital signal and $\overset{*}{F}_p$ is the complex amplitude-frequency characteristic of the pth FFT filter. The processing system is a linear or a square-law detector, depending on whether the absolute values of amplitudes $U\overset{*}{F}_p$ are used. When the weighting summation algorithm is applied, the discrimination characteristic (DC) of an FD with a linear detector is

$$u(f) = \frac{\sum\limits_{i=1}^{r} (2i - (r+1))\, F_{p(i)}(f)}{\sum\limits_{i=1}^{r} F_{(i)}(f)}, \tag{3.7}$$

where r is the number of the FD filters. For an FD with a square-law detector, one must replace $F_{p(i)}(f)$ by $F^2_{p(i)}(f)$. The absolute values of the amplitude–frequency characteristics of the FFT filters are

$$F_{p(i)}(f) = M \left| \frac{\sin\left[\pi\left(MfT_d - p(i) + 1\right)\right]}{\pi\left(MfT_d - p(i) + 1\right)} \right|,$$

where the number of a filter is $p(i) = \frac{1}{2}\left(M - r(\nu - 1)\right) + i, \quad \nu = 1 - r \bmod 2$, and $T_d^{-1} = f_d$ is the sampling frequency. The exact value f_0 of the transient frequency of the input signal at which the FD DC vanishes is given by the formula $f_0 = \frac{M - \nu}{2M} f_d$. It is conventional to represent the DC in terms of relative frequency offsets f_α with respect to the transient frequency $f_\alpha = M\left(2\frac{f}{f_d} - 1\right) + \nu$.

In this case,

$$F_{p(i)}(f_\alpha) = M \left| \frac{\sin\left[0{,}5\pi\left(f_\alpha + r + 1 - 2i\right)\right]}{0{,}5\pi\left(f_\alpha + r + 1 - 2i\right)} \right|. \tag{3.8}$$

The substitution of (3.8) into (3.7) yields the relative DC $u(f_\alpha)$. Formula (3.8) corresponds to the FFT performed with the implicit use of the rectangular Dirichlet window. Due to the high level of sidelobes in the spectrum of this window, the FD DC is substantially irregular, which causes errors in direct frequency measurements. Therefore, it is expedient to use other weighting windows, in particular, atomic windows, whose advantage is that their Fourier transforms can be calculated by simple explicit formulas. Figures 3.3 *a*, 3.3 *b*, and 3.3 *c* – 3.3 *f* display the relative DCs (r=6) for the rectangular, Hamming, and Kravchenko–Rvachev (w_1, w_3, w_5, and w_7) windows, respectively [7, 8]. The DCs for the windows w_2, w_4, and w_6 are not shown in the figures, because they are virtually identical to those for w_1, w_3, and w_5, respectively. One can see that the DC is smoothed and the maximum smoothness is observed for the window w_7. In addition, the weighting windows increase the linear section of the characteristic, because this section corresponds to the frequency band of the main lobes of the amplitude–frequency characteristic of an FFT filter. The best result is provided by

w_5. At the same time, the weighting windows different from a rectangular one slightly decrease the signal-to-noise ratio, which is due to the expansion of the main lobe of the window spectrum. Figures 3.4 *a*–3.4 *f* illustrate the use of the same windows in an FD with a square-law detector and with the same number of filters.

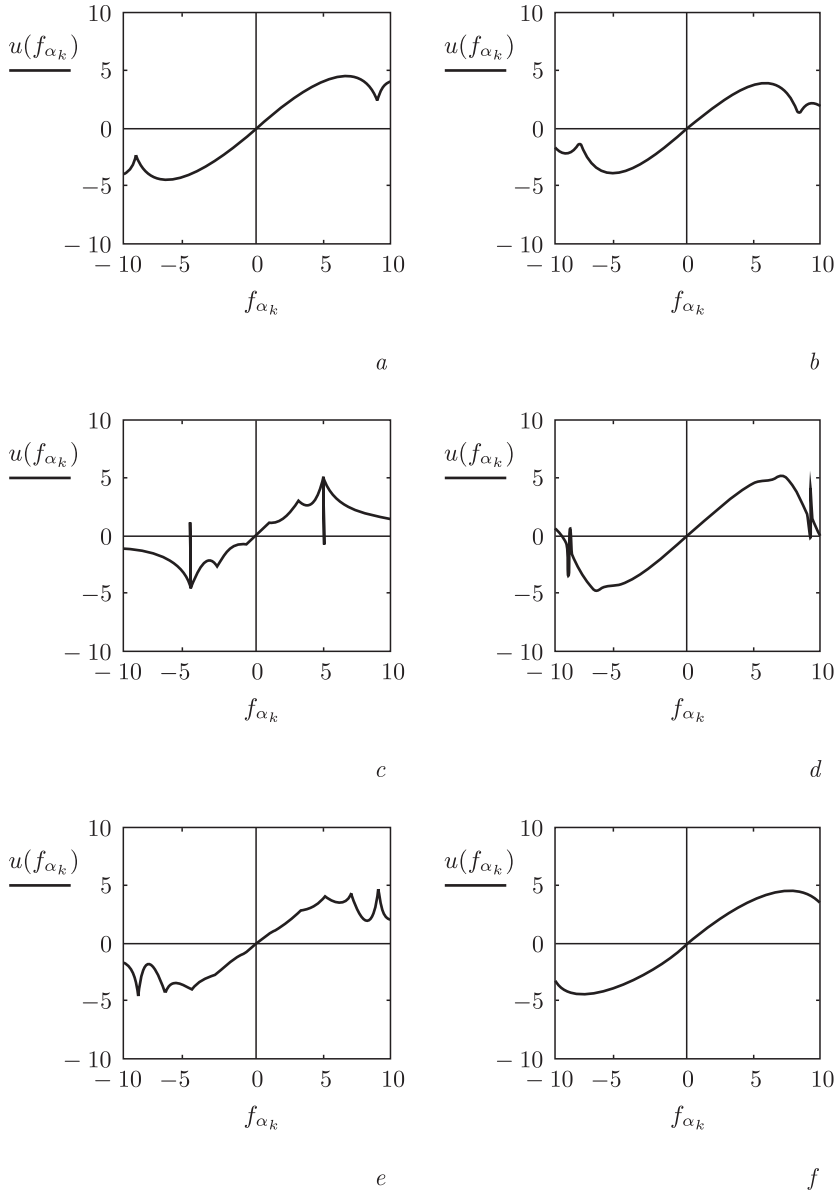


Fig. 3.3. Relative DC of an FD with a linear detector for the FFT with (*a*) rectangular, (*b*) Hamming, and Kravchenko–Rvachev windows (*c*) w_1, (*d*) w_3, (*e*) w_5, and (*f*) w_7.

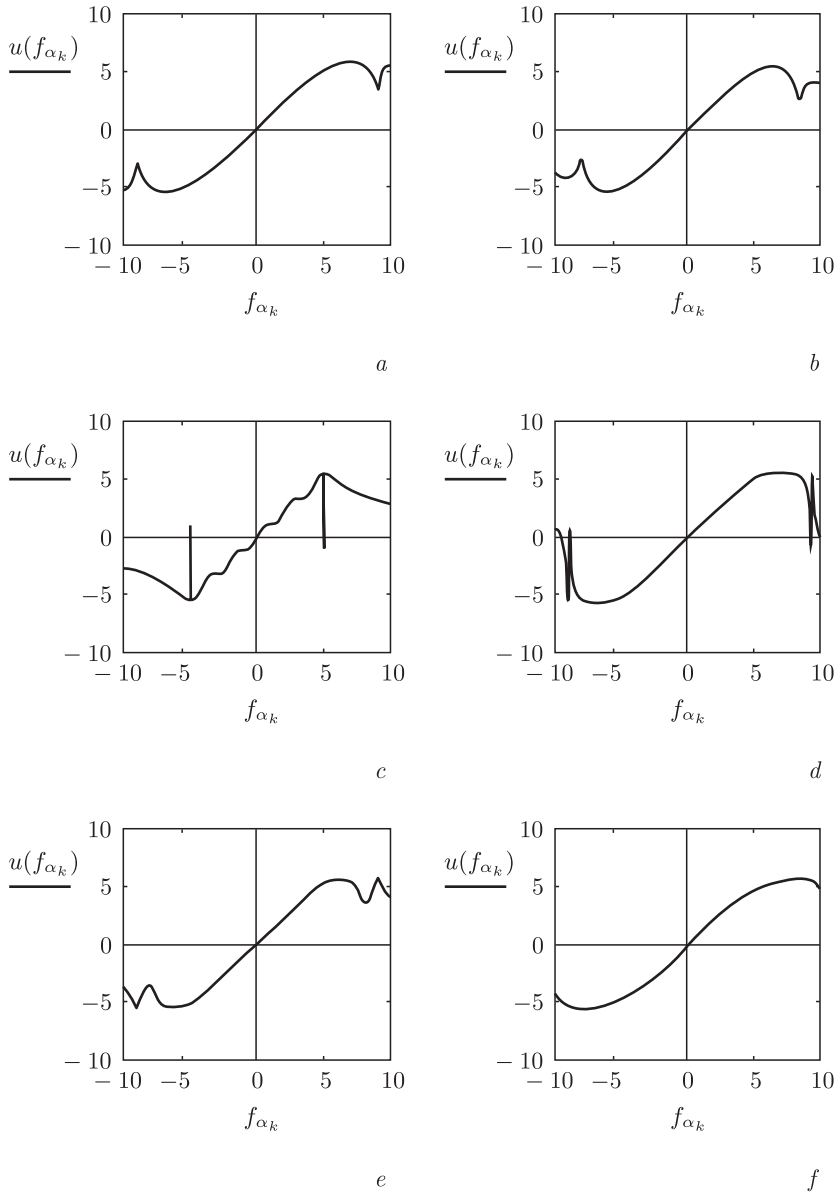


Fig. 3.4. Relative DC of an FD with a square-law detector for the FFT with (*a*) rectangular, (*b*) Hamming, and Kravchenko–Rvachev windows (*c*) w_1, (*d*) w_3, (*e*) w_5, and (*f*) w_7.

3.3. The Use of the New Types of Windows in Electroencephalography

Electroencephalograms (EEG) allow one to obtain the picture of the brain activity. EEG analysis is used for separation of specific types of electric potentials and determining their localization in brain. During the clinical description of the EEG readings,

the specific morphological components are distinguished, namely, rhythms, criteria of epileptimorphic activity, and temporal and topography parameters [9, 10].

Rhythms. The EEG rhythm is a definite type of electric activity corresponding to the specific brain state and connected with cerebral mechanisms. Usually, the following four types of rhythms of increasing frequency are used in clinical investigations: delta-, theta-, alpha-, and beta-rhythms. *Alpha-rhythm* has the frequency of 8–13 Hz and amplitude up to 100 μV and is basic for preliminary detection of deviations from the normal state. It is registered for 85–95% of relaxed adults with closed eyes. *Beta-rhythm* has the frequency of 14–40 Hz and amplitude up to 15 μV and is a leading rhythm of the active state. Beta-rhythm is connected with somatic, sensor, and motor brain mechanisms. It gives the response on moving activity or tactile stimulation. Often, two ranges of beta-rhythms are distinguished, β_1 and β_2, with the frequencies of 14–18 Hz and 18–40 Hz, respectively. Usually, beta-rhythm is weak (3–7 μV) and is masked by noise and electromiograms (EMG). Slow rhythms, such as *theta-rhythm* with the frequency of 4–6 Hz and *delta-rhythm* with the frequency of 0.5-3 Hz have the amplitudes of 40–300 μV and, in normal stage, are typical for some sleep stages.

To investigate the use of the new weighting functions (windows), the following signals were used: the standard signal $x_e(t)$ with the sampling frequency of 800 Hz, and test signals $x(t)$ and $y(t)$ with the sampling frequency of 100 Hz. The interval of measurements was equal to 0.5 s. In this case, the Gibbs effect of the power leakage is observed, resulting in the increase of the main lobe widths and origination of the parasitic periodic components of the form $\sin(x)/x$ for each spectral peak. This effect is caused by the finite realization of a signal obtained by the use of a rectangular window.

EEG parameters. Using the DFT, expand original signals $x(t)$, $y(t)$, $x_e(t)$ into the Fourier series

$$DF_x(t) = a_0 + \sum\nolimits_{i=1}^{(n-1)/2} (a_i \cos(i \cdot t) + b_i \sin(i \cdot t)).$$

The amplitude of a signal is determined as $A(\omega_i) = \sqrt{a_i^2 + b_i^2}$, and the phase-frequency characteristic is calculated by the formula $\varphi(\omega_i) = \operatorname{arctg}\left(-\frac{b_i}{a_i}\right)$. Values of the signal average amplitude in the beta-range, $A_x^\beta = \frac{1}{n} \sum\limits_{\omega=\omega_\alpha}^{\omega_\beta} A_x(\omega_i)$ are presented in Table 3.2.

To weaken the influence of the Gibbs effect, instead of the average value of the amplitude, one can use its maximum value in the beta-range, determined as $A^\beta_{x\max} = \max\limits_{\omega_i \in [\omega_\alpha, \omega_\beta]} A_x(\omega_i)$.

Figures 3.5 and 3.6 illustrate the difference between the analyzed EEG signal parameters ((a) the standard signal with frequency 800 Hz, (b) the test signal $x(t)$, and (c) the test signal $y(t)$) for the cases when the rectangular window and the KKB_1 are used. Figures 3.7 and 3.8 present frequency responses of the investigated processes, namely, the spectra of the analyzed EEG signals (the standard signal with a frequency of 800 Hz and the test signal $x(t)$) and cross-spectra of the processes $x(t)$ and $y(t)$.

Only some of the problems of digital signal processing on the basis of the AFs have been considered. It should be noted that the new mathematical methods can be applied effectively for processing of large arrays of digital information, for example, in such areas as synthetic-aperture radar, television, radioastronomy, telemedicine, computer thermography, etc. Considerable promises are offered by the perfection of digital processing techniques using the wavelet analysis on the basis of the AFs.

Table 3.2. The ratio between average amplitudes of the beta-rhythm for signals $x(t)$ and $x_e(t)$.

Windows	$A^{\beta}_{x_e}$ for $x_e(t)$	A^{β}_{x} for $x(t)$	$A^{\beta}_{x}/A^{\beta}_{x_e}$
Rectangular	0.4521	0.7112	1.5805
Kravchenko–Bessel ($\nu = 0$)	0.2283	0.3933	1.7495
Kravchenko–Rvachev (KR)	0.2294	0.3988	1.7669
Bernstein–Rogozinskii	0.2722	0.4866	1.8043
Hamming	0.2123	0.3853	1.8486
Kaiser	0.2036	0.3791	1.8918
Kravchenko–Bessel of the 1-st kind ($\nu = 0$) (m)	0.2279	0.4457	1.9830
Kravchenko–Hamming (KH)	0.2072	0.4078	1.9925
Kravchenko–Kaiser (KK)	0.2166	0.4301	1.9972
Kravchenko–Hamming (KH_1)	0.2221	0.4446	2.0091
Kravchenko-Kaiser (KK_2)	0.2413	0.4838	2.0130
Kravchenko (K_3)	0.2395	0.4811	2.0161
Kravchenko (K_1)	0.2118	0.4240	2.0183
Kravchenko–Hamming (KH_2)	0.2348	0.4728	2.0208
Kravchenko (K_2)	0.2271	0.4584	2.0240
Kravchenko–Kaiser (KK_1)	0.2292	0.4627	2.0248
Kravchenko–Bessel of the 1-st kind ($\nu = 0.5$) (m)	0.3187	0.4179	1.3303
Кравченко–Bessel of the 1-st kind ($\nu = 1$) (m)	0.1740	0.2518	1.4479
Кравченко–Bessel of the 1-st kind ($\nu = 0.5$)	0.3776	0.5456	1.4483
Кравченко–Bessel of the 1-st kind ($\nu = 1$)	0.4232	0.6763	1.5993

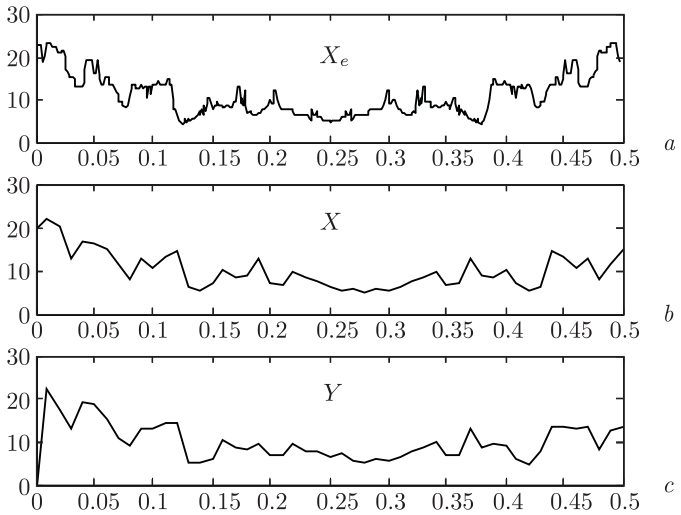


Fig. 3.5. EEG signals processed by the rectangular window: (a) the standard signal with frequency 800 Hz, (b) the test signal $x(t)$, and (c) the test signal $y(t)$.

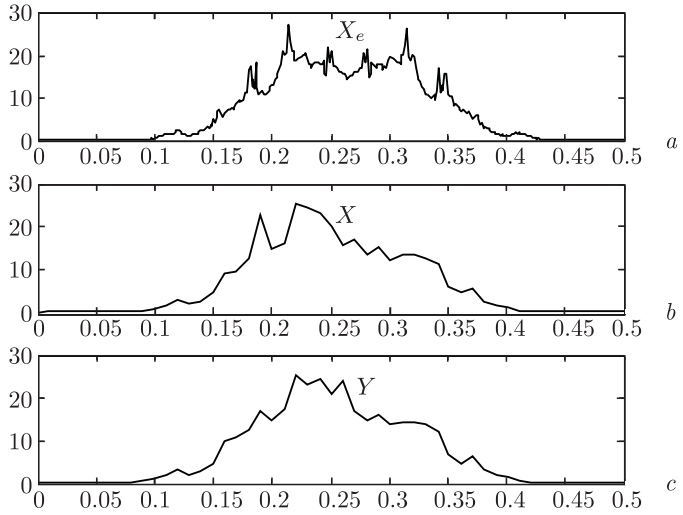


Fig. 3.6. EEG signals processed by the Kravchenko window KKB_1: (*a*) standard signal with frequency 800 Hz, (*b*) test signal $x(t)$, and (*c*) test signal $y(t)$.

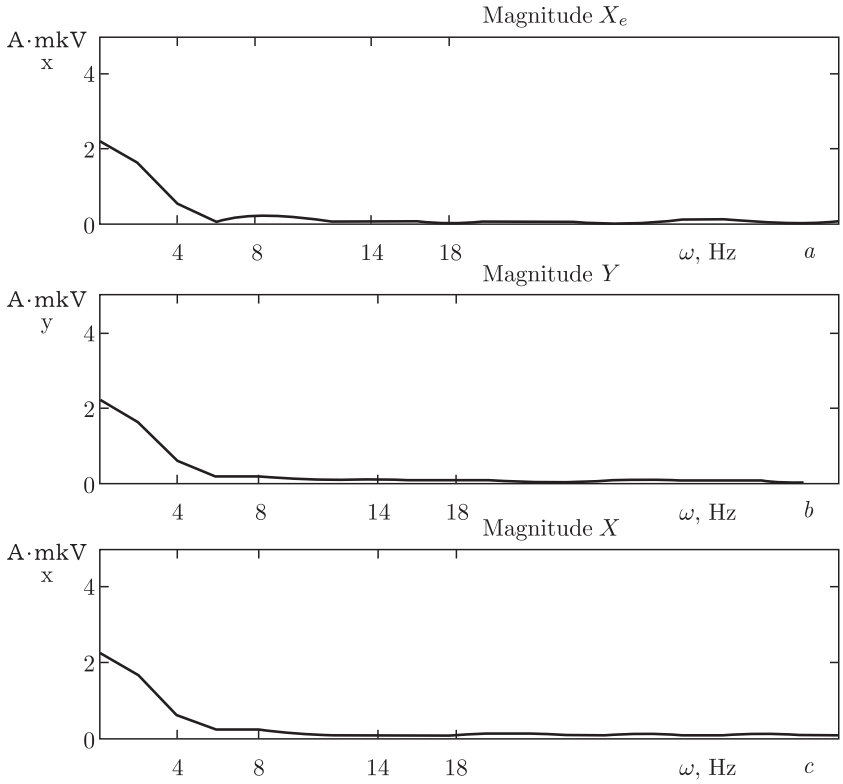


Fig. 3.7. Spectra of EEG signals processed by the Kravchenko window KKB_1: (*a*) standard signal with frequency 800 Hz, (*b*) test signal $x(t)$, and (*c*) test signal $y(t)$.

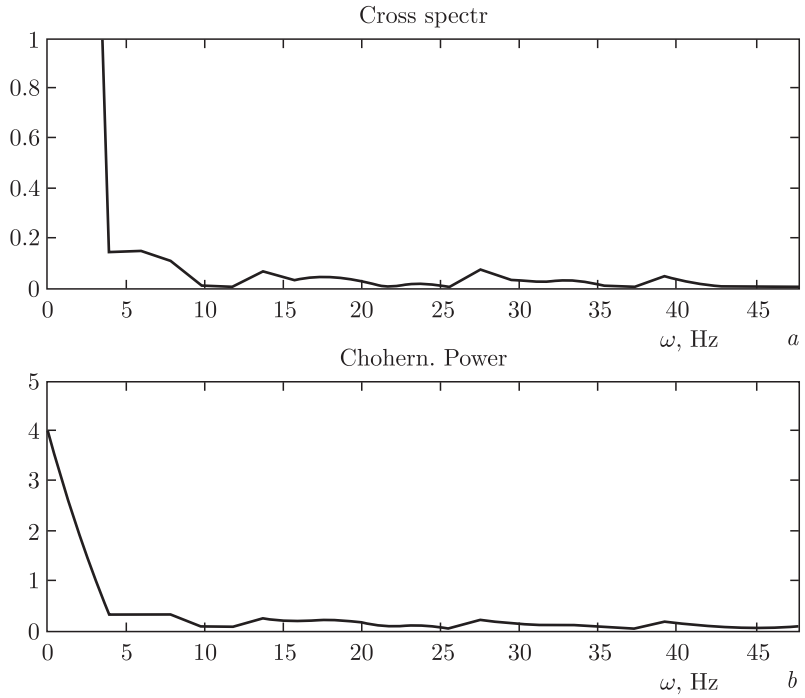


Fig. 3.8. Characteristics of investigated processes $x(t)$ and $y(t)$ using the Kravchenko window: (a) cross-spectrum; (b) coherent gain.

3.4. Approximation of a Given Function by Entire Functions of Exponential Type

One of the central problems in the theory of antenna synthesis is the possibility to approximate a given radiation pattern (RP) by entire functions of exponential type, i.e., functions of the class W_σ [11]. Indeed, if a given RP belongs to this class, then the synthesis problem has a unique solution. As is known [12], the required RPs are usually determined by practice and do not belong to the class of such functions. In this case, the synthesis problem does not have an exact solution and it is necessary to approximate a given RP with a prescribed accuracy by functions of the class W_σ.

Setting of the Problem. The following method of approximation is proposed in [12]. A given RP is approximated by a polynomial $P_k(z)$ of sufficiently high order. According to the Weierstrass theorem, such a procedure can be realized with any accuracy. Then, the obtained polynomial is multiplied by an auxiliary function $U_m(z)$ possessing the following properties:

1) $U_m(z)$ belongs to the class W_σ;

2) On the real axis, the function $U_m(z)$ is an infinitesimal value of order $o\,(1/z^m)$ as $z \to \infty$, where $m > k$;

3) On the interval of definition of the RP, $U_m(z)$ tends to unity as m tends to infinity, i.e., there exists such m that, for any ε_1, $|1 - U_m(z)| < \varepsilon_1$ in the domain $-L/\lambda \leqslant z \leqslant L/\lambda$, where L is the antenna length and λ is the wavelength. The product $P_k(z)U_m(z)$ belongs to the class W_σ, and therefore there exists an amplitude–phase current distribution in the antenna aperture which provides the given RP. Indeed,

since $|R(z) - P_k(z)| < \varepsilon_2$, we have

$$|R(z) - P_k(z)U_m(z)| \leqslant |R(z) - R(z)U_m(z)| + \\ + |R(z)U_m(z) - P_k(z)U_m(z)| < |R(z)|\,\varepsilon_1 + |U_m(z)|\,\varepsilon_2. \quad (3.9)$$

On the interval $-L/\lambda \leqslant z \leqslant L/\lambda$, conditions $|R(z)| \leqslant 1$ and $|U_m(z)| \leqslant 1$ hold for any m, therefore

$$|R(z) - P_k(z)U_m(z)| \leqslant \varepsilon_1 + \varepsilon_2, \quad m > k. \quad (3.10)$$

Similar reasoning can also be applied to the case of trigonometric polynomials $T_k(z)$ used as approximating functions. The RPs obtained belong to the class W_σ and are realizable and represented by the Kotelnikov series [12]:

$$R_s(z) = \sum_m R_s(z_m)\, S(z - z_m), \quad (3.11)$$

where

$$S(z) = \sin(\sigma z)/\sigma z,$$
$$z_n = \pi n/\sigma.$$

In [12], the Fourier transform of a cosine function to the power m was proposed to be used as the auxiliary function $U_m(z)$:

$$U_m(z) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{izy} \left[\frac{\Gamma^2(m/2+1)}{m!} \left(2\cos\frac{y}{2}\right)^m \right] dy, \quad (3.12)$$

where $\Gamma(\alpha)$ is the gamma-function. For odd m,

$$U_m(z) = \frac{\sin \pi z}{\pi z \prod_{p=1}^{m} \left(1 - \frac{z^2}{p^2}\right)}; \quad (3.13)$$

for even m,

$$U_m(z) = \frac{\cos \pi z}{\prod_{p=1}^{m} \left(1 - \left(\frac{2z}{2p+1}\right)^2\right)}. \quad (3.14)$$

Both the functions belong to the class W_π. At point $z = 0$ they have unity maxima. The first zero is located at the points $z = m/2 + 1$. At all points where $z = n$ and $n \leqslant m$, we have

$$U_m(n) = \frac{(m!)^2}{(m+n)!(m-n)!}. \quad (3.15)$$

At the points $z = n$ with $n > m$, we have $U_m(n) = 0$. The choice of the number m allows the approximation with any accuracy on a given interval, so it is possible to approximate the given RP with required accuracy (3.10). Note that, in practical calculations, the application of the function $U_m(z)$ in the capacity of an auxiliary function is not always effective. In the cases when the ratio L/λ is large, the order of a polynomial $P_k(z)$ must also be large but, since $m > k$, the function $U_m(z)$ beyond the interval $L/\lambda \leqslant |z| \leqslant m$ tends to zero slowly. This results in origination of sidelobes, which can be effectively suppressed by the methods of the theory of atomic functions (AF).

Atomic Functions. As is known [13], there exists a large family of AFs. The simplest of them is denoted by $\mathrm{up}(y)$.

The Fourier transform of $\mathrm{up}\,(y)$ is

$$Up(z) = \frac{1}{2\pi} \int\limits_{-\infty}^{\infty} e^{iyz}\,\mathrm{up}\,(y)dy = \prod_{p=1}^{\infty} \frac{\sin z2^{-p}}{z2^{-p}}. \tag{3.16}$$

It was used in the capacity of an auxiliary function for approximation of a given RP [13]. The RP obtained by the approximation is minimal beyond the interval $-L/\lambda \leqslant z \leqslant$ $\leqslant L/\lambda$. In [13], the AF $\Xi_n(y)$ was used for solving the synthesis problem. Its Fourier transform is

$$K_n(z) = \prod_{p=1}^{\infty} \frac{\sin z(n+1)^{-p}}{z(n+1)^{-p}}. \tag{3.17}$$

At $m = 1$, $\Xi_1(y) = \mathrm{up}\,(y)$. We have the following approximations of functions $R_s(z)$ and corresponding current distributions $f_s(y)$.

A. Approximation by trigonometric polynomials

$$R_s(z) = K_n\,(\alpha z) \sum_{m=-\beta(1-\alpha)/\alpha}^{\beta(1-\alpha)/\alpha} a_m e^{-i\beta z m}, \tag{3.18}$$

$$f_s(y) = \sum_{m=-\beta(1-\alpha)/\alpha}^{\beta(1-\alpha)/\alpha} a_m \Xi_n \left(\frac{y-\beta m}{\alpha}\right). \tag{3.19}$$

B. Approximation by translations of functions $K_n(z)$

$$R_s(z) = \sum_m a_m K_n\,(\alpha\,(z-\beta m)), \tag{3.20}$$

$$f_s(z) = \Xi_n \left(\frac{z}{\alpha}\right) \sum_m a_m e^{-i\beta y m}. \tag{3.21}$$

C. Polynomial approximation

$$R_s(z) = K_n\,(\alpha z) \sum_{m=0}^{k} a_m z^m, \tag{3.22}$$

$$f_s(y) = \alpha \sum_{m=0}^{k} (-i/\alpha)^m\, a_m \Xi_n^{(m)}\,(y/\alpha). \tag{3.23}$$

The choice of the parameters α, β, and a_n of these functions is determined by approximation methods (mean-square, uniform, or point-wise) [13].

3.5. The Use of Weighting Windows Based on the AF in SAR Digital Signal Processing

3.5.1. Introduction. The approaches to the synthesis of SAR digital signal processing systems are distinguished by their goals (object recognition, mapping, humidity investigation, etc.), physical interpretation of the processing procedure, mathematical tools, and methods used [19]. Therefore, different optimum criteria are applied for processing SAR signals, such as the Neumann–Pearson, maximum signal-to-noise ratio, minimum mean-square error criteria, etc. However, to within a constant factor, in any

of these approaches the optimal device must form a signal corresponding to a radar image by the SAR signal processing procedure as

$$J_i(\eta) = \left| \int_{-T/2}^{T/2} \dot{\xi}_i(t+\eta)\dot{h}(t)dt \right|, \tag{3.24}$$

where $J_i(\eta)$, $\dot{\xi}_i(t)$, and $\dot{h}(t)$ are the radar image signal, received signal, and support function, respectively, and T is the time of the antenna aperture synthesis. The support function, to within the initial phase, is a weighted function complex conjugate of the reflected signal:

$$\dot{h}(t) = H(t)\exp[j\Phi(t)]. \tag{3.25}$$

Here, $H(t)$ is a real-valued weighting function and $\Phi(t)$ is the support function phase variation law. If $H(t) = 1$, the SAR response to a single pointwise target has a high level of sidelobes (-13 dB). Therefore, other weighting functions are popular in practice (Gauss, Hamming, Kaiser, etc.) [20]. Here, application of the atomic functions (AF) [21] for constructing support functions used in SAR is discussed [30]. The main parameters of the new weighting windows are presented and compared with those of well-known windows.

3.5.2. Setting of the Problem of SAR Signal Processing. Consider the case when the antenna pattern axis is perpendicular to the line of the antenna phase center straightforward movement or the trajectory of flight. The Earth's surface is assumed to be flat. Axis OX is directed along the trajectory of flight, axis OZ is perpendicular to the Earth's surface, and axis OY is located in the plane of the Earth's surface, perpendicular to axes OX and OZ. We do not consider signal refraction and distortion in the troposphere, as well as other factors insignificant for studying main principles of the SAR. Let us describe a radiated signal by the harmonic function $u_0 = U_0\cos(\omega_0 t + \varphi_0)$, where U_0, ω_0, and φ_0 are constants. The simplest model of the reflected signal is based on the representation of the Earth's surface by a continuous or discrete set of elementary pointwise reflectors with different intensities. The reflected sensing signal corresponding to each pointwise target is received by a SAR antenna with a delay equal to the time required for the signal to propagate from the SAR antenna to the surface and back. The model is assumed to be linear, and the principle of superposition of signals is valid here. This allows us to simplify the finding of a complex object radar image.

Suppose that a pointwise target is situated at a point $A(0, y_i, 0)$ of the Earth's surface. The reflected signal has the form

$$u_i(t) = U_i G(t)\cos[\omega_0(t-\tau_i) + \varphi_0 + \varphi_i], \tag{3.26}$$

where U_i is the maximum amplitude of the signal, $G(t)$ is a normalized function characterizing the modulation of sensing and reflected signals during their transfer and reception by an antenna pattern, and φ_i is the change of the signal phase caused by reflection. In (3.26), τ_i is the time delay defined as $\tau_i = 2r_i(t)/c$, where c is the velocity of light, $r_i(t) = \sqrt{x^2(t) + y_i^2 + h_0^2}$ is the distance to the target, $x(t) = Vt$, V is the speed, and h_0 is the flight height. Hence, the expression for the reflected signal can be rewritten as

$$u_i(t) = U_i G(t)\cos[\omega_0 t - \psi_r(t) + \varphi_0 + \varphi_i], \tag{3.27}$$

where $\psi_r(t) = 2r_i(t)\omega_0/c = 4\pi r_i(t)/\lambda$.

Here, the SAR wavelength is $\lambda = 2\pi c/\omega_0$. If $L \ll r_0 = r_i(0) = \sqrt{y_i^2 + h_0^2}$, then, approximately, we can write

$$r_i(t) \approx r_0 + V^2 t^2/(2r_0), \tag{3.28}$$

$G(t) \approx 1$, and

$$u_i(t) \approx U_i \cos[\omega_0 t - 2\pi V^2 t^2/(\lambda r_0) + \psi_p], \tag{3.29}$$

where $\psi_p = \varphi_0 + \varphi_i - 4\pi r_0/\lambda$ is the unknown initial phase of the reflected signal.

To suppress the noise in signal processing, the complex-valued analytical signal $\dot{s}_i(t)$ is formed as $\dot{s}(t) = u_i(t) + i\,\mathrm{Hi}\,\{u_i(t)\}$, where $\mathrm{Hi}\,\{\cdot\}$ denotes the Hilbert transform. The equiphase and quadrature components of the reflected signal are determined as

$$\begin{gathered} u_{ic}(t) = U_i \cos[-2\pi V^2 t^2/(\lambda r_0) + \psi_i] \\ \text{and} \\ u_{is}(t) = U_i \sin[-2\pi V^2 t^2/(\lambda r_0) + \psi_i]. \end{gathered} \tag{3.30}$$

Here, $\psi_i = \varphi_i - 4\pi r_0/\lambda$ is the unknown random phase of the complex-valued signal, which is constant for the given target. So, the complex-valued envelope can be written as follows:

$$\dot{s}_i(t) = u_{ic}(t) + iu_{is}(t) = U_i \exp[-i(\psi_r(t) - \varphi_i)]. \tag{3.31}$$

The last expression determines the trajectory signal of a single pointwise target. As it follows from (3.31), the signal $\dot{s}_i(t)$ is modulated by the phase $\Phi_i(t) = 2\pi V^2 t^2/(\lambda r_0)$. The real signal to be processed usually is a sum of the reflected signal and noise, i.e.,

$$\dot{\xi}_i(t) = \dot{s}_i(t) + \dot{n}(t). \tag{3.32}$$

$\dot{n}(t)$ is the complex-valued Gaussian white noise whose real and imaginary parts have the normal distribution, a zero mean value, and a uniform spectral density on the whole frequency axis. Let us consider the synthesis of a processing system allowing one to extract useful information from reflected signals distorted by noise. As was mentioned earlier, the optimal device should form a signal corresponding to a radar image, to within a constant factor, by means of the SAR signal processing procedure (3.24):

$$J_i(\eta) = |\dot{J}(\eta)| = \left| \int_{-T/2}^{T/2} \dot{\xi}_i(t+\eta)\dot{h}(t)dt \right|,$$

where $J_i(\eta) = J_i(\chi/V)$ is the radar image signal; $\dot{J}(\eta)$ is the signal at the output of the processing system linear part; η and χ are the temporal and spatial shifts between the received signal $\dot{\xi}_i(t)$ and support function $\dot{h}(t)$, respectively. The support function is a weighted function that, to within the initial phase, is complex conjugate of reflected signal (3.25): $\dot{h}(t) = H(t)\exp[j\Phi_i(t)]$, where $H(t)$ is the weighting function.

3.5.3. Weighting windows based on the atomic functions in SAR signal processing. The use of weighting functions (windows) in the time domain influences essentially the energy leakage effect. The effect of this function on the analysed finite part of the signal is equivalent to the use of the rectangular window. This window is not optimal in analysis of signals due to discontinuity at the ends of the segment processed. A better weighting function must have zero values at both ends and vary monotonically inside the region of the processed part of the signal. The use of windows different from rectangular and smoothing discontinuities of the signal at the ends of the segment allows one to decrease the sidelobe level but, at the same time, the main lobe extends

and resolution is degraded. So, if signal spectral components close in the amplitudes are situated both in the vicinity and far from the weak component, then windows with equal sidelobe level should be chosen. If high resolution between close signal components is needed and distant components are absent, then windows with a very narrow main lobe and minimal amplitude of neighboring sidelobes are required. Signals with smooth spectra do not require windows at all.

Here, the atomic windows will be taken as $H(t)$. The following $(M+1)$-term windows can be used [23–25]:

$$\widetilde{w}(x) = w(x) + \sum_{k=1}^{M} c_k w^{(2k)}(x), \tag{3.33}$$

where $w(x)$ is a single-term atomic window normalized and centered as follows: $w(x)=0$ for $|x| > 1$, $w(0)=1$, and $w(-x) = w(x)$. The optimal choice of undetermined coefficients c_k in (3.33) allows one to reduce essentially the sidelobe level. Here, we will use only single- and two-term windows ($M=1$).

To compare different windows, the following system of parameters will be used: the maximum sidelobe level, dB; half-power beamwidth, deg; the angular distance to the first zero, deg; and the gain factor (aperture efficiency). Table 1 gives the important characteristics of several classical distributions in comparison with those based on the AFs. Here, l is the total length of the antenna aperture. As is seen from this table, the atomic window 6 is analogous to triangle window 2, although their forms are quite different. Windows 7, 8, and 12 are compatible with classical cosine windows 3 with $n=2$, 3 and 4, respectively. As for window 10, it should be noted that, with respect to the main characteristics, it is close to windows 4 and tends to window 1 as $a \to \infty$. The asymptotic decay of sidelobes for all atomic windows is equal to infinity. The family of atomic windows is very flexible and allows one to meet different requirements. Figure 1 shows the atomic windows and their power radiation patterns.

Recently a wide class of new windows based on combinations (products and convolutions) of the AFs with classical functions was proposed by V. F. Kravchenko [27, 28]. Some of them possess extraordinary properties making them useful in different problems of digital signal processing, including those connected with SAR.

As an example, Fig. 3.9 illustrates the real and imaginary parts of support functions (left) based on the atomic windows along with the corresponding synthesized radiation patterns (right). The time T of the synthesizing interval was equal to 3 sec. The other parameters were as follows: $h_0 = 400$ m, $y_i = 100$ m, $V = 220$ km/h, and $\omega_0 = 10\pi \cdot 10^7$ rad/sec.

So we may conclude that the properties of the atomic functions allow one to synthesize weighting functions with good parameters for the use in SAR digital signal processing. The results of numerical experiments prove the efficiency and flexibility of the novel approach. The application of two-dimensional and multi-dimensional weighting windows based on the AFs and R-functions [29] is also very promising for solving more complicated problems of SAR signal processing.

3.6. Synthesis of Two-Dimensional Window Functions on the Basis of Atomic Functions

Digital processing of multidimensional signals is one of the most promising directions in the combined use of the R-functions and AFs. These functions can be used in processing of two-dimensional signals in digital radar, synthesis of two-dimensional FIR and IIR filters, etc.

Table 3.3. Classical and Kravchenko–Rvachev weighting windows.

Type of distribution	Parameter	Half-power beamwidth, deg	Angular distance to first zero, deg	Maximum sidelobe level, dB	Gain factor
Classical windows					
rectangular	–	$50.8\,\lambda/l$	$57.3\,\lambda/l$	-13	1.000
$1-\lvert 2x/l\rvert$	–	$73.4\,\lambda/l$	$114.6\,\lambda/l$	-26	0.750
$\cos^n(\pi x/l)$	$n=1$	$68.8\,\lambda/l$	$85.9\,\lambda/l$	-23	0.810
	$n=2$	$83.2\,\lambda/l$	$114.6\,\lambda/l$	-32	0.667
	$n=3$	$95.1\,\lambda/l$	$143.2\,\lambda/l$	-40	0.575
	$n=4$	$110.6\,\lambda/l$	$171.9\,\lambda/l$	-48	0.515
$1-(1-\Delta)(2x/l)^2$	$\Delta=0.8$	$52.7\,\lambda/l$	$60.7\,\lambda/l$	-16	0.994
	$\Delta=0.5$	$55.6\,\lambda/l$	$65.3\,\lambda/l$	-17	0.970
	$\Delta=0$	$65.9\,\lambda/l$	$81.9\,\lambda/l$	-21	0.833
$\exp[-\alpha(2x/l)^2]$	$\alpha=2$	$66.5\,\lambda/l$	$71.0\,\lambda/l$	-31	0.811
	$\alpha=2.5$	$97.0\,\lambda/l$	$114.6\,\lambda/l$	-35	0.754
Kravchenko–Rvachev windows					
$\mathrm{up}\,(2x/l)$	–	$88.2\,\lambda/l$	$114.6\,\lambda/l$	-23	0.618
$\gamma[\mathrm{up}\,(2x/l)+$ $+\alpha\,\mathrm{up}\,''(2x/l)]$	$\alpha=0.005$	$87.0\,\lambda/l$	$113.9\,\lambda/l$	-26	0.643
	$\alpha=0.01$	$81.4\,\lambda/l$	$114.6\,\lambda/l$	-32	0.667
	$\alpha=0.02$	$77.3\,\lambda/l$	$113.9\,\lambda/l$	-28	0.714
$\gamma\cdot\mathrm{fup}_1(3x/l)$	–	$100.8\,\lambda/l$	$171.9\,\lambda/l$	-37	0.537
$\gamma[\mathrm{fup}_2(4x/l)+$ $+\alpha\,\mathrm{fup}_2''(4x/l)]$	$\alpha=0.05$	$99.1\,\lambda/l$	$162.9\,\lambda/l$	-41	0.559
	$\alpha=0.1$	$86.9\,\lambda/l$	$115.1\,\lambda/l$	-26	0.639
$\gamma\cdot h_a[2x/l(a-1)]$	$a=3$	$74.5\,\lambda/l$	$85.9\,\lambda/l$	-17	0.755
	$a=5$	$64.2\,\lambda/l$	$71.6\,\lambda/l$	-15	0.858
$\gamma[h_{3/2}(4x/l)+$ $+\alpha h''_{3/2}(4x/l)]$	$\alpha=1/10$	$87.0\,\lambda/l$	$115.1\,\lambda/l$	-28	0.633
	$\alpha=1/12$	$91.8\,\lambda/l$	$126.1\,\lambda/l$	-32	0.607
	$\alpha=1/16$	$94.2\,\lambda/l$	$145.7\,\lambda/l$	-36	0.574
$\gamma\cdot\Xi_2(2x/l)$	–	$103.1\,\lambda/l$	$171.9\,\lambda/l$	-34	0.528

As a rule, two-dimensional FIR-filters are synthesized on the basis of two-dimensional window functions defined on rectangular, circular, and hexagonal support domains (apertures). In the simplest case of a rectangular support area, the two-dimensional weighting window is formed by means of a tensor product of one-dimensional windows: $w[n_1, n_2] = w_1[n_1] \cdot w_2[n_2]$.

The most popular atomic weighting functions, which have been mentioned above, are as follows:

$$w_1(x) = \mathrm{up}\,(x),$$

$$w_2(x) = \text{up}(x) + 0.01\,\text{up}''(x),$$
$$w_3(x) = \text{fup}_1(3x/2)/\,\text{fup}_1(0),$$
$$w_4(x) = \frac{(\text{fup}_1(3x/2) + 0.0036\,\text{fup}_1''(3x/2))}{(\text{fup}_1(0) + 0.0036\,\text{fup}_1''(0))},$$
$$w_5(x) = h_{3/2}(x),$$
$$w_6(x) = 1.0696(h_{3/2}(x) + h''_{3/2}(x)/121),$$
$$w_7(x) = \Xi_2(x)/\Xi_2(0).$$

The weighting functions satisfy the following normalizing conditions: $w(x)$=0 for $|x| > 1$, $w(0)$=1, and $w(-x) = w(x)$. With the help of expression for the 2D filter $w[n_1, n_2] = w_1[n_1] \cdot w_2[n_2]$, one can synthesize windows with rectangular apertures.

We will use the following system of parameters to compare characteristics of two-dimensional windows in the plane ω_2=0:

equivalent noise bandwidth

$$b_1 = 4\frac{\int\limits_{-1}^{1}\int\limits_{-1}^{1} w^2(x,y)dxdy}{\left[\int\limits_{-1}^{1}\int\limits_{-1}^{1} w(x,y)dxdy\right]^2},$$

50% overlapping region correlation

$$b_2 = \frac{\int\limits_{-1}^{1}\int\limits_{0}^{1} w(x,y)w(x-1,y)dxdy}{\int\limits_{-1}^{1}\int\limits_{-1}^{1} w^2(x,y)dxdy} \cdot 100\%,$$

spurious amplitude modulation (in decibels) $b_3 = -10\log\left|\frac{W(\pi/2,0)}{W(0,0)}\right|^2$,

where $W(p,q)$ is the two-dimensional Fourier transform of the window function;

maximum conversion losses (in decibels) $b_4 = 10\log(b_1) + b_3$;

maximum sidelobe level (in decibels) $b_5 = 10\log\max_k\left|\frac{W(u_k,0)}{W(0,0)}\right|^2$,

where $\{u_k\}$ are the local maximum points (excluding u_0);

fsymptotic decay rate of side lobes (in decibels per octave) $b_6 = 10\log\lim_{u\to\infty}\left|\frac{W(2u,0)}{W(u,0)}\right|^2$;

window width at the six-decibel level b_7=2u,

where u is the highest frequency such that $10\log\left|\frac{W(0,0)}{W(u,0)}\right|^2 = 6$,

coherent gain $b_8 = \frac{1}{4}\int\limits_{-1}^{1}\int\limits_{-1}^{1} w(x,y)dxdy$.

The same parameters can also be given for the plane ω_1=0. Table 3.4 presents calculated physical parameters for some two-dimensional Kravchenko–Rvachev windows with a square support domain. As in the one-dimensional case, due to the infinite

smoothness of the AFs, the side lobes of these windows are characterized by infinite asymptotic decay rate.

The weighting window with circular a aperture is obtained by a rotation of a one-dimensional weighting window around the axis of symmetry:

$$w[n_1, n_2] = w[\sqrt{n_1^2 + n_2^2}], \tag{3.34}$$

where $w(\cdot)$ is a function of a one-dimensional continuous window. With the use of (3.34), one can obtain two-dimensional discrete, both classic (Blackman, Hamming, Kaiser, et.al.) and atomic (Kravchenko–Rvachev), windows possessing the circular symmetry. Such symmetry is desirable, for example, in image processing when all directions in the plane of the image are equivalent.

Table 3.4. Main physical parameters of two-dimensional Kravchenko–Rvachev windows with a square support domain.

Windows	Equivalent noise bandwidth, bin	Overlapping regions correlation (50% overlap),%	Spurious amplitude modulation, dB	Maximum conversion losses, dB	Maximum side-lobe level, dB	Window width at the six-decibel level, bin	Coherent gain
	b_1	b_2	b_3	b_4	b_5	b_7	b_8
$w_1(x)$	2.62	12	1.2	5.4	−23	2.1	0.25
$w_2(x)$	2.25	17	1.4	4.9	−32	1.9	0.25
$w_3(x)$	3.47	6	0.9	6.3	−37	2.4	0.15
$w_4(x)$	3.27	7	1.1	6.3	−51	2.3	0.16
$w_5(x)$	1.64	30	0.7	2.9	−36	2.9	0.27
$w_6(x)$	1.5	32	0.8	2.6	−51	2.5	0.31
$w_7(x)$	3.59	5	0.9	6.5	−34	2.4	0.14

Finally, the window with a hexagonal aperture is constructed on the basis of a one-dimensional window $w[n]$ by means of the tensor product

$$w[n_1, n_2] = w[n_1] \cdot w\left[\frac{n_1 + n_2\sqrt{3}}{2}\right] \cdot w\left[\frac{n_1 - n_2\sqrt{3}}{2}\right]. \tag{3.35}$$

Figures 3.9–3.10 present two-dimensional windows with circular and hexagonal support areas, based on the AF $\mathrm{up}\,(x)$ as well as their logarithmic FRs, in dB. Figure 3.9 c is an arbitrary normal section of the FR passing through the origin, and Figure. 3.10 c is a section of the FR by the plane $\omega_2 = 0$.

Figures 3.11–3.12 demonstrate the FR of low-pass FIR-filters on rectangular, circular, and hexagonal support domains constructed on the basis of the one-dimensional window $\mathrm{up}\,(x)$. The cutoff frequency $\omega_{\mathrm{c}} = 0.5\pi$; the boundary pass frequency $\omega_p = \omega_{\mathrm{c}} - 0.1\pi$; and the boundary delay frequency $\omega_d = \omega_{\mathrm{c}} + 0.1\pi$ are presented there.

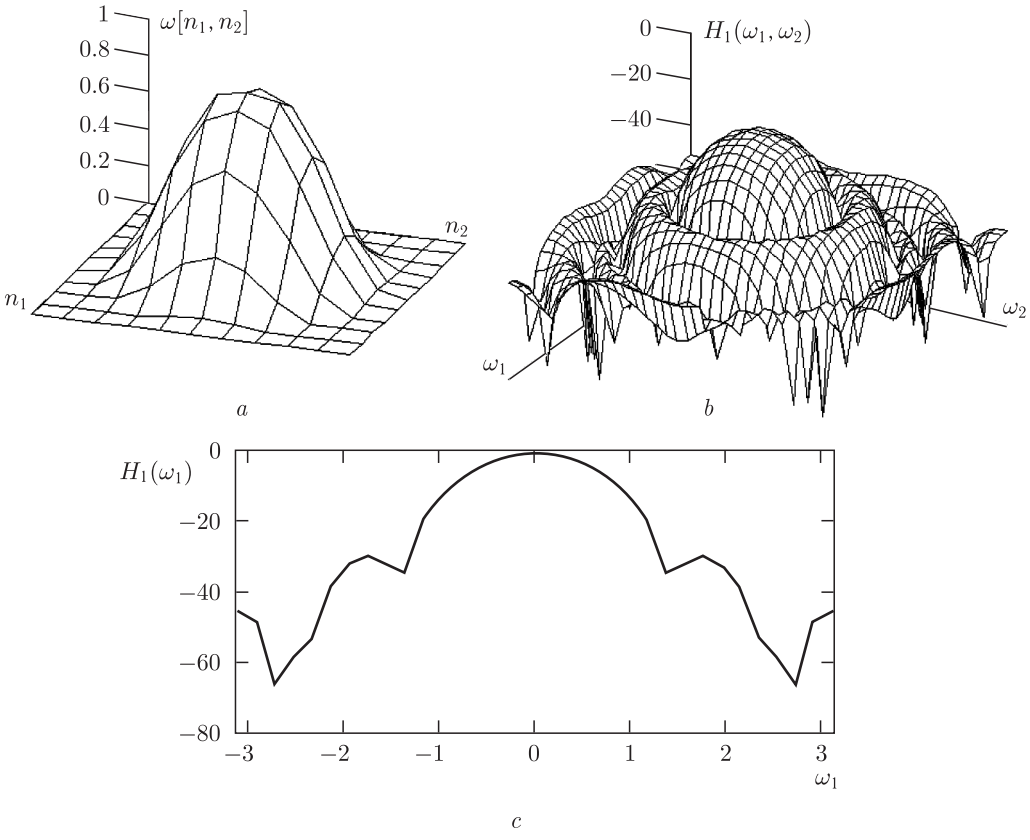


Fig. 3.9. (a) Two-dimensional window $w[n_1, n_2] = \text{up}\left[\sqrt{n_1^2 + n_2^2}\right]$ with circular aperture, (b) its logarithmic FR, and (c) a section of the FR by the plane $\omega_2 = 0$.

We assume that the ideal FR for a rectangular domain has the following form:

$$H_0(\omega_1, \omega_2) = \begin{cases} 1, & |\omega_1| \leqslant \omega_c,\ |\omega_2| \leqslant \omega_c, \\ 0, & otherwise, \end{cases}$$

and those for circular and hexagonal domains are

$$H_0(\omega_1, \omega_2) = \begin{cases} 1, & \sqrt{\omega_1^2 + \omega_2^2} \leqslant \omega_c, \\ 0, & otherwise. \end{cases}$$

Unfortunately, the aforementioned approaches (tensor product and rotation) do not allow us to synthesize two-dimensional filters with arbitrary support domains. This especially concerns concave domains, for example, cross- or star-shaped. For such areas only filters based on the simplest Dirichlet window can be constructed. Perhaps, due to this difficulty, only three aforementioned types of support domains are used in practice. Below we propose a technique based on the RFM for a synthesis of two-dimensional windows on arbitrarily shaped domains.

In problems of digital filtration of two-dimensional signals, filters with a finite impulse response (FIR-filters) are widely used. One of their advantages in comparison

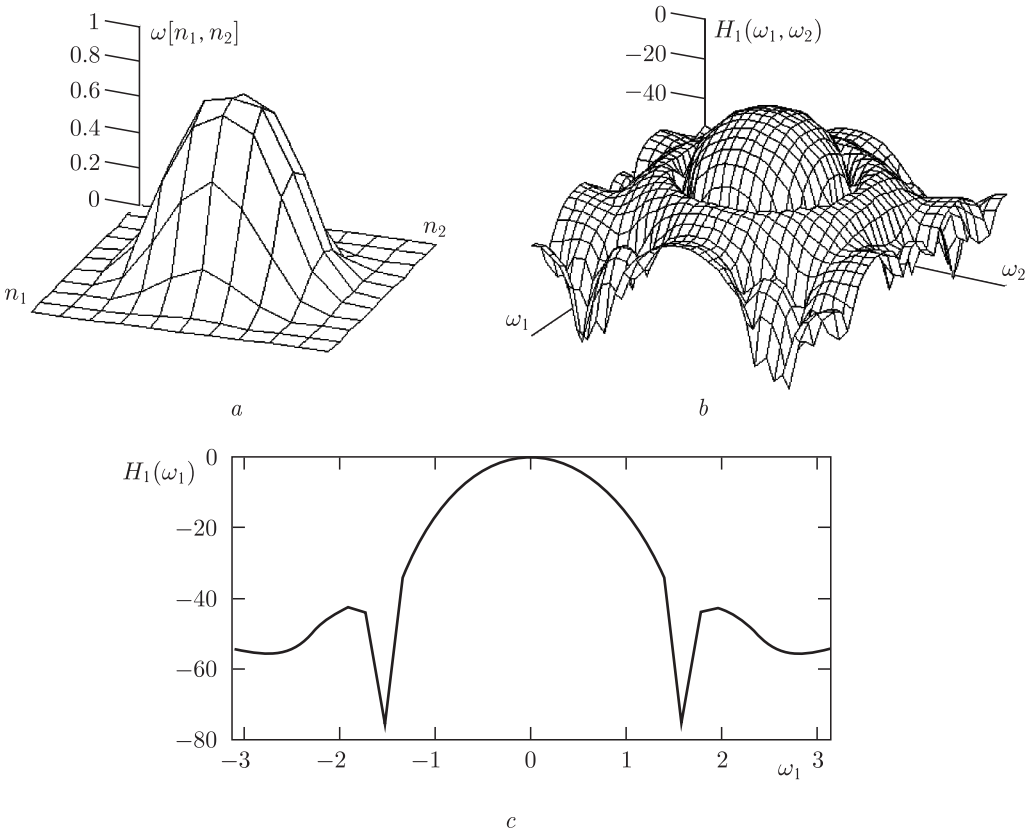


Fig. 3.10. (*a*) Two-dimensional window $w[n_1, n_2]$ with hexagonal aperture, (*b*) its logarithmic FR, and (*c*) a section of the FR by the plane $\omega_2 = 0$.

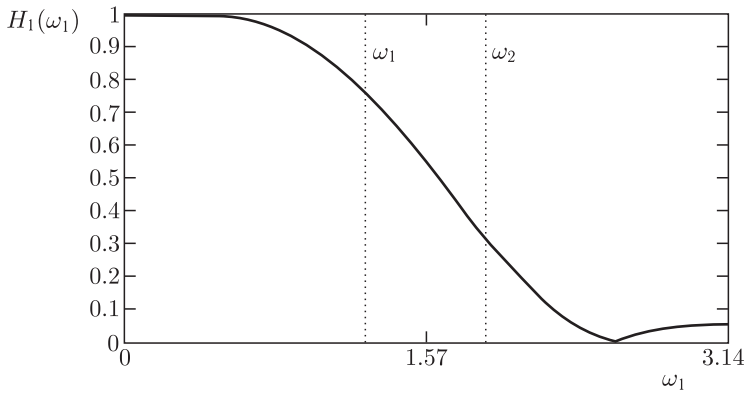


Fig. 3.11. (*a*)The FR of the low-pass FIR-filter constructed by the window $w[n_1, n_2] = \text{up}\,[n_1] \times$ $\times\, \text{up}\,[n_2]$ and (*b*) the section of the FR by plane $\omega_2 = 0$.

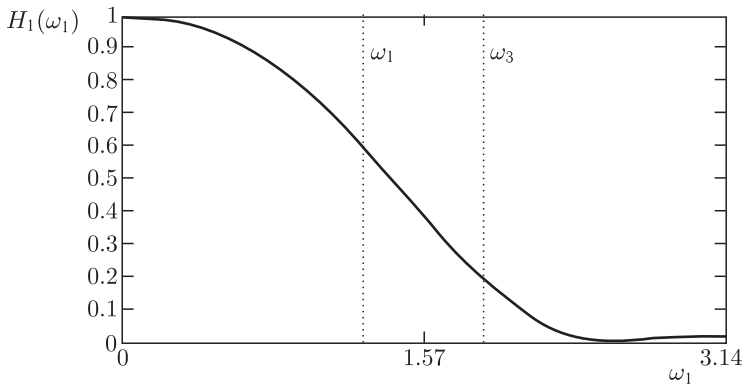


Fig. 3.12. The FR of the low-pass FIR-filter constructed by the window $w[n_1, n_2] = \text{up}\left[\sqrt{n_1^2 + n_2^2}\right]$ and (*b*) the section of the FR by plane $\omega_2 = 0$.

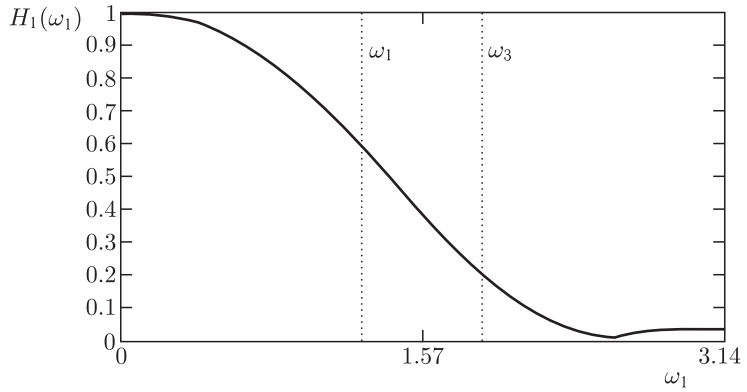


Fig. 3.13. The FR of the low-pass FIR-filter constructed by the window $w[n_1, n_2] = \text{up}\,[n_1] \cdot \text{up}\left[\frac{n_1 + n_2\sqrt{3}}{2}\right] \cdot \text{up}\left[\frac{n_1 - n_2\sqrt{3}}{2}\right]$ and (*b*) the section of the FR by plane $\omega_2 = 0$.

with filters with infinite impulse response (IIR-filters) is the possibility of synthesizing filters with a zero phase lag. This property is very important for various applications of two-dimensional signal processing. In particular, filters with a nonzero phase lag can cause damages of lines and borders on images being processed. Besides, the techniques of two-dimensional IIR-filter synthesis are more complicated because of difficulties of providing their stability.

3.7. Synthesis of Two-Dimensional FIR-Filters

A two-dimensional FIR-filter provides a zero phase lag if its frequency response is real-valued, i.e., $H(\omega_1, \omega_2) = H^*(\omega_1, \omega_2)$ or, if its impulse response is symmetric with respect to the origin,

$$h[n_1, n_2] = h^*[-n_1, -n_2]. \tag{3.36}$$

There are several methods for two-dimensional FIR-filter synthesis. Some of them are the methods of weighting windows, frequency sampling, and frequency transforms. Here, we will consider the method of weighting windows as most widely used in practice.

According to it, the required two-dimensional filter frequency response $H_0(\omega_1, \omega_2)$ is presented in the form of the Fourier series

$$H_0(\omega_1, \omega_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} h_0[n_1, n_2] e^{-j(\omega_1 n_1 + \omega_2 n_2)} \tag{3.37}$$

with the coefficients $h_0[n_1, n_2]$ determined by the expression

$$h_0[n_1, n_2] = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} H_0(\omega_1, \omega_2) e^{j(\omega_1 n_1 + \omega_2 n_2)} d\omega_1 d\omega_2. \tag{3.38}$$

Here, $h_0[n_1, n_2]$ is an infinite impulse response for a two-dimensional filter corresponding to the required frequency response $H_0(\omega_1, \omega_2)$. For a realizable two-dimensional FIR-filter, the bounds of summation in (3.37) must be limited. This deteriorates the convergence of the truncated series (3.37) to the required frequency response at its points of discontinuity (the Gibbs effect). To improve the convergence, we should multiply the coefficients $h_0[n_1, n_2]$ by a proper two-dimensional weighting window function $w[n_1, n_2]$, i.e., we get the following coefficients:

$$h[n_1, n_2] = w[n_1, n_2] \cdot h_0[n_1, n_2]. \tag{3.39}$$

For filters with a zero phase lag, the window function must satisfy the condition

$$w[n_1, n_2] = w^*[-n_1, -n_2]. \tag{3.40}$$

The new two-dimensional windows proposed and justified in this work can be used in problems of multidimensional signal processing in Doppler's and synthetic-aperture radar, for resolution and compression of signals, in telemedicine, mathematical modeling of the heart generator, computer thermography and tomography, etc.

References

1. *Dudgeon, D.E. and Mersereau, R.M.* Multidimensional Digital Signal Processing. N.J., Prentice-Hall, Inc., Englewood Cliffs, 1984.
2. *Rabiner, L.R. and Gold, B.* Theory and Applications of Digital Signal Processing. N.J., Prentice-Hall, Inc., Englewood Cliffs, 1975.
3. *Kuzmin, S.Z.* Tsifrovaya Radiolokatsiya (Digital Radar). Kiev, Kvits, 2000.
4. *Gulyaev, Yu.V., Kravchenko, V.F., and Rvachev, V.A.* Synthesis of Weighting Windows Based on Atomic Functions. DAN RAN, 1995, vol. 342, no. 1, pp. 29–31.
5. *Kravchenko, V.F., Basarab, M.A., and Perez-Meana, H.* Atomic Functions and the Construction of New Windows in Problems of Digital Speech Analysis and Modeling. Telecommunications and Radio Engineering, 2001, vol. 56, no. 1, pp. 3–17.
6. *Kravchenko, V.F., Basarab, M.A., and Perez-Meana, H.* Spectral Properties of Atomic Functions in Digital Signal Processing. Journal of Communications Technology and Electronics, 2001, vol. 46, no. 5, pp. 494–511.
7. *Kravchenko, V.F., Basarab, M.A., and Perez-Meana, H.* The New Class of Weighting Windows Based on Atomic Functions in Problems of Digital Signal Processing, Proc. of the 1^{st} International Workshop on Mathematical Modeling of Physical Processes in Inhomogeneous Media, March 20–22, 2001, Guanajuato, Gto., Mexico, pp. 5–7.
8. *Kravchenko, V.F., Rvachev, V.A., and Rvachev, V.L.* Construction of New Windows Based on Atomic Functions for Signal Processing. DAN AN of USSR, 1989, vol. 306, no. 1, pp. 78–81.

9. Telemeditsina. Novye informatsionnye tekhnologii na poroge XXI veka (Telemedicine. New Information Technologies on the Eve of the 21th Century). Eds. R. M. Yusupov and R. I. Polonnikov, St.Petersburg: Inst. Inform. i Avtomatiz. of the Russian Acad. Sc., 1998.
10. *Kulaichev*, *A. P.* Komp'yuternaya elektrofiziologiya v klinicheskoi i issledovatel'skoi pratike (Computer Electrophysiology in Clinical and Research Practice). Moscow: NPO Informatika i komp'yutery, 1999.
11. *Akhiezer*, *N. I.* Lectures on the Approximation Theory (in Russian), Moscow, Nauka, 1965.
12. *Zelkin*, *E. G. and Sokolov*, *V. G.* Methods of Antenna Synthesis (in Russian), Moscow, Sov.Radio, 1980.
13. Kravchenko, V. F., Pustovoit, V. I., and Timoshenko, V. V. Doklady RAN, 1999, vol. 368, no. 2.
14. *Kravchenko*, *V. F.* Electromagnetic Waves and Electronic Systems, 1998, vol. 3, no. 1, pp. 40–47.
15. *Zelkin*, *E. G.* Postroenie izluchayushchei sistemy po zadannoi diagramme napravlenosti (Constructing a Radiator System by Its Radiation Pattern), Moscow–Leningrad, Gosenergoizdat, 1963.
16. Handbook on Antenna Engineering. vol. 1. Eds. Ya.N.Feld and E.G. Zelkin. Moscow, IPRZhR, 1997.
17. *Varga*, *R. S.* Functional Analysis and Approximation Theory in Numerical Analysis. Society for Industrial and Applied Mathematics, Philadelphia, 1974.
18. *Basarab*, *M. A.* Synthesis of Antennas with Single-Lobed Radiation Patterns on the Base of Atomic Functions. Antenny, 2002, issue 5(60), pp. 4–8.
19. *Soumekh*, *M.* Synthetic Aperture Radar Signal Processing, NY: John Wiley & Sons, Inc., 1999.
20. *Harris*, *F.* On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. Proc. of the IEEE, 1978, vol. 66, no. 1, pp. 51–83.
21. *Rvachev*, *V. L.*, *and Rvachev*, *V. A.* Non-Classical Methods of Approximation Theory in Boundary Value Problems. Kiev, Naukova Dumka, 1979. (in Russian)
22. *Goncharenko*, *A. A.*, *Kravchenko*, *V. F.*, *and Ponomarev*, *V. I.* Remote Sensing of Inhomogeneous Media. Moscow: Mashinostroenie, 1991. (in Russian)
23. *Kravchenko*, *V. F.*, *Rvachev*, *V. A.*, *and Rvachev*, *V. L.* Construction of New Windows for Signal Processing on the Basis of Atomic Functions. DAN of the SSSR, 1989, vol. 306, no. 1, pp. 78–81.
24. *Gulyaev*, *Yu. V.*, *Kravchenko*, *V. F.*, *and Rvachev*, *V. A.* Synthesis of Weight Windows on the Basis of Atomic Functions. Doklady Physics, 1995, vol. 51, no. 3, pp. 456–458.
25. *Kravchenko*, *V. F.*, *Rvachev*, *V. A.*, *and Rvachev*, *V. L.* Mathematical Methods for Signal Processing Based on Atomic Functions. Journal of Communications Technology and Electronics, 1995, vol.40 (12), pp. 118–137.
26. *Kravchenko*, *V. F.*, *Basarab*, *M. A.*, *and Perez-Meana*, *H.* Spectral Properties of Atomic Functions in Digital Signal Processing. Journal of Communications Technology and Electronics, 2001, vol. 46, no. 5, pp. 494–511.
27. *Zelkin*, *Ye. G. and Kravchenko*, *V. F.* Atomic Functions in Antenna Synthesis Problems and New Synthesized Windows. Journal of Communications Technology and Electronics, 2001, vol. 46, no. 8, pp. 829–857.
28. *Kravchenko*, *V. F.* New Synthesized Windows. DAN RAN, 2002, vol. 382, no. 2, pp. 190–198.
29. *Basarab*, *M. A. and Kravchenko*, *V. F.* R-Functions and Atomic Functions in Problems of Digital Processing of Multivariate Signals. Telecommunications and Radio Engineering, 2001, vol. 56, no. 4&5, pp. 4–44.
30. *Kravchenko*, *V. F.*, *and Basarab*, *M. A.* The Use of Weighting Windows Based on Atomic Functions in SAR Digital Signal Processing. Electromagnetic Waves and Electronic Systems, 2002, vol. 7, no. 4–5, pp. 134–140.

31. *Kravchenko*, *V.F. and Basarab*, *M.A.* A New Method of Multidimensional-Signal Processing with the Use of R functions and Atomic Functions. Doklady Physics, 2002, vol. 47, no. 3, pp. 195–200.
32. *Basarab*, *M.A.*, *Kravchenko*, *V.F.*, *and Masyuk*, *V.M.* R-Functions, Atomic Functions and Their Applications. Zarubezhnaya Radioelektronika. Uspekhi Sovremennoi Radioelektroniki, 2001, no. 8, pp. 5–40.
33. *Gulyaev*, *Yu.V.*, *Kravchenko*, *V.F.*, *and Rvachev*, *V.A.* Synthesis of Weighting Windows Based on Atomic Functions. Dokl. Akad. Nauk, 1995, vol. 342, no. 1, pp. 29–31.
34. *Kravchenko*, *V.F.*, *Basarab*, *M.A.*, *Pustovoit*, *V.I.*, *and Perez-Meana*, *H.* New Constructions of Weighting Windows on the Base of Atomic Functions in Problems of Speech Synthesis. DAN RAN, 2001, vol. 377, no. 2, pp. 183–189.
35. *Dudgeon*, *D. and Mersereau*, *R.* Multidimensional Digital Signal Processing. N.J., Prentice-Hall, Inc., Englewood Cliffs, 1984.
36. *Goncharenko*, *A.A.*, *Kravchenko*, *V.F.*, *and Ponomarev*, *V.I.* Distantsionnoe zondirovanie neodnorodnykh sred (Remote Probing of Ihomogeneous Media), Moscow, Mashinostroenie, 1991.
37. *Kuz'min*, *S.Z.* Tsifrovaya radiolokatsiya (Digital Radar). Kyiv, Kvits, 2000.

Chapter 4

WAVELET SYSTEMS AND ATOMIC FUNCTIONS

4.1. Introduction

Wavelet systems (hereafter called W-systems) have become very popular now. They are discussed in scientific journals and textbooks [1–4], in monographs [4–6], and in hundreds of articles. Many reports at special conferences on the W-systems have been made. These systems have found wide use in areas where solution by other methods is especially difficult, such as mathematical analysis, theory of partial differential equations, seismology, radio physics, music and speech analysis, and image processing. Suggestions are being made that expansions in the W-systems will compete successfully with the Fourier expansions and Fourier transforms and that the work of human auditory and visual organs is based on the «wavelet» principle. The W-systems are as popular now as splines in their time. It is sufficient to note that the report on the W-systems made by I. Daubechies, the inventor of compactly supported W-systems, was among the several plenary reports devoted to the deepest questions of pure mathematics at the International Congress of Mathematicians (Zurich, 1994). This fact was unprecedented in forums of such a kind (splines were not honored with such attention).

The way to wide and justified application of the W-systems has some obstacles. The main one is a relatively awkward mathematical apparatus. Here, we present the description of constructive methods for the simplest and, in our opinion, most useful W-systems, analysis of their benefits and shortcomings, methods of improving them, and also their comparison with other orthogonal systems. The main focus is given to W-systems, their modifications, and methods of constructing them that are investigated by the authors. Here, the commonly used ideology of wavelets as orthogonal systems consisting of translations and dilations of one or more functions possessing both spatial and frequency localization was used. To simplify the presentation, we will restrict ourselves to one-dimensional W-systems. Multidimensional ones can be obtained as products of one-dimensional W functions, although other methods of constructing multidimensional W-system are also of great interest.

4.2. Basic Principles of Wavelet Analysis

The idea of the W-systems consists in the consideration of functional systems generated by translations and dilations of some generative functions such that the functions of the W-system along with their Fourier transforms are localized, to some degree, in a finite domain. The coefficients of expansion of an arbitrary function into these systems contain, in a sufficiently explicit form, information on the behavior of the function and its instantaneous spectrum, i.e., the W-systems must realize a local Fourier analysis (LFA). The most frequently considered orthogonal W-systems are constructed by means of a proper choice of generative functions. The first variants of the W-systems had one or two generative functions. Now it is common to use two generative functions

(in the one-dimensional case), although a greater number of them should be taken in order to provide a good expansion in the frequency domain when LFA is realized.

The main requirements to the W-systems are as follows: localization in the time domain; localization in the frequency domain; good approximation properties sufficient for the application using the specific W-system; smoothness, i.e., a sufficient continuous differentiability; and the ease of calculation.

In its turn, each of these properties of the W-system has several gradations.

As for the localization, the W-system functions can be

1. decreasing as $|x|^{-m}$ if $|x| \to \infty$ for some fixed m, where x is a spatial (temporal) variable;
2. decreasing faster than $|x|^{-m}$ for all m as $|x| \to \infty$, i.e., rapidly decreasing;
3. decreasing as $e^{-\alpha|x|}$, $\alpha > 0$ if $|x| \to \infty$, i.e., exponentially decreasing (exponential localization);
4. equal to zero outside a finite interval, i.e., compactly supported.

In addition to the aforementioned classification, a more detailed quantitative classification determining the localization is of practical interest. For example, we may say that the function $f(x)$ is δ-localized at the point x_0 with degree ε if

$$\frac{\int\limits_{x_0-\delta}^{x_0+\delta} |f(x)|^2\,dx}{\int\limits_{-\infty}^{\infty} |f(x)|^2\,dx} \geqslant 1-\varepsilon. \tag{4.1}$$

In the qualitative sense, a compactly supported localization is better than an exponential one. However, if we define the ε-support of a function $f(x)$ as the set $\varepsilon - \operatorname{supp} f(x) = \{x, |f(x) > \varepsilon|\}$, then the ε-support of an exponentially localized function can be shorter than that of a compactly supported function.

As is known, the time-domain localization of the function and the frequency-domain localization of its Fourier transform are opposite in some sense, especially when this localization is evaluated qualitatively. Certainly, there exist rapidly decreasing functions whose Fourier transforms are also rapidly decreasing. The necessary and sufficient condition for this is that the function itself is of the class C^∞, i.e., infinitely differentiable. However, the Fourier transform of a rapidly decreasing C^∞ function has a small ε-support (when the ratio $\varepsilon/\|f\|_{L_2}$ is small). The Fourier transform of a compactly supported function also cannot be compactly supported being an entire function of exponential type.

The smoothness of the W-system is due to its the frequency-domain localization. So, Y. Meyer's W-systems have compactly supported Fourier transforms; therefore, they are entire functions of exponential type. Functions of the Daubechies and the Stromberg–Lemarie–Battle W-systems have only a finite number of continuous derivatives; hence, their Fourier transforms decrease as $|t|^{-m}$ when $t \to \infty$ (t is an independent variable in the frequency domain).

From the viewpoint of high resolution with respect to the frequency, a good frequency-domain localization is requiered from functions of a W-system in order to provide an effective estimation of instantaneous spectrum of the signal when the LFA is realized.

A natural approach to this local Fourier analysis could be as follows. Let the function $\varphi(x)$ be nonnegative, equal to zero outside the interval $[-1,1]$, close to unity when

$x \in [-1+\varepsilon, 1-\varepsilon]$, and

$$\sum_{k=-\infty}^{\infty} \varphi(x-k) \equiv 1, \tag{4.2}$$

i.e., its translations form the partition of unity.

The following functions are taken as the LFA basic functions:

$$\varphi_{n,k,r}(x) = \varphi\left(2^r x - k\right) \exp\left(2\pi i n x 2^r x\right). \tag{4.3}$$

In this connection, we should note D. Gabor's proposal to use the system

$$g_k(x) = \exp\left(-(x-k)^2/2 + ix\right). \tag{4.4}$$

The function $\exp\left(-x^2\right)$ is not compactly supported but very rapidly decreasing. On the other hand, the system $g_k(x)$ is invariant with respect to the Fourier transform.

The atomic function $\mathrm{up}(x)$ also can be taken as $\varphi(x)$.

The nonorthogonality of such systems is a shortcoming of the approach described. The Gabor system is not even minimal: the omission of one of the functions in the Gabor system does not affect the completeness of the remaining system. In essence, this approach is a Fourier analysis with the use of time-domain windows.

4.3. W-systems

An alternative approach is to use the W-systems. Usually, the W-systems are introduced by means of multiresolution analysis. Let L_n, $n = 0, 1, 2, \ldots$, be a sequence of subspaces of $L_2(R)$-space of functions square-integrable on the real axis R (or on any translation-invariant space on R), so that the space L_n contains in L_{n+1} ($L_n \subset L_{n+1}$). Then, the term "nested subspaces" may be used. The space L_n is generated by translations of functions $\varphi_{n,s}(x)$, $s = 1, 2, \ldots, m_n$, by the divisible steps $h_n^{(s)}$; i.e., $L_n = C1\,\mathrm{Span}\,\varphi_{n,s}\left(x - k\,h_n^{(s)}\right)$, where $C1$ is the closure and Span denotes the linear span (here usually $m = 1$).

Functions $\varphi_{n,s}(x)$ must be localized to some extent with respect to temporal and frequency variables; i.e., both $\varphi_{n,s}(x)$ and their Fourier transforms must be concentrated close to some points $x_{n,s}$ and $t_{n,s}$, respectively. In this case, the localization with respect to the spatial variable must increase as $h_n \to 0$ and n increases. A variant of multiresolution analysis when n changes from $-\infty$ to $+\infty$ also exists. In that case, the localization with respect to the frequency variable increases as $h_n \to \infty$ and $n \to -\infty$.

Thus, functions from the space L_n have the form

$$f(x) = \sum_{k=-\infty}^{\infty} \sum_{s=1}^{\infty} c_{k,s} \varphi_{n,s}\left(x - k h_n^{(s)}\right), \tag{4.5}$$

where $\int\limits_{-\infty}^{\infty} |f(x)|^2 < +\infty$, i.e.,

$$\sum_{k=-\infty}^{\infty} \sum_{s=1}^{\infty} |c_{k,s}|^2 < +\infty \quad \text{if} \quad 0 < c < \int\limits_{-\infty}^{\infty} \left|\varphi_{n,s}(x)^2\right| dx < C.$$

If $\varphi_{n,s}(x)$ are orthonormal functions, then $\int\limits_{-\infty}^{\infty} |\varphi_{n,s}(x)|^2\, dx = 1$.

It is also required that the closure of the union of L_n coincides with the whole space $L_2(R)$:

$$C1 \bigcup_{n=0}^{\infty} L_n = L_2(R). \tag{4.6}$$

Then, in each space L_n, one can find an orthogonal complement $W_n = L_{n-1}^{\perp}$ for the space L_{n-1}, i.e., $\int\limits_{-\infty}^{\infty} \psi(x)\overline{\varphi}(x)dx = 0$ (the bar corresponds to complex conjugation).

Thus, $L_n = L_{n-1} \oplus W_n$; i.e., any element $f(x) \in L_n$ is uniquely representable by the sum of functions $f_1(x) \in L_{n-1}$ and $f_2(x) \in W_n$, where $f_1(x)$ and $f_2(x)$ are orthogonal. Suppose also that $W_0 = L_0$.

For any function $f_1(x) \in L_2(R)$, its orthogonal projection $pr_n f(x)$ on the space L_n is determined; i.e., the function $f(x) - pr_n f(x)$ is orthogonal to $pr_n f(x)$.It is clear that $pr_n f(x) = pr_{n-1} f(x) + pr_{w_n} f(x)$. According to condition (4.6), $\lim\limits_{n\to\infty} \|f(x) - pr_n f(x)\|_{L_2(R)} = 0$.

Hence, $f(x)$ is uniquely representable in the form $f(x) = \sum\limits_{n=0}^{\infty} w_n(x)$, where functions $w_n(x) \in W_n$ are orthogonal.

Now, one should choose an orthogonal basis in each space W_n. This basis must consist of translations of $h_n^{*(s)}$, the specific functions providing the orthogonality of $\mu_{n,s}(x) \in W_n,\ s = 1, \ldots, m_n^*$, where $h_n^{*(s)}$ and m_n^* can be different from $h_n^{(s)}$ and m_n. Then, we obtain the union of all functions $\mu_{n,s}\left(x - k\, h_n^{*(s)}\right),\ n = 0, 1, \ldots, m_n^*$. The complete orthogonal system is the required W-system. Any function $f(x)$ from $L_2(R)$ is uniquely representable as the sum of the series

$$f(x) = \sum_{n=0}^{\infty} \sum_{s=1}^{m_n^*} \sum_{k=-\infty}^{\infty} c_{n,s,k} \mu_{n,s}\left(x - k\, h_n^{*(s)}\right),$$

where

$$c_{n,s,k} = \frac{\int\limits_{-\infty}^{\infty} f(x)\overline{\mu}_{n,s}\left(x - k\, h_n^{*(s)}\right) dx}{\int\limits_{-\infty}^{\infty} |\mu_{n,s}(x)|^2\, dx}.$$

The partial sum $S_n(f)$ of this series belongs to the space L_n. It gives us the smoothed picture of the signal $f(x)$. The temporal resolution is enhanced and low-frequency components increase as n increases. This expansion allows one to implement the signal analysis including the LFA, its synthesis, filtering of some frequency components, and efficient coding.

Functions $\mu_{n,s}(x)$ must be chosen maximally localized with respect to the temporal and frequency variables, whereas they often cannot be localized in space, in contrast to the initial nonorthogonal functions $\varphi_{n,s}(x)$. However, functions $\mu_{n,s}(x)$ are localized in the frequency domain and provide better frequency resolution as compared to the initial functions $\varphi_{n,s}(x)$.

The aforementioned system of multiresolution analysis is satisfied by Mayer, Stromberg, Lemarie–Battle, and Daubechies wavelets, i.e., the W-systems obtained by means of orthogonalization in the set of spaces UP_n generated by translations

up $(x - k\,2^{-n})$. As described above, the construction of a W-system with the help of multiresolution analysis consists of the following steps:

1. Choose a nested sequence of spaces L_n generated by translations of functions localized both in time and in frequency domains.
2. Pass from spaces L_n to orthogonal spaces W_n, which means orthogonalization between different levels or scales in the frequency domain; this corresponds to the orthogonalization of different ranges (when $h_n = 2^{-n}$) from different octaves.
3. Orthogonalize functions from the same W_n (i.e., orthogonalize the functions whose Fourier transforms are located in the same range and in the same octave).

Let us consider the first stage. The nested spaces L_n $(L_n \subset L_{n+1})$ generated by translations of the same function $\varphi_n(x)$ are not quite arbitrary. More exactly, the generative functions $\varphi_n(x)$ must be infinite convolutions of atomic lattice measures. This means that the Fourier transform of the function $\widetilde{\varphi}_m(x)$ is an infinite product of periodic functions.

Thus, the Fourier transform of a B spline of order r generating Stromberg and Lemarie–Battle W-systems is presented in the form

$$\widetilde{B}_r(t) = \left(\frac{\sin t/2}{t/2}\right)^{r+1} = \prod_{k=1}^{\infty}\left(\cos\frac{t}{2^{k+1}}\right)^{r+1}. \tag{4.7}$$

The Fourier transform of $up(x)$ is representable as $F(t) = \prod\limits_{k=1}^{\infty}\left(\cos t/2^{k+1}\right)^k$.

If each space L_n has several generative functions, then their Fourier transforms are represented by means of infinite matrix products of periodic functions.

In the case of Daubechies W-system, the even generating function satisfies the condition

$$\widetilde{\varphi}_0(t) = \begin{cases} 1 & \text{inside } \left[-\frac{2}{3}\pi;\frac{2}{3}\pi\right], \\ 0 & \text{outside } \left[-\frac{4}{3}\pi;\frac{4}{3}\pi\right]. \end{cases} \tag{4.8}$$

If $\mu_0(t) - 2\pi$ is a periodic expansion of the function $\widetilde{\varphi}_0(t)$ from the interval $[-\pi, \pi]$ onto the whole real axis R, then $\widetilde{\varphi}_0(t) = \prod\limits_{k=0}^{\infty}\mu_0\left(t\,2^{-k}\right)$.

Therefore, if we use Mayer's approach due to the choice of $\varphi_0(x) = \varphi_n\left(2^n x\right)$ when the function $\widetilde{\varphi}_0(x)$ is arbitrary on the interval $(2\pi/3, 4\pi/3)$, we obtain the nested sequence of spaces L_n generated by translations $\varphi_0(2^n x - k)$; i.e., the condition $L_n \subset L_{n+1}$ is fulfilled. This allows us to construct an orthogonal W-system as the final result.

To realize the second stage or to generate orthogonal spaces W_n corresponding to different levels with respect to the spatial resolution and frequency ranges, we can construct in a standard way an orthogonal basis consisting of translations in the space L_n. Further, we should orthogonally project generative functions of the space L_{n+1} onto the space L_n by dividing them into projections with respect to the orthogonal basis of translations in L_n. Then, the difference between generative functions for spaces L_{n+1} and their projections will form generative functions for spaces W_n. In specific situations, we can solve the problem of constructing spaces W_n more easily by using finite sums instead of series.

4.4. Examples of W-systems

Let us consider concrete examples. In regard to the W-systems of the first type, let us turn to spline W-systems which were the first historically. As the space L_n, we will

take the space of polynomial splines of degree r with defect d defined on the uniform mesh kh_n, where the ratio h_n/h_{n+1} must be integer for condition (i) to be satisfied, i.e., $L_n \subset L_{n+1}$. As a rule, $h_n = 2^{-n}$. This means that a function $f(x) \in L_n$ is an algebraic polynomial of order no greater than r on each interval $(kh_n, (k+1)h_n)$; moreover, $f(x)$ has $(r-d)$ continuous derivatives; i.e., polynomials on neighboring intervals are such that their derivatives up to the order $(r-d)$ are equal at the boundary points of the intervals. Splines of zero defect are algebraic polynomials. We may omit the case of $d = 0$, because the condition $\int\limits_{-\infty}^{\infty} |f(x)|^2\, dx < +\infty$ is required. Splines of defect 1 are also called natural splines.

Assume that $r = 0$, $d = 1$, $h_n = 2^{-n}$, $n = 0, 1, \ldots$. Then, L_n is a space of piecewise constant functions possibly having jumps at the points $k\,2^{-n}$. The space L_n is generated by the functions $\varphi_n\,(x - 2^n k)$, where

$$\varphi_n(x) = \begin{cases} 1, & \text{if} \quad 0 \leqslant x < 2^{-n}, \\ 0, & \text{if} \quad 0 < x \text{ or } x > 2^{-n}. \end{cases} \tag{4.9}$$

In this case, the functions $\varphi_n\,(x - 2^n k_1)$ and $\varphi_n\,(x - 2^n k_2)$ are orthogonal when $k_1 \neq k_2$. To construct the space W_{n+1}, we project (orthogonally) the function $\varphi_{n+1}(x)$ onto $\varphi_n\,(x - 2^n k)$. The only function different from zero is $\varphi_{n+1}(x) = \varphi_n(x)/2$. Therefore, an orthogonal complement W_{n+1} for the space L_n in L_{n+1} is generated by a translation $\psi_{n+1}\,(x - 2^{-n})$ of the function

$$\psi_{n+1}(x) = \varphi_{n+1}(x) - \varphi_n(x)/2 = \left(\varphi_{n+1}(x) - \varphi_n\left(x - 2^{-n-1}\right)\right)/2, \tag{4.10}$$

because $\varphi_n(x) = \varphi_{n+1}(x) + \varphi_{n+1}\left(x - 2^{-n-1}\right)$.

Therefore, the orthogonal basis in the space $W_0 = L_0$ consists of functions $\varphi_0(x - k)$ and, in the spaces W_n $(n \geqslant 1)$, of functions

$$\begin{gathered} \psi_1(x) = [\varphi_0(x) + \varphi_0(x - 1/2)]/2, \\ \psi_n\left(x - k\,2^{-n-1}\right) = \psi_1\left(2^{n-1}x - k\right). \end{gathered} \tag{4.11}$$

Thus, the W-system is constructed. Note that it is generated by translations and dilations of two functions $\varphi_0(x)$ and $\psi_1(x)$ (father wavelet and mother wavelet). In this case, both functions $\varphi_0(x - k)$ and functions $\psi_n\,(x - k2^{-n})$ were found to be orthogonal; therefore, additional orthogonalization in spaces W_n is not required. The constructed W-system is a classical Haar system. Usually, the classical system

$$\begin{gathered} h_{n,k}(x) = 2^{n/2+1}\psi_0\,(2^n x - k), \quad n \geqslant 1, \\ h_{0,k}(x) = \varphi_0(x - k) \end{gathered} \tag{4.12}$$

is considered.

Denote this system by WS_0. It is localized optimally in the time domain but, in the frequency domain, its localization is not so good. In the first place, the Fourier transforms of functions from WS_0 decrease with the rate as $|t|^{-1} \to \infty$. The function $\varphi_0(x)$ is the simplest triangular window, but its shortcomings are well-known. Secondly, if we split the frequency domain into the octaves $T_n = 2^n\pi \leqslant t \leqslant 2^{n+1}\pi$, then each octave will contain, in a certain sense, only Fourier transforms for functions of WS_0 relating to the level n of the form $\psi_0(2^n x - k)$, i.e., translations of one function $\psi_0(2^n x)$, differing from one anotherin the absolute value with the factor $\exp(ik\,t)$. Therefore, it is difficult to predetermine immediately the behavior of an «instantaneous spectrum» at a presupposed point x_0 and in a subinterval of the octave T_n using the magnitude

of Fourier–Haar coefficients $\alpha_{n,k} = \int\limits_{-\infty}^{\infty} f(x)h_{n,k}(x)dx$ for the function $f(x)$. Further transformations of these coefficients are required.

As is well known, in the space of piecewise constant functions defined on a uniform mesh, there exists another orthogonal basis different from the orthogonal Haar basis, possessing the ideal spatial (temporal) localization. It is the Walsh basis, which is more popular in signal processing. The Walsh functions are not localized in the time domain, but they are well localized in the frequency domain. Therefore, in signal processing with the use of the W-systems, one should turn to hybrid W-systems obtained by the use of local Walsh transformations for the Haar system, in particular, when it is required to obtain information on an instantaneous spectrum distribution with respect to different ranges of a single octave (digital music and speech processing: in devices transforming phonograms into notes and text).

In the simplest case, instead of using the Haar system, one can replace each pair of functions $h_{n,2k-1}(x)$ and $h_{n,2k}(x)$ by the following pair:

$$\begin{aligned} w^1_{n,k}(x) &= (h_{n,2k-1}(x) + h_{n,2k}(x))/\sqrt{2}\,, \\ w^2_{n,k}(x) &= (h_{n,2k-1}(x) - h_{n,2k}(x))/\sqrt{2}\,. \end{aligned} \tag{4.13}$$

Here, the spatial resolution decreases by a factor of two, while the frequency resolution is doubled inside the octave. When local Fourier analysis is realized, it becomes possible to distinguish high and low pitches inside a single octave.

Some investigators have composed libraries of W-systems with respect to the required number of pitches distinguishable inside a single octave [5].

The shortcoming of the WS_0 system is that piecewise constant functions are not best suited for approximation of smooth functions. If the signal under processing is smooth, then the use of smoother W-systems probably allows one to obtain better quality for the same number of levels in multiresolution analysis or to use a fewer number of levels to provide the required quality. In other words, if the frequency of the signal to be processed implies that the difference $\Delta x_k = x_{k+1} - x_k$ is small as compared with x_k $\left|\dfrac{\Delta x_k}{x_k}\right| \ll 1$, then the appropriateness of using smoother W-systems should be considered.

Another, not so trivial example of a W-system allowing us to observe the details of constructing a spline W-system (and also atomic W-systems) is the system WS_1 obtained with $r = 1$ and $d = 1$ for the same mesh of width $h_n = 2^{-n}$. In this case, the splines are continuous polygonal lines (piecewise linear functions). The space L_0 is generated by the translations $\varphi_0(x-k)$ of a compactly supported function (triangular window)

$$\varphi_0(x) = \begin{cases} 1 - |x|\,, & |x| \leqslant 1, \\ 0, & |x| > 1, \end{cases} \tag{4.14}$$

and the space L_n for $n \geqslant 1$ is generated by the translations $\varphi_n\left(x - k2^{-n}\right)$, where $\varphi_n(x)$ is a compression of the function $\varphi_0(x)$: $\varphi_n(x) = \varphi_0(2^n x)$.

This implies that, for constructing orthogonal complements W_n for L_{n-1} in L_n, it is sufficient to find the function $\psi_0(x)$ in the form

$$\Psi_0(x) = \sum_{j=M_1}^{M_2} c_j \varphi_0(x-j), \tag{4.15}$$

which is orthogonal to all the functions

$$\varphi\left(x/2 - k\right). \tag{4.16}$$

It is sufficient to take $\mu_1 = -3$ and $\mu_2 = 1$ and consider $\psi_0(x)$ in the form

$$\psi_0(x) = \alpha\varphi_0(x+3) + \beta\varphi_0(x+2) + \gamma\varphi_0(x+1) + \beta\varphi_0(x) + \alpha\varphi_0(x-1). \tag{4.17}$$

Then, the orthogonality of $\psi_0(x)$ to $\varphi_0(x/2+2)$ and $\varphi_0(x/2+1)$ must be ensured, because its orthogonality to $\varphi_0(x/2)$ and $\varphi_0(x/2-1)$ follows from the symmetry and the orthogonality of $\psi_0(x)$ to other functions $\varphi_0(x/2+k)$ is fulfilled automatically because $\psi_0(x)\,\varphi_0(x/2-k) \equiv 0$.

After computations we obtain

$$\psi_0(x) = \varphi_0(x+3) + 6\varphi_0(x+2) + 10\varphi_0(x+1) - 6\varphi_0(x) + \varphi_0(x-1). \tag{4.18}$$

4.5. Methods for Constructing Orthogonal Bases

Thus, the orthogonal spaces W_n have been constructed. Now, in spaces $W_0 = L_0$ and W_n, $n \geqslant 1$, we must construct orthogonal bases consisting of translations of some generative functions $\mu_0(x)$ for W_0 and $\nu_n(x) = \nu_0(2^n x)$, $n \geqslant 1$, of the form $\mu(x-k)$, $\nu_n\left(x - k2^{-n+1}\right)$. To do this, it is necessary to consider methods for constructing orthogonal bases consisting of translations of a single function in spaces generated by translations of a single function (nonorthogonal).

Let the space L be generated by translations $\varphi_0(x-k)$ of a function $\varphi(x)$. It is necessary to find the function $\mu(x)$ such that their translations $\mu(x-k)$ are orthogonal and generate L.

Firstly, let us consider the case of a compactly supported function $\varphi(x)$ vanishing outside the interval $[-a, a]$.

Assume that $a_{ik} = \int\limits_{-\infty}^{\infty} \varphi(x-i)\varphi(x-k)dx$. Then, $a_{ik} = a_{ki}$ (in the case of a complex-valued $\varphi(x)$), $a_{ik} = \int\limits_{-\infty}^{\infty} \varphi(x-i)\overline{\varphi(x-i)}dx$, and $a_{ik} = \overline{a}_{ik}$. Note that $a_{ik} = a_{i-k}$. The function $\mu(x)$ can be presented in the form

$$\mu(x) = \sum_{l=-\infty}^{\infty} x_l\varphi(x-l), \quad \text{with} \quad \sum_{l=-\infty}^{\infty} x_l^2 < +\infty.$$

Then, the orthogonality conditions give us the system of equations

$$\sum_{i,k=-\infty}^{\infty} a_{i-k}x_i x_{k+s} = \begin{cases} 0 & \text{if} \quad s \neq 0, \\ 1 & \text{if} \quad s = 0. \end{cases} \tag{4.19}$$

Compactness of the function $\varphi(x)$ yields the coefficients $a_t = 0$ when $|t| > M$, where $M = [2a]$. Then, system (4.19) can be rewritten in the form

$$\sum_{t=-M}^{M} a_t \sum_{i=-\infty}^{\infty} x_i x_{i-t-s} = \begin{cases} 0 & \text{if} \quad s \neq 0, \\ 1 & \text{if} \quad s = 0. \end{cases}$$

Suppose that

$$y_k = \sum_{i=-\infty}^{\infty} x_i x_{i-k}.$$

If $y_k = y_{-k}$, we obtain the system

$$\sum_{t=-M}^{M} a_t y_{t+s} = \begin{cases} 0 & \text{if} \quad s \neq 0, \\ 1 & \text{if} \quad s = 0. \end{cases} \tag{4.20}$$

Its solution such that $\sum\limits_{k=-\infty}^{\infty} y_k^2 < +\infty$ can be obtained in the form

$$y_k = \sum_{l=1}^{M} c_l \lambda_l^{|k|}, \tag{4.21}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_M$ are the roots of the characteristic equation

$$\sum_{s=0}^{2M} a_{S-M} \lambda^s = 0, \tag{4.22}$$

such that $|\lambda_l| < 1$, $l = 1, 2, \ldots, M-1$.

Now, we can find coefficients x_k from the system

$$\sum_{i=-\infty}^{\infty} x_i x_{i-k} = y_k \tag{4.23}$$

either in the form

$$x_i = \begin{cases} \sum\limits_{l=1}^{M} d_l \lambda_l^i & \text{if} \quad i \geqslant 0, \\ 0 & \text{if} \quad i < 0, \end{cases} \tag{4.24}$$

(for the right-hand solution) or as

$$x_i = \begin{cases} \sum\limits_{l=1}^{M} d_l \lambda_l^{-i} & \text{if} \quad i \geqslant 0, \\ 0 & \text{if} \quad i < 0, \end{cases} \tag{4.25}$$

(for the left-hand solution).

Let apply this technique in the case of spaces of polygonal lines.

Firstly, consider the construction of an orthogonal basis in the space L_0. In this case, $a_1 = a_{-1} = 1/6$, $a_0 = 2/3$, and $a_k = 0$ when $|k| > 1$.

If we multiply the function $\varphi(x)$ by $6^{1/2}$, we can further assume that $a_0 = 4$, $a_1 = 1$, and $a_{\pm 1} = 1$.

The characteristic equation in this case has the form

$$\lambda^2 + 4\lambda + 1 = 0.$$

Its root $\lambda_1 = -2 + \sqrt{3}$ has a modulus less than 1, and, correspondingly,

$$y_k = \left(-2 + \sqrt{3}\right)^{-|k|}.$$

The coefficients x_i are

$$x_i = \begin{cases} \left(c\left(-2 + \sqrt{3}\right)_1^i\right) & \text{if} \quad i \geqslant 0, \\ 0 & \text{if} \quad i \geqslant 0. \end{cases}$$

Substituting x_i into the equation $\sum\limits_{i=-\infty}^{\infty} x_i x_{i-k} = y_k$, we obtain

$$\sum_{i=-\infty}^{\infty} x_i x_{i-k} = \begin{cases} \sum\limits_{i\geqslant k}^{\infty} c^2 \lambda_1^{2i-k} = \dfrac{c^2 \lambda_1^k}{1-\lambda_1^2} = \lambda_1^k \quad \text{for} \quad k \geqslant 0, \\ \sum\limits_{i=0}^{\infty} c^2 \lambda_1^{2i-k} = \dfrac{c^2 \lambda_1^{-k}}{1-\lambda_1^2} = \lambda_1^{-k}. \end{cases} \tag{4.26}$$

It follows that $c = \sqrt{1-\lambda_1^2} = \sqrt{4\sqrt{3}-6}$.

Thus,

$$x_i = \begin{cases} \left(-2+\sqrt{3}\right)^i & \text{if} \quad i \geqslant 0, \\ 0 & \text{if} \quad i < 0, \end{cases} \tag{4.27}$$

and

$$\mu(x) = \sum_{i=0}^{\infty} \frac{\left(-2+\sqrt{3}\right)^i}{\sqrt{4\sqrt{3}-6}} \varphi_0(x-i).$$

This is a right-hand wavelet. A left-hand wavelet has the form

$$\mu(x) = \sum_{i=-\infty}^{0} \frac{\left(-2+\sqrt{3}\right)^{-i}}{\sqrt{4\sqrt{3}-6}} \varphi_0(x-i).$$

Now, let us turn to the construction of orthogonal bases consisting of translations $\nu\left(x-k2^{-n+1}\right)$ of functions the $\nu_n(x)$in spaces W_n when $n \geqslant 1$.

It is sufficient to find a function $\nu(x)$ in the form

$$\nu(x) = \sum_{l=-\infty}^{\infty} x_l \psi(x-2l), \qquad \sum_{l=-\infty}^{\infty} x_l^2 < +\infty, \tag{4.28}$$

such that its translations $\nu(x-2k_1)$ and $\nu(x-2k_2)$ for $k_1 \neq k_2$ are orthogonal. Then, one can assume $\nu_n(x) = \nu_n(x2^n)$.

Applying the aforementioned approach with the function $\sqrt{3}(\psi(x))/2$ instead of $\psi(x)$, we obtain $a_0 = 54$, $a_{\pm 1} = 10$, $a_{-1} = 10$, $a_{\pm 2} = -1$, and $a_k = 0$ at $|k| > 2$.

Here, the characteristic equation is written as

$$\lambda^4 - 10\lambda^3 - 54\lambda^2 - 10\lambda + 1 = 0, \tag{4.29}$$

the absolute values of its roots are less than unity and equal to $\lambda_1 = 7 - 4\sqrt{3}$, $\lambda_2 = -2 + \sqrt{3}$.

Substituting (4.21) into (4.20), we obtain the following system in order to find c_1 and c_2:

$$54(c_1+c_2) + 20(c_1\lambda_1 + c_2\lambda_2) - 2\left(c_1\lambda_1^2 + c_2\lambda_2^2\right) = 1,$$
$$10(c_1+c_2) + 53(c_1\lambda_1 + c_2\lambda_2) + 10\left(c_1\lambda_1^2 + c_2\lambda_2^2\right) - \left(c_1\lambda_1^3 + c_2\lambda_2^3\right) = 0.$$

This implies that

$$c_1\left(54 + 20\lambda_1 - 2\lambda_1^2\right) + c_2\left(54 + 20\lambda_2 - 2\lambda_2^2\right) = 1,$$
$$c_1\left(10 + 53\lambda_1 + 10\lambda_1^2 - \lambda_1^3\right) + c_2\left(10 + 53\lambda_2 + 10\lambda_2^2 - \lambda_2^3\right) = 0$$

and

$$\begin{gathered} c_1 = \left(10 + 53\lambda_2 + 10\lambda_2^2 - \lambda_2^3\right)/\Delta, \\ c_2 = \left(10 + 53\lambda_1 + 10\lambda_1^2 - \lambda_1^3\right)/\Delta, \\ \Delta = \left(54 + 20\lambda_1 - 2\lambda_1^2\right)\left(10 + 53\lambda_2 + 10\lambda_2^2 - \lambda_2^3\right) - \\ - \left(54 + 20\lambda_2 - 2\lambda_2^2\right)\left(10 + 53\lambda_1 + 10\lambda_1^2 - \lambda_1^3\right) \neq 0. \end{gathered} \tag{4.30}$$

The coefficients x_k are obtained in the form

$$x_k = \begin{cases} d_1\lambda_1^k + d_2\lambda_2^k, & k \geqslant 0, \\ 0, & k < 0. \end{cases}$$

Here, we can write the following system for d_1 and d_2:

$$\begin{cases} \dfrac{d_1^2}{1-\lambda_1^2} + \dfrac{d_1 d_2}{1-\lambda_1\lambda_2} = c_1, \\ \dfrac{d_2^2}{1-\lambda_2^2} + \dfrac{d_1 d_2}{1-\lambda_1\lambda_2} = c_2. \end{cases} \tag{4.31}$$

This, in turn, can be reduced to the biquadratic equation

$$d_1^4\left[\frac{1}{1-\lambda_1^2} - \frac{(1-\lambda_1\lambda_2)^2}{\left(1-\lambda_1^2\right)^2\left(1-\lambda_2^2\right)}\right] + \\ + d_1^2\left[\frac{2(1-\lambda_1\lambda_2)^2}{\left(1-\lambda_1^2\right)\left(1-\lambda_2^2\right)} - (c_1 - c_2)\right] + \frac{c_1^2(1-\lambda_1\lambda_2)^2}{1-\lambda_2^2} = 0. \tag{4.32}$$

We obtain a right-hand wavelet $v_r(x)$. To obtain a left-hand wavelet $v_l(x)$, we use

$$x_k = \begin{cases} d_1\lambda_1^{-k} + d_2\lambda_2^{-k}, & k \geqslant 0, \\ 0, & k < 0, \end{cases}$$

or assume $v_l(x) = v_r(-x)$.

There is another method based on the use of the Fourier transform to find an orthogonal basis consisting of translations of a certain function $\mu(x)$ in the space L generated by nonorthogonal translations of the function $\varphi(x)$.

Let $\widetilde{\varphi}(t)$ be the Fourier transform of the function $\varphi(x)$ and $\widetilde{\mu}(t)$ be the Fourier transform of the function $\mu(x)$. Therefore, if $\mu(x) = \sum\limits_{l=-\infty}^{\infty} x_l\varphi(x-l)$, $\sum\limits_{l=-\infty}^{\infty} x_l^2 < +\infty$, then

$$\widetilde{\mu}(t) = \widetilde{\varphi}(t)\sum_{l=-\infty}^{\infty} x_l e^{ilt} = b(t)\widetilde{\varphi}(t), \tag{4.33}$$

where $b(t) = \sum\limits_{l=-\infty}^{\infty} x_l e^{ilt} \in L_2(R)$.

The orthogonality condition for functions $\mu(x-k)$ and $\mu(x-m)$, when $k \neq m$, implies that

$$\int_{-\infty}^{\infty} |\widetilde{\mu}(t)|^2 e^{i(k-m)t} dt \quad \text{for} \quad k \neq m, \tag{4.34}$$

or, taking into account the 2π-periodicity of the function $\exp(i(k-m)t)$,

$$\int_{-\pi}^{\pi} \sum_{s=-\infty}^{\infty} e^{i(k-m)t} \left|\widetilde{\mu}(t-2\pi s)\right|^2 dt = 0, \quad \text{when} \quad k \neq m. \tag{4.35}$$

Here, $\sum\limits_{s=-\infty}^{\infty} \left|\widetilde{\mu}(t-2\pi s)\right|^2 = |b(t)|^2 \sum\limits_{s=-\infty}^{\infty} \left|\widetilde{\varphi}(t-2\pi s)\right|^2$ due to the 2π-periodicity of the function $b(t)$.

From (4.35) it follows that

$$|b(t)|^2 \sum_{s=-\infty}^{\infty} \left|\widetilde{\varphi}(t-2\pi s)\right|^2 \equiv A \neq 0. \tag{4.36}$$

If we assume that $A = 1$, then $|b(t)|^2 = 1/\Phi(t)$, where

$$\Phi(t) = \sum_{s=-\infty}^{\infty} \left|\widetilde{\varphi}(t-2\pi s)\right|^2.$$

The function $\Phi(t) \geqslant 0$ has a period 2π.

If $\Phi(t) > 0$ and $b(t) = 1/\sqrt{\Phi(t)}$, then the coefficients x can be found by means of the Fourier-integral expansion of $b(t)$ as

$$x_l = \frac{1}{2\pi} \int_{-\pi}^{\pi} b(t) e^{ilt} dt.$$

In particular, if $\varphi(x)$ is even, then $\Phi(t)$ is even too. In this case, $x_l = \frac{1}{2\pi} \int\limits_{-\pi}^{\pi} b(t) \cos lt\, dt = \frac{1}{\pi} \int\limits_{0}^{\pi} B(t) \cos lt\, dt$. Here, the wavelet function $\mu(x)$ is also even in contrast to the earlier constructed right-hand and left-hand W-functions that are equal to zero on the half-line $x < 1$ or $x > -1$.

In the case of a polygonal line space L_0, $\widetilde{\varphi}_0(t) = \left(\frac{\sin t/2}{t/2}\right)^2$ and function $\Phi(t)$ has the form

$$\Phi(t) = \sin^4 \frac{t}{2} \sum_{s=-\infty}^{\infty} \frac{1}{(t/2 - 2s\pi)^2} = \frac{2+\cos t}{3}. \tag{4.37}$$

For the even W-function, we have

$$x_l = \frac{1}{6\pi} \int_{-\pi}^{\pi} \frac{\cos lt}{\sqrt{2+\cos t}} dt = \frac{1}{3\sqrt{2}\,\pi} \int_{0}^{\pi} \cos lt \left(1 + \frac{1}{2}\cos t\right)^{-1/2} dt,$$

which implies that

$$x_l = \frac{1}{3\sqrt{2}\,\pi} \int_{0}^{\pi} \cos lt \left(1 + \sum_{k=1}^{\infty} \frac{(-1)^k (2k-1)!!}{k! 2^{2k}} \cos^k t\right)^{-1/2} dt.$$

Taking into account the property

$$\cos^k t = (1/2)^{k-1} \left(\cos kx + c_n^1 \cos(k-2)x + c_n^2 \cos(k-4)x + \ldots\right)$$

and the orthogonality of $\cos n\,x$, we obtain

$$x_l = \frac{1}{12\sqrt{2}} \sum_{k=l}^{\infty} \frac{(-1)^k (2k-1)!!}{l!(k-l)!2^{3k}}. \tag{4.38}$$

We can also proceed in the following way. The positive trigonometric polynomial $\Phi(t)$ is representable in the form

$$\Phi(t) = P(t)\overline{P(t)} = |P(t)|^2,$$

where $P(t)$ is a complex-valued trigonometric polynomial. Suppose that

$$P(t) = \sqrt{2+\sqrt{3}}\ \left(1 - \lambda_1 e^{it}\right),$$

where $\lambda_1 = -2 + \sqrt{3}$.

Then

$$b(t) = 1/P(t),$$

when

$$b(t) = \frac{1}{\sqrt{2+\sqrt{3}}} \cdot \frac{1}{1-\lambda_1 e^{it}} = \frac{1}{\sqrt{2+\sqrt{3}}} \sum_{l=0}^{\infty} \lambda^l e^{ilt},$$

i.e., $x_l = \dfrac{\lambda^l}{\sqrt{2+\sqrt{3}}}$ for $l \geqslant 0$ and $x_l = 0$ for $l < 0$. It is clear that we can take

$$P(t) = \sqrt{2+\sqrt{3}}\ \left(1 - \lambda_1 e^{-it}\right) \tag{4.39}$$

to have

$$x_l = \begin{cases} \lambda_1^{-l} & \text{for} \quad l \leqslant 0, \\ 0 & \text{for} \quad l > 0. \end{cases}$$

At last, we have obtained the right-hand and left-hand wavelets with the help of the Fourier transform.

Which functions are better in practice: symmetric W-functions or one-sided ones? Which of the methods of obtaining them is simpler? We leave these questions to the reader's judgement.

The function $v(x)$ whose compressed translations generate an orthogonal basis in spaces W_n for $n \geqslant 1$ also can be constructed in a similar way. In this case, one can obtain both symmetric and one-sided W-functions. Note that the constructed W-functions have an exponential rate of decay as $|t| \to \infty$ (one-sided functions even equal zero on the half-line). In the nonsymmetric case, the rate of decay is determined by the root of characteristic equation (whose modulus is less than unity) that is maximal with respect to tha absolute value: $\lambda_1 = -2 + \sqrt{3} = -0.2679$.

In the symmetric case, the integrals $1/2\pi \int\limits_{-\pi}^{\pi} 1/\sqrt{\Phi(t)}\,\cos lt\,dt$ still must be calculated. However, since the function $1/\sqrt{\Phi(t)}$ is analytic, they must decrease exponentially.

A W-system constructed of first-order splines is called a WS_1 system. In reality, depending on the choice of either one-sided or symmetric W-functions, we deal with different WS_1 systems. The WS_1 system is the simplest example of the Stromberg and Battle–Lemarie W-systems. Functions of a WS_1 system are localized in the frequency domain with the decrease rate of $|t|^{-2}$ as $|t| \to \infty$ and, in the time domain, provide us an

approximation error no better than ch^2, where h is the mesh width of a polygonal line. As we can see, even the construction of this simplest W-system is more complicated than the construction of the Haar system.

In the symmetric case, the Fourier transform of the function $v(x)$ whose compressed translations $v\left(2^n x - 2k\right)$ form an orthogonal basis in the space W_n can be represented as

$$\widetilde{v}(t) = \widetilde{\varphi}_0(t)Q(t) = \left(\frac{\sin t/2}{t/2}\right)^2 Q(t),$$

where

$$Q(t) = \frac{\cos^2 t - 3\cos t + 2}{\sqrt{14 + 5\cos 2t - \cos^2 2t}}. \tag{4.40}$$

In the nonsymmetric case,

$$\widetilde{v}(t) = \left(\frac{\sin t/2}{t/2}\right)^2 \frac{\cos^2 t - 3\cos t + 2}{P(t)}. \tag{4.41}$$

Here, $P(t)$ is an arbitrary trigonometric polynomial (more precisely, a polynomial with respect to $\exp(i2t)$ and $\exp(-i2t)$) such that

$$P(t)\overline{P}(t) = |P(t)|^2 = 14 + 5\cos 2t - \cos^2 2t.$$

Therefore, in the symmetric case,

$$v(x) = \sum_{l=-\infty}^{\infty} x_l \varphi_0(x - l), \tag{4.42}$$

where

$$x(l) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{\cos^2 t - 3\cos t + 2}{\sqrt{14 + 5\cos 2t - \cos^2 2t}} \cos lt\, dt,$$

and, in the nonsymmetric case,

$$x(l) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{\cos^2 t - 3\cos t + 2}{P(t)} e^{-ilt} dt, \tag{4.43}$$

where $P(t)$ can be taken in the form

$$P(t) = \alpha\left(1 - \left(\sqrt{3} - 2\right)e^{i2t}\right)\left(1 - \left(7 - 4\sqrt{3}\right)e^{i2t}\right),$$
$$\alpha = \left(3\sqrt{6} + 5\sqrt{2}\right)/4.$$

In order to calculate (4.43), we represent $1/P(t)$ as

$$\frac{1}{P(t)} = \frac{1}{a}\left\{\frac{\left(3 + \sqrt{3}\right)/6}{1 - \left(\sqrt{3} - 2\right)e^{i2t}} + \frac{\left(2 - \sqrt{3}\right)/\left(3 - \sqrt{3}\right)}{1 - \left(7 - 4\sqrt{3}\right)e^{i2t}}\right\}. \tag{4.44}$$

In the symmetric case, for $k \geqslant 0$,

$$x_{2k} = \frac{1}{a}\left(\frac{3+\sqrt{3}}{6}\left(2\left(\sqrt{3}-2\right)^{k}+\frac{1}{2}\left(\sqrt{3}-2\right)^{k-1}+\frac{1}{2}\left(\sqrt{3}-2\right)^{k+1}\right)\right)+$$
$$+\frac{1}{a}\left(\frac{2-\sqrt{3}}{3-\sqrt{3}}\left(2\left(7-4\sqrt{3}\right)^{k}+\frac{1}{2}\left(7-4\sqrt{3}\right)^{k-1}+\left(7-4\sqrt{3}\right)^{k+1}\right)\right)=$$
$$=\frac{1}{a}\frac{2-\sqrt{3}}{3-\sqrt{3}}\left(7-4\sqrt{3}\right)^{k-1}\left(63-36\sqrt{3}\right).$$

$$x_{2k+1} = \frac{1}{a}\left(\frac{-3-\sqrt{3}}{2}\left(\sqrt{3}-2\right)^{k}\left(\sqrt{3}-1\right)-\right.$$
$$\left.-\frac{2-\sqrt{3}}{3-\sqrt{3}}3\left(7-4\sqrt{3}\right)^{k}\left(8-4\sqrt{3}\right)\right). \quad (4.45)$$

To calculate x_k in the symmetric case, we use the formula

$$\frac{1}{\sqrt{14+5\cos 2t-\cos^2 2t}} = \frac{1}{\sqrt{14}}\sum_{k=0}^{\infty}\frac{(-1)^k(2k-1)!!}{k!2^k}\times$$
$$\times\left(\frac{\cos 2t}{7}\right)^2\sum_{s=0}^{\infty}\frac{(-1)^s(2s-1)!!}{s!2^s}\left(\frac{\cos 2t}{2}\right)^s, \quad (4.46)$$

which, being substituted into (4.42), will allow us to express x_k in the form of a series.

4.6. Wavelets based on Hermitian Splines

The systems designed in the previous, naturally, have a low frequency resolution for evaluating the instantaneous spectrum in the LFA. Only one function of the W-system concentrated in the vicinity of a fixed point x_0 in the time domain substantially contributes in each octave of the frequency domain. If we want to have a W-system whose expansion coefficients for a musical signal allow us to obtain immediately the score of this signal, then the W-system must have at least twelve functions concentrated in the vicinity of the given point in the time domain. These functions must have their Fourier transforms situated mainly inside one octave and in different places. In order to reconstruct the score for recording of a symphonic orchestra, a more flexible W-system is necessary. The same can be said about W-systems used for the color image analysis. Thus, the classical W-systems generated by translations of one single function compressed by a factor of 2^n are not suitable for these purposes (without proper modifications). If, instead of a compression coefficient of 2, we use a compression coefficient $(1+1/m)$ or $2^{1/m}$, the situation will be improved. As is known, human ear and brain can perceive identically a melody that is raised or lowered not only by a whole octave but also by several tones. Unfortunately, in the case of spline W-systems on a uniform mesh, the compression coefficient is equal to h_n/h_{n+1} and must be integer.

To improve a W-system in this sense, i.e., to achieve a higher frequency resolution, one can use in the LFA the sequence of spaces $L_n^{(m)}$, $n = 0, 1, \ldots, L_n^{(m)} \subset L_{n+1}^{(m)}$, where the space $L_n^{(m)}$ is generated by translations of functions $\varphi_{n,k}(x)$ having the form

$$\varphi_{n,k}(x) = \varphi_0(2^n x)\, x^k, \quad k = 0, 1, 2, \ldots, m, \quad (4.47)$$

with the function $\varphi_0(x)$ taken from (4.14) with a step of 2^{-n}.

This means that we use splines of degree m and defect $m-1$, i.e., so called Hermitian splines. To construct orthogonal spaces $W_n^{(m)}$ such that

$$L_n^{(m)} = L_{n-1}^{(m)} \oplus W_n^{(m)}, \tag{4.48}$$

we will find $(m-1)$ functions $\psi_0(x), \ldots, \psi_m(x)$ of the form

$$\psi_s(x) = \sum_{l=-2}^{2} \sum_{k=0}^{m} \alpha_{lk}^{(s)} x^k \varphi_0(x-l) \tag{4.49}$$

from the condition of their orthogonality to the functions:

$$\begin{gathered} x^r\varphi_0(x/2+2)\,,\ x^r\varphi_0(x/2+1)\,,\ x^r\varphi_0(x/2)\,, \\ x^r\varphi_0(x/2-1)\,, \quad r=0,1,\ldots,m. \end{gathered} \tag{4.50}$$

This gives us $4(m+1)$ homogeneous linear algebraic equations with respect to $5(m+1)$ unknowns α_{lk}, which will give $(m+1)$ linearly independent solutions $\alpha_{lk}^{(s)}$, $s=0,1,2,\ldots$ $\ldots, m$.

It remains to turn to the orthogonal bases $\{\mu_s(x-k),\ s=0,1,\ldots,m,\ k\ is\ integer\}$ in $L_0^{(m)} = W_0^{(m)}$ and $\{\nu_s(2^n x-2k),\ s=0,1,\ldots,m,\ k\ is\ integer\}$ in the spaces W_n, $n \geqslant 1$. The methods described above are applicable in this case too. Here, the function $\Phi(t)$ will be an $m \times m$ square matrix-function, and the expression $1/\sqrt{\Phi(t)}$ must be interpreted as $\left(\sqrt{\Phi(t)}\right)^{-1}$, where $\left(\sqrt{\Phi(t)}\right)^2 = \Phi(t)$. One can increase the resolution for LFA without leaving the space of polygonal lines.

Consider the simplest example. Let L_0 be the space of polygonal lines of the unit step, introduced above. Instead of the basis consisting of translations $\varphi_0(x-k)$ of functions $\varphi_0(x)$, we will take the basis consisting of translations of two functions, $\varphi_0(x-2k)$ and $\varphi_1(x-2k+1)$, with the step of 2, where

$$\varphi_1(x) = \varphi_0(x+1) - 4\varphi_0(x) + \varphi_0(x-1). \tag{4.51}$$

Then, for all integer k, l the functions $\varphi_0(x-2k)$ and $\varphi_0(x-2l+1)$ will be orthogonal, i.e.,

$$\int_{-\infty}^{\infty} \varphi_0(x-2k)\varphi_1(x-2l+1)dx = 0. \tag{4.52}$$

It is also obvious that $\varphi_0(x-2k)$ and $\varphi_0(x-2l)$ when $k \neq l$ are orthogonal. In the view of this fact, construction of the corresponding W-system is not difficult. Compared to the WS_1-system, the spatial (temporal) resolution is halved on the given level, but the resolution in each octave is doubled when LFA is realized.

Write the functions $\psi_1(x)$ and $\psi_2(x)$ whose translations $\psi_1(2^n x-4k)$ and $\psi_2(2^n x-4k)$ generate orthogonal the spaces W_n for $n \geqslant 1$:

$$\begin{gathered} \psi_1(x) = -\varphi_0(x+3) + 6\varphi_0(x+2) - 11\varphi_0(x+1) + 12\varphi_0(x) + \\ +6\varphi_0(x-2) - 11\varphi_0(x-1) - \varphi_0(x-3), \\ \psi_2(x) = \varphi_0(x+3) - 6\varphi_0(x+2) + 9\varphi_0(x+1) - \\ -9\varphi_0(x-1) + 6\varphi_0(x-2) - \varphi_0(x-3). \end{gathered} \tag{4.53}$$

To obtain a high decay rate for the Fourier transforms of the W functions as $|t| \to \infty$ and a higher rate of approximation by means of these functions, it is necessary to use smoother splines.

The system WS_0 has the decay rate of $|t|^{-1}$, and the system WS_1 has thedecay rate of $|t|^{-2}$. This means that the presence of a peak in one octave has a negative effect on perceiving a local spectrum picture in neighboring octaves. This shortcoming may be substantial for some problems of signal processing. If we use splines of degree r and minimum defect (natural splines), then, in spaces $L_n \subset L_{n+1}$, we can take bases of translations of compactly supported functions $B_r(2^n x - k)$, where $B_r(x)$ are the so-called Schoenberg B splines, i.e., compactly supported splines with minimal support.

They have the form

$$B_r(x) = \sum_{k=0}^{r+1} (-1)^k c_{r+1}^k (x-k)_+^r, \tag{4.54}$$

where

$$x_+^r = \begin{cases} x^r & \text{if} \quad x \geqslant 0, \\ 0 & \text{if} \quad x < 0. \end{cases}$$

The previously considered function $\varphi_0(x)$ is $B_1(x+1)$. The function $B_r(x)$ is equal to zero outside the interval $[0, r+1]$ and is positive when $0 < x < r+1$. Its Fourier transform is

$$\widetilde{B}_r(t) = e^{it\frac{r+1}{2}} \left(\frac{\sin t/2}{t/2} \right)^{r+1}. \tag{4.55}$$

To construct orthogonal spaces W_n, it is necessary to construct the function $\psi(x)$ in the form

$$\psi(x) = \sum_{k=0}^{M} \alpha_k B_r(x-k), \tag{4.56}$$

which is orthogonal to all functions $B_r(x/2 - m)$, where the number of unknowns is equal to α_k for a spline of odd degree $r = 2p+1$ (i.e., $M+1$ can be taken equal to $3r+2$), and in the case of even degree $r = 2p$, $M+1 = 3r+3$. The number of unknowns can be halved due to the symmetry condition. The function $\psi(x)$ will be compactly supported and equal to zero outside the interval [0,r+1+M]. Then, in spaces W_n, $n \geqslant 1$, the functions $\psi(2^n x - 2k)$ will form a basis. The conversion to an orthogonal W-system generated by the functions $\{\mu(x-k),\ \nu(2^n x - 2k),\ n \geqslant 1,\ k\ is\ integer\}$ is realized analogously to that for the WS_1system of the first-order splines.

In this case, the functions

$$\Phi_1(t) = \sum_{k=-\infty}^{\infty} \left| \widetilde{B}_r(t - 2k\pi) \right|^2 \tag{4.57}$$

and

$$\Phi_2(t) = \sum_{k=-\infty}^{\infty} |\psi(t - 2k\pi)|^2 \tag{4.58}$$

will be also positive trigonometric polynomials. This allows us to obtain explicitly the coefficients of expansions of $\mu(x)$ with respect to translations $B_r(x–k)$ and those of $\nu(x)$ with respect to $\psi(x–k)$. The corresponding W-system will be denoted by WS_r. Functions of this W-system have their Fourier transform decreasing with the rate $|t|^{-r-1}$ as $|t| \to \infty$. Thus, for example, the system WS_3 will be more noise-resistant than WS_1 with respect to a strong noise in neighboring octaves when the LFA is realized. To increase the resolution inside an octave when the LFA is realized, one should take some compactly supported functions with a greater support as compared with Bsplines, such that their Fourier transforms correspond to different parts of the octave. Then the W-system should be constructed according to the method described above.

One can also use splines of degree r and defect $d > 1$ so that both d and $r - d$ will be sufficiently large. Thus, if we use splines of degree 3 and defect 2, then the Fourier transforms for functions of a corresponding W-system will decrease with the rate $|t|^{-3}$ as $|t| \to \infty$, because these splines have a continuous first derivative and their second derivative has discontinuities of the first kind. On the other hand, the resolution of the LFA (LFAR) inside one octave still allows us to distinguish two tones (high and low). In this case, the functions whose translations generate L_0 (more exactly, $L_0^{3,2}$), denoted by $\varphi_1(x)$ and $\varphi_2(x)$, have the form

$$\varphi_1(x) = \begin{cases} 1 - 3x^2 + 2|x|\,x^2, & |x| \leqslant 1, \\ 0, & |x| > 1; \end{cases} \tag{4.59}$$

$$\varphi_2(x) = \begin{cases} x\,(1 - |x|)^2, & |x| \leqslant 1, \\ 0, & |x| > 1. \end{cases} \tag{4.60}$$

Instead of spline spaces, one can consider spaces generated by translations of arbitrary functions $\varphi_n(x)$ when the condition $L_n \subset L_{n+1}$ is fulfilled, which is essential in construction of orthogonal W-systems. As we mentioned above, functions $\varphi_n(x)$ must satisfy some conditions. Namely, they must be infinite convolutions of atomic measures. The condition of compactness implies that the Fourier transform $\widetilde{\varphi}_n(t)$ of the function $\varphi_n(x)$ must have the form

$$\widetilde{\varphi}_n(t) = \prod_{k=n+1}^{\infty} P_k(t), \tag{4.61}$$

where $P_k(t)$ are trigonometric polynomials.

For the specific choice of $P_k(t)$, the functions $\varphi_0(x-k)$ and $\psi_n\left(x - k2^{-n+1}\right)$ are found to be orthogonal projectors to L_{n-1}. Here, $\psi_n(x)$ are compactly supported functions whose translations generate the spaces W_n. These functions are orthogonal in L_n and form a W-system. This was also the case in the Haar system. The fact that W-systems exist and consist of compactly supported functions of any finite smoothness with $\psi_n(x) = \psi_n\left(2^n x\right)$ is a well-known result of wavelet theory. Such systems were constructed by Daubechies [3]. Since they are widely known, we will not present the corresponding formulas. Note only that, since these W-systems are generated on each level by translations of one function, their LFAR inside the octave is equal to zero. Naturally, they can be modified by means of combining translations of one level in blocks of m adjacent translations and subjecting W-functions in each block to an orthogonal transformation such that the Fourier transforms of the newly obtained functions will be within a frequency octave corresponding to the given level to be separated, i.e., allowing us to distinguish different parts of the octave. In other words, each of these modified W-functions must correspond to one of m subintervals of the given octave.

One cannot modify the Daubechies W-functions themselves but can realize some corresponding orthogonal transformations (especially, DPF) with blocks of Fourier coefficients with respect to an orthogonal Daubechies orthogonal W-system. In this case, to provide a higher frequency resolution, one has to take smoother Daubechies W-systems. Here, in the first place, Fourier transforms of corresponding W-functions will decrease faster as $|t| \to \infty$, and, therefore, the negative effect of noise from adjacent octaves of the evaluation of spectrum in this octave will be smaller. In the second place, the length of the Daubechies W-functions increases as their smoothness increases. Therefore, the number of functions for the specific level of the W-system whose supports contain the given point x_0 increases, which allows us to enlarge the block length in the DFT when the LFA is realized. Finally, we obtain once more the LFA using temporal windows of a special form.

4.7. Meyer Wavelets

Let us consider an approach for constructing W-systems opposite to some extent to the one described above, although formally it satisfies the scheme of multiresolution analysis. This approach was proposed by Meyer [4]. In contrast to the previously described methods of constructing W-systems by means of functions compactly supported in the time domain, his approach presupposes the use of functions with compactly supported Fourier transforms. In this case, the localization in the frequency domain is maximal, whereas the localization in the time domain can be neither maximal (such W-functions cannot be compactly supported) no exponential.

On the other hand, these W-systems of infinitely differentiable functions and even entire functions of exponential type can possess good approximation properties.

Let $S(t)$ be an infinitely differentiable odd function such that

$$S(1) = 0, \quad S^{(k)}(1) = 0, \tag{4.62}$$

where $k = 1, 2, \ldots$, and let the even functions $\alpha(t)$ and $\beta(t)$ be determined by the formulas

$$\alpha(t) = \begin{cases} 0 & \text{if} \quad |t| \geqslant 8\pi/3, \\ \pi/4\left(s\left(\frac{3}{\pi}|t| - 3\right) + 1\right) & \text{if} \quad 2\pi/3 \leqslant |t| \leqslant 4\pi/3, \\ \pi/4\left(s\left(3 - \frac{3}{2\pi}|t|\right) + 1\right) & \text{if} \quad 2\pi/3 \leqslant |t| \leqslant 8\pi/3, \end{cases} \tag{4.63}$$

$$\beta(t) = \begin{cases} 1 & \text{if} \quad |t| \leqslant 2\pi/3, \\ \pi/4\left(S\left(3 - \frac{3}{\pi}|t|\right) + 1\right) & \text{if} \quad 2\pi/3 \leqslant |t| \leqslant 4\pi/3, \\ 0 & \text{if} \quad |t| \geqslant 4\pi/3. \end{cases} \tag{4.64}$$

Let functions $v(x)$ and $\mu(x)$ be determined by the formulas

$$\mu(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{itx} \sin \beta(t) dt, \quad v(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{itx} \sin \alpha(t)\, e^{it/2} dt. \tag{4.65}$$

Then, the functions $\mu(x - k)$ and $v\,(2^n x - k)$, where $n = 0, 1, \ldots, k$ is integer, form Meyer's W-system. More precisely, we consider a continuum of W-systems corresponding to an arbitrary infinitely differentiable odd function $S(t)$ satisfying condition (4.62). A concrete W-system is obtained by means of a specific choice of $S(t)$. In [4], Meyer presents a graph for $v(x)$, although he does not mention to which $S(t)$ it corresponds.

Let explain how we achieve an orthogonality of this class of W-systems. In fact, it is necessary to verify the orthogonality of functions $\mu(x - k)$ and $v(x - l)$ for integer k and l, orthogonality of $\mu\,(x - k_1)$ and $\mu\,(x - k_2)$ for $k_1 \neq k_2$, $v\,(x - l_1)$ and $v\,(x - l_2)$ for $l_1 \neq l_2$, and, finally, orthogonality of $v(x - k)$ and $v(2x - l)$ for all integer k and l.

The orthogonality for other pairs of functions follows from the Parceval equality because products of their Fourier transforms are identically equal to zero.

Therefore, we only must verify the identities

$$\sum_{k=-\infty}^{\infty} |\widetilde{\mu}(t-2k\pi)|^2 = \sum_{k=-\infty}^{\infty} |\widetilde{v}(t-2k\pi)|^2 \equiv 1,$$
$$\sum_{k=-\infty}^{\infty} \widetilde{\mu}(t-2k\pi)\,\overline{\widetilde{v}(t-2k\pi)} \equiv 0, \tag{4.66}$$
$$\sum_{k=-\infty}^{\infty} \widetilde{v}(t-2k\pi)\,\overline{\widetilde{v}(t/2-k\pi)} \equiv 0,$$

where $\widetilde{\mu}(t) = \sin\beta(t)$ and $\widetilde{v}(t) = e^{it/2}\sin\alpha(t)$ are the Fourier transforms of the functions $\mu(x)$ and $v(x)$.

If $S_1(t)$ is an even function of the class C^∞ with the support $[-1,1]$ and $\int\limits_{-1}^{1} S_1(t)dt = 1$, then function $S(t)$ of the form

$$S(t) = -1 + 2\int_{-1}^{t} S_1(\tau)\,d\tau \tag{4.67}$$

satisfies condition (4.62).

In particular, one can use the function $\mathrm{up}\,(t)$ as $S_1(t)$, as it was done in [13]. Then, $S(t) = -1 + 2\,\mathrm{up}\,(t/2 - 1/2)$.

The fact that the function $\mathrm{up}\,(x)$ satisfies functional-differential equation (see chapter 1, section 1.1) can be used for evaluating functions $\mu(x)$ and $v(x)$ by means of (4.65).

Otherwise, the evaluation of functions $\mu(x)$ and $v(x)$, especially for large x, in the form of improper integrals of rapidly oscillating functions can be a difficult problem.

4.8. Kotelnikov–Shannon Wavelets

If we refuse of the infinite differentiability of function $S(t)$ presupposed by Meyer, then we obtain W-systems rapidly decreasing with the rate $|x|^{-m}$ as $|x| \to \infty$ when m is finite. As follows from [16–18], there exist W-systems consisting of elementary functions. For brevity, we shall call them WE-systems. Now, let describe these systems.

The simplest of such WE-systems, as simple as the Haar system and dual to it in some sense, is the orthogonal system WE_1, which will be called the Kotelnikov–Shannon system. It has the form

$$\mu(x-k), \quad v\,(2^n x - k)\,, \quad n = 0, 1, 2, \ldots, k \quad \text{is integer}, \tag{4.68}$$

where $\mu(x) = \sin\pi x/\pi x$ and $v(x) = \dfrac{\sin\pi x/2}{x}\cos 3\pi x/2$ are obtained for the following choice of $S(t)$,

$$S(t) = \begin{cases} -1 & \text{if} \quad t < 0, \\ 1 & \text{if} \quad t \geqslant 0, \end{cases} \tag{4.69}$$

if we use relations (4.63)–(4.65). The factor $\exp(it/2)$ in (4.65) can be omitted, otherwise we obtain a WE-system with $v_1(x) = v\,(x - 1/2)$.

Thus, for the WE_1-system,

$$\widetilde{\mu}(t) = \begin{cases} 1 & \text{if} \quad |t| \leqslant \pi, \\ 0 & \text{if} \quad |t| > \pi; \end{cases} \tag{4.70}$$

$$\widetilde{v}(t) = \begin{cases} 1 & \text{if} \quad \pi \leqslant |t| \leqslant 2\pi, \\ 0 & \text{if} \quad |t| < \pi \text{ or } |t| > 2\pi. \end{cases} \tag{4.71}$$

The Fourier transforms corresponding to the function $v(2^n x - k)$ are equal to $\exp(i2^{-n}kt)$ in the octave $2^n\pi < |t| < 2^{n+1}\pi$ and equal to zero outside it. Thus, the functions of the Kotelnikov–Shannon W-system of the same resolution level with respect to one coordinate affect only one frequency octave. To obtain the required solution inside one octave when the LFA is realized, one can simply modify the WE_1-system, introducing, instead of two modifications $\mu(x)$ and $\gamma_2(t) = (3/4)\left(\sin t - (1/3)\sin^3 t + 2/3\right)$, a system of $2m$ functions, $\mu_s(x)$ and $v_s(x)$, $s = 1, \ldots, m$, whose Fourier transform is concentrated inside the part number s of m parts of ranges $[-\pi, \pi]$ and $[-2\pi, -\pi] \cup [\pi, 2\pi]$, respectively. One can use the initial WE_1 system, combine Fourier coefficients with respect to this system in blocks, and realize the DFT in each block.

Actually, the only but significant shortcoming of the Kotelnikov–Shannon W-systems is a slow decrease of functions $\mu(x)$ and $v(x)$, which belong to $L_2(R)$ but not to $L_1(R)$.

Now, consider the system WE_2 obtained from (4.63)–(4.65) if $S(t) = t$.

Then

$$\widetilde{\mu}(t) = \begin{cases} 1, & 0 \leqslant |t| \leqslant \dfrac{2\pi}{3}; \\ \sin\left(\pi - \dfrac{3}{4}|t|\right), & \dfrac{2\pi}{3} < |t| \leqslant \dfrac{4\pi}{3}; \\ 0, & |t| > \dfrac{4\pi}{3}. \end{cases} \tag{4.72}$$

$$\widetilde{v}(t) = \begin{cases} \sin\left(\dfrac{3}{4}|t| - \dfrac{\pi}{2}\right) e^{it/2}, & \dfrac{2\pi}{3} \leqslant |t| \leqslant \dfrac{4\pi}{3}; \\ \sin\left(\pi - \dfrac{3}{8}|t|\right) e^{it/2}; & \dfrac{4\pi}{3} \leqslant |t| \leqslant \dfrac{8\pi}{3}; \\ 0, & |t| \leqslant \dfrac{2\pi}{3} \quad \text{or} \quad |t| \geqslant \dfrac{8\pi}{3}. \end{cases} \tag{4.73}$$

Consequently,

$$\mu(x) = \frac{\cos 2\pi/3(2x-1)}{\pi\,(1/4+x)\,(7/4-x)} + \frac{\sin(2\pi x/3 - \pi/3)\,(1/2x + 13/18)}{2\pi\,(x-1/2)\,(x+1/4)\,(7/4-x)}, \tag{4.74}$$

$$v(x) = \frac{1}{2\pi}\left(\frac{3/4\cos(8\pi x/3 - 4\pi/3)}{9/64 - (x-1/2)^2} - \right.$$

$$\left. - \frac{9}{16}\frac{(x-1/2)\sin(4\pi x/3 - 2\pi/3)}{\left((x-1/2)^2 - 9/16\right)\left(9/64 - (x-1/2)^2\right)}\right) -$$

$$- \frac{3}{2\pi}\frac{\cos(2\pi x/3 - \pi/3)}{\left((x-1/2)^2 - 9/16\right)}. \tag{4.75}$$

The W-system WE_2 consists of functions $v(2^n x - k)$ and $\mu(x - k)$, where $n = 0, 1, 2, \ldots$ $\ldots, k$ is integer. Functions of this system decrease with the rate as $|x| \to \infty$, but the Fourier transform of functions $\mu(x)$ and $v(x)$ is two times wider as compared to that of

WE_1, and only two levels of the W function affect each octave of the frequency domain. If we suppose

$$S(t) = \begin{cases} \alpha t & \text{if} \quad |t| \leqslant 1/\alpha < 1,\ \alpha > 1, \\ \operatorname{sign} t & \text{if} \quad 1/\alpha < |t| \leqslant 1, \end{cases} \tag{4.76}$$

then, due to (4.69)–(4.65), we obtain the W-system denoted by $WE_{1-1/\alpha}$. When $\alpha = 1$, it is the WE_2 system; when $\alpha = \infty$, the WE_1system. If $1 < \alpha < \infty$, then it is the W-system of functions decreasing with the rate $|x|^{-2}$ as $|x| \to \infty$, and the degree of overlapping for adjacent levels with respect to frequency will be less than that of WE_2. These WE systems can also be called rectangular, trapezoidal, and triangular ones, respectively, due to the forms of the Fourier transform for the function $v(x)$.

4.9. WE-systems with Arbitrary Number of Continuous Derivatives

Now let us consider the method for constructing WE-systems (elementary wavelet systems) with an arbitrary number of continuous derivatives of functions $\widetilde{\mu}(t)$ and $\widetilde{\nu}(t)$, and, correspondingly, functions $\mu(x)$ and $\nu(x)$ decreasing with the rate $|x|^{-m}$ as $|x| \to \infty$ for any finite m. We can achieve this in the following way. For a system $\mu(x-k)$, $\nu\left(2^n x - k\right)$, $n = 0, 1, 2, \ldots$, where k is integer, to be orthogonal, identities (4.66) must be fulfilled. Meyer obtains formula (4.66) from the identity $\sin^2(x) + \sin^2(\pi/2 - x) \equiv 1$. Instead of it, we will use the identity $\left(\sqrt{x}\right)^2 + \left(\sqrt{x-1}\right)^2 \equiv 1$; i.e., instead of $\sin y$, we will use $\sqrt{y}$.Let $r \geqslant 1$ be an integer and

$$\gamma_r(t) = \frac{\left(\int\limits_0^t \cos^{2r-1} \tau d\tau + \alpha_r\right)}{2\alpha_r}, \tag{4.77}$$

where $\alpha_r = \int\limits_0^{\pi/2} \cos^{2r-1} t\, dt$.

The function $\gamma_r(t)$ is a trigonometric polynomial with $\gamma_r(t) \geqslant 0$ for all real t. Hence, there exists a trigonometric polynomial $\delta_r(t)$ (polynomial with respect to $\exp(it)$), generally, complex-valued and such that

$$\gamma_r(t) = \delta_r(t)\overline{\delta}_r(t) = |\delta_r(t)|^2, \tag{4.78}$$

where $\delta_r(\pi/2) = 1$.

Assume that, for $t \geqslant 0$,

$$\varphi_r(t) = \begin{cases} 1, & 0 \leqslant t \leqslant \dfrac{2\pi}{3}, \\ \delta_r\left(-\dfrac{3}{2}t + \dfrac{3}{2}\pi\right), & \dfrac{2\pi}{3} \leqslant t \leqslant \dfrac{4\pi}{3}, \\ 0, & \dfrac{4\pi}{3} < t, \end{cases} \tag{4.79}$$

and, for $t < 0$,

$$\varphi_r(t) = \overline{\varphi}_r(-t), \tag{4.80}$$

(the bar denotes complex conjugation).

Then, for $t > 0$,

$$\psi_r(t) = \begin{cases} 0, & \text{if} \quad 0 \leqslant t \leqslant 2\pi/3 \text{ and } 8\pi/3 \leqslant t, \\ \delta_r\left(3t/2 - 3\pi/2\right), & \text{if} \quad 2\pi/3 \leqslant t \leqslant 4\pi/3, \\ \delta_r\left(-3t/4 + 3\pi/2\right), & \text{if} \quad 4\pi/3 \leqslant t \leqslant 8\pi/3, \end{cases} \tag{4.81}$$

for $t<0$, $\varphi_r(t)=\overline{\varphi_r(t)}$, and, finally,

$$\mu_r(x)=\frac{1}{2\pi}\int\limits_{-\infty}^{\infty}e^{-itx}\varphi_r(t)dt, \tag{4.82}$$

$$v_r(x)=\frac{1}{2\pi}\int\limits_{-\infty}^{\infty}e^{-it\left(x+\frac{1}{2}\right)}\psi_r(t)dt. \tag{4.83}$$

Then, the system $\mu_r(x-k)$, $v_r\left(2^nx-k\right)$, $n=0,1,2,\ldots$, where k is integer, is an orthogonal W-system.

The conditions $\varphi_r(t)=\overline{\varphi_r(-t)}$ and $\psi_r(t)=\overline{\psi_r(-t)}$ indicate that the real parts of functions $\varphi_r(t)$ and $\psi_r(t)$ are even functions of t and the imaginary parts are odd ones. Therefore, functions $\mu_r(x)$ and $v_r(x)$ are real-valued (when $x$is real).

Formulas (4.82) and (4.83) can be rewritten in the form

$$\mu_r(x)=\frac{1}{\pi}\int\limits_{0}^{\infty}\left(\cos tx\Re\varphi_r(t)+\sin tx\Im\varphi_r(t)\right)dt, \tag{4.84}$$

$$v_r(x)=\frac{1}{\pi}\int\limits_{0}^{\infty}\left(\cos tx\Re\psi_r(t)+\sin tx\Im\psi_r(t)\right)dt. \tag{4.85}$$

The functions $\varphi_r(t)$ and $\psi_r(t)$, by their definition, belong to C^r; i.e., they have r continuous derivatives. Here, the derivative of order $(r+1)$ has discontinuities of the first kind only. Therefore, the functions $\mu_r(x)$ and $v_r(x)$ decrease with the rate $|x|^{-r-1}$ as $|x|\to\infty$. In view of the fact that the functions $\varphi_r(t)$ and $\psi_r(t)$ are piecewise trigonometric polynomials (trigonometric splines), their Fourier transforms are

$$\begin{aligned}\delta_2(t)=1/\left[\left(\sqrt{3}-1\right)4\right]\Big[\left(-\sin t+2-\sqrt{3}\right)(\cos 2t-2\sin t-1)-\\ -\cos t\,(\sin 2t+2\cos t)+i\,(\cos t\,(\cos 2t-2\sin t-1)+\\ +(\sin 2t+2\cos t)\left(-\sin t+2-\sqrt{3}\right)\Big)\Big],\end{aligned}$$

and $v_r(x)$ are elementary functions. More exactly, they are finite sums of expressions

$$\cos\alpha_k x/(x-x_k),\quad \sin\alpha_k x/(x-x_k).$$

Let denote this system by WE_{r+1}, $r\geqslant 1$. When $r=1$, it coincides with the WE_2-system constructed above. Consider in detail the construction of the WE_3-system for $r=2$.

In this case,

$$\gamma_2(t)=(3/4)\left(\sin t-(1/3)\sin^3 t+2/3\right). \tag{4.86}$$

For the function $\delta_2(t)$, we obtain the expression

$$\begin{aligned}\delta_2(t)=1/\left[\left(\sqrt{3}-1\right)4\right]\Big[\left(-\sin t+2-\sqrt{3}\right)(\cos 2t-2\sin t-1)-\\ -\cos t\,(\sin 2t+2\cos t)+i\,(\cos t\,(\cos 2t-2\sin t-1)+\\ +(\sin 2t+2\cos t)\left(-\sin t+2-\sqrt{3}\right)\Big)\Big].\end{aligned} \tag{4.87}$$

Hence,

$$\begin{aligned}\Re\delta_2(t) &= \frac{\left(\sqrt{3}-2+\left(2\sqrt{3}-3\right)\sin t-\sqrt{3}\cos 2t-\sin 3t\right)}{\left(\sqrt{3}-1\right)4},\\ \Im\delta_2(t) &= \frac{\left(\left(3-2\sqrt{3}\right)\cos t-\sqrt{3}\sin 2t+\cos 3t\right)}{4\left(\sqrt{3}-1\right)}.\end{aligned} \tag{4.88}$$

Then, the function $\mu_2(x)$ is

$$\begin{aligned}\mu_2(x) = -\frac{\sin(2\pi/3)\,x}{\pi x} &+ \frac{\left(\sqrt{3}-2\right)}{\pi\left(\sqrt{3}-1\right)4}\,\frac{\sin\left(\frac{4\pi}{3}\right)x-\sin\left(\frac{2\pi}{3}\right)x}{x}-\\ &- \frac{1}{4\pi\left(\sqrt{3}-1\right)}\,\frac{\sin\left(\frac{4\pi}{3}\left(x+\frac{9}{2}\right)\right)-\sin\left(\frac{2\pi}{3}\left(x+\frac{9}{2}\right)\right)}{x+3/2}+\\ &+ \frac{\sqrt{3}}{4\pi\left(\sqrt{3}-1\right)}\,\frac{\sin\left(\frac{4\pi}{3}(x+3)\right)-\sin\left(\frac{2\pi}{3}(x+3)\right)}{x+3}-\\ &- \frac{\left(2\sqrt{3}-3\right)}{4\pi\left(\sqrt{3}-1\right)}\,\frac{\sin\left(\frac{4\pi}{3}\left(x+\frac{3}{2}\right)\right)-\sin\left(\frac{2\pi}{3}\left(x+\frac{3}{2}\right)\right)}{x+3/2}.\end{aligned} \tag{4.89}$$

Analogously, for the function $v_2(x)$, we obtain

$$\begin{aligned}v_2(x) = \frac{1}{4\pi\left(\sqrt{3}-1\right)}\Bigg[&\frac{\sin\left(\frac{4\pi}{3}(x-4)\right)-\sin\left(\frac{2\pi}{3}(x-4)\right)}{x-4}+\\ &+\sqrt{3}\,\frac{\sin\left(\frac{4\pi}{3}\left(x-\frac{5}{2}\right)\right)-\sin\left(\frac{2\pi}{3}\left(x-\frac{5}{2}\right)\right)}{x-\frac{5}{2}}-\\ &-\left(2\sqrt{3}-3\right)\frac{\sin\left(\frac{4\pi}{3}(x-1)\right)-\sin\left(\frac{2\pi}{3}(x-1)\right)}{x-1}+\\ &+\left(\sqrt{3}-2\right)\frac{\sin\left(\frac{4\pi}{3}\left(x+\frac{1}{2}\right)\right)-\sin\left(\frac{2\pi}{3}\left(x+\frac{1}{2}\right)\right)}{x+\frac{1}{2}}-\\ &-\frac{\sin\left(\frac{8\pi}{3}\left(x+\frac{11}{4}\right)\right)-\sin\left(\frac{4\pi}{3}\left(x+\frac{11}{4}\right)\right)}{x+\frac{11}{4}}+\\ &+\sqrt{3}\,\frac{\sin\left(\frac{8\pi}{3}(x+2)\right)-\sin\left(\frac{4\pi}{3}(x+2)\right)}{x+2}-\\ &-\left(2\sqrt{3}-3\right)\frac{\sin\left(\frac{8\pi}{3}\left(x+\frac{5}{4}\right)\right)-\sin\left(\frac{4\pi}{3}\left(x+\frac{5}{4}\right)\right)}{x+\frac{5}{4}}+\\ &+\left(\sqrt{3}-2\right)\frac{\sin\left(\frac{8\pi}{3}\left(x+\frac{1}{2}\right)\right)-\sin\left(\frac{4\pi}{3}\left(x+\frac{1}{2}\right)\right)}{x+\frac{1}{2}}\Bigg].\end{aligned} \tag{4.90}$$

4.10. Wavelets Systems based on Atomic Functions

Consider W-systems constructed by means of the atomic functions.

Atomic functions are compactly supported (i.e., they are equal to zero outside a finite interval) solutions of linear functional-differential equations with constant coefficients and linear transformations of their argument

$$Ly(x) = \lambda \sum_{k=1}^{M} c_k y\,(ax - b_k), \quad |a| > 1, \tag{4.91}$$

where L is a linear differential operator with constant coefficients [8, 10, 15]. Atomic functions are well localized because of their compactness. Their Fourier transforms decrease on the real axis faster than any power function. Moreover, these Fourier transforms are infinite products of periodic functions. This means that atomic functions themselves generate W-systems. We will call them WA-systems according to the scheme described at the beginning. These orthogonal WA-systems consist of exponentially localized infinitely differentiable functions. Naturally, functions of these systems are not the compressed translations of one or two functions. This means that the form of generative functions varies from one level to another (when multiresolution analysis is realized). However, the demand of constancy for this form is not necessary in many cases. Moreover, an asymptotic similarity nevertheless is a form of generative functions. For levels with large numbers, it is close to the density function of the normal distribution, i.e., to $\exp\left(-x^2/2\right)$.

Consider a concrete example. Let UP_n be the spaces of linear combinations of translations of functions $\mathrm{up}\,(x)$

$$\sum_k c_k \,\mathrm{up}\,\left(x - k \cdot 2^{-n}\right). \tag{4.92}$$

A space UP_n has a basis consisting of translations of the compactly supported atomic function $Fup_n(x)$ [12] equal to zero outside the interval $\left[-(n+2)2^{-n-1}, (n+2)2^{-n-1}\right]$ with length $(n+2)2^{-n} \to 0$ as $n \to \infty$. Obviously, the condition $UP_n \subset UP_{n+1}$ is fulfilled owing to (4.53). As was shown in [11, 13–15], the spaces UP_n possess optimal properties from the viewpoint of approximation theory. One of the main properties of functions from the class C^r (r times infinitely differentiable) is their capability of being approximated by elements of spaces UP_n with the best possible rate. In other words, spaces UP_n are extremal or asymptotically extremal from the viewpoint of the Kolmogorov widths [26] and also for all r. Such a combination of an approximate universality and localization (the presence of a basis of functions with small supports) is unique. The approximation technique used earlier was either approximately universal but not localized (classical technique of polynomials and rational functions), or localized but not approximately universal, as the novel techniques of splines, which possess the approximation saturation. To approximate smoother functions, one can use smoother splines. In a certain sense, atomic functions are considered as infinitely smooth splines of the class C^∞. They are used in different domains of mathematics and its applications [9–15, 20–23], particularly in signal and image processing including the synthesis of weighting window functions for the Fourier analysis [21–22].

All the aforesaid allows us to conclude that the use of the WA-system is very promising, especially in solving boundary value problems for partial differential equations.

In the first study of the function $\mathrm{up}\,(x)$ [7], published in 1971, long before the appearance of ondelettes or wavelets, it was proposed to use $\mathrm{up}\,(2^n x - k \cdot 2^m)$ as trial

functions, i.e., to associate a function f with a set of coefficients $c_{n,m,k}$ that are the values of f on the trial function

$$\text{up}\,(2^n x - k\cdot 2^m)\,, \quad c_{n,m,k} = f\,(\,\text{up}\,(2^n x - k\cdot 2^m)) \tag{4.93}$$

or

$$c_{n,m,k} = \int\limits_{-\infty}^{\infty} f(x)\,\text{up}\,(2^n x - k\cdot 2^m)\,dx. \tag{4.94}$$

Therefore, a nonorthogonal W-system was considered. It should be noted that, in the modern theory of W-systems, nonorthogonal W-systems are widely used.

The functions $Fup_n(x)$ have the form

$$Fup_n(x) = \frac{1}{2\pi}\int\limits_{-\infty}^{\infty} e^{itx}F_n(t)dt, \tag{4.95}$$

where

$$F_n(t) = \left(\frac{\sin t\cdot 2^{-n-1}}{t\cdot 2^{-n-1}}\right)^{n+1}\prod_{k=n+2}^{\infty}\frac{\sin t\cdot 2^{-k}}{t\cdot 2^{-k}}. \tag{4.96}$$

From (4.96), it follows that the function $Fup_n(x)$ is a convolution of the B spline $B_n\,(2^n x - n - 1)$ with the function $\text{up}_n\left(2^{n+1}x\right)2^{n+1}$ (to within normalization).

$$Fup_n(x) = \alpha_n B_n\,(2^n x - n - 1)\otimes\,\text{up}_n\left(2^{n+1}x\right)2^{n+1}.$$

If we consider a sequence of spaces $S_{n,n}$ of the natural splines of degree n on a mesh with the width 2^{-n}, i.e., a sequence of spline spaces with their degree increasing to obtain an approximate universality, then the condition $L_n \subset L_{n+1}$, necessary for constructing a W-system, is not fulfilled $S_{n,n} \not\subset S_{n+1,n+1}$.

If we act on the space $S_{n,n}$ with a linear operator $C\,(\,\text{up}\,(n))$ of a convolution with the function $2^{n+1}\cdot\left(\text{up}_n\left(2^{n+1}x\right)\right)$, then we will obtain a sequence of spaces $UP_n = C\,(\,\text{up}\,(n))\,S_{n,n}$, where $UP_n \subset UP_{n+1}$. Obviously, the $C\,(\,\text{up}\,(n)) \to I$ as $n\to\infty$ (where I*is* an identity operator), because $2^{n+1}\left(\text{up}_n\left(2^{n+1}x\right)\right)$ tends to $\delta(x)$ (δ is the Dirac delta-function). The space UP_n for even n is generated by translations of the function $Fup_n(x)$ of the form $Fup_n\,(x - 2^{-n}k)$, and, for odd n, by $Fup_n\left(x - 2^{-n}k + 2^{-n-1}\right)$. To construct spaces W_n such that $UP_{n+1} = UP_n \oplus W_{n+1}$ and $W_{n+1}\perp UP_n$, i.e., orthogonal projectors to UP_n in UP_{n+1}, one must find coefficients of a finite linear combination of translations of the function $Fup_{n+1}(x)$ of the form $\sum\limits_l c_k Fup_n\left(x - l\cdot 2^{-n-1} + 2^{-n-2}\right)$ that is orthogonal to all translations of the function $Fup_n(x)$ of the form $Fup_n\,(x - 2^{-n}k)$ for even n, and of the form $\sum\limits_l c_k Fup_n\left(x - l\cdot 2^{-n-1}\right)$, which is orthogonal to all $Fup_n\left(x - 2^{-n}k + 2^{-n-1}\right)$, for odd n.

Since the translation width of $Fup_{n+1}(x)$ in the space UP_{n+1} is smaller by a factor of two than that of the function $Fup_n(x)$ in UP_n, and the support of $Fup_{n+1}(x)$ is approximately shorter by a factor of two than that of$Fup_n(x)$, this problem is easily solvable.

Let $\psi_{n+1}(x)$ be a nonzero function with minimal support from the space UP_{n+1} obtained as a result of solving a finite homogeneous system of linear equations. Then, the translations $\psi_{n+1}\,(x - k\cdot 2^{-n})$ form a basis for the required space W_{n+1}, $n = 0, 1, \ldots$, and W_0 coincides with UP_0 being generated by the functions $\text{up}\,(x - k)$.

To obtain an orthogonal W-system, it remains to construct an orthogonal basis $\mu(x - k)$ formed by translations of the function $\mu(x)$ in W_0 and translations $v\left(x - k \cdot 2^{-n+1}\right)$ of the function $v_n(x)$ in spaces W_n, $n \geqslant 1$ by the methods described above and illustrated by the example of constructing a WS_n-system. Moreover, one can obtain both one-sided and symmetric W-functions. Here, the W-system will be exponentially localized, but, due to the dependence of the characteristic equation on n, the investigation of the rate of exponential decay of $v_n(x)$ as $|x| \to \infty$ demands additional analysis that is beyond the scope of this work.

References

1. *Stromberg, J.O.* A Modified Franklin System and Higher-Order Systems, Conf. on Harmonic Analysis in Honor of A. Zygmund, 1983, vol. 2, pp. 475–494.
2. *Lemarie, P.G.* Ondelettes a localisation exponentielle. J. Math., Pures et Appl., 1987, vol. 67 , pp. 227–236.
3. *Daubechies, I.* Orthogonal Bases of Compactly Supported Wavelets. Comm. Pure Appl. Math., 1988, vol. 41, pp.E909–996.
4. *Meyer, Y.* Ondelettes et operateurs, vols. 1–3, Paris, Hermann, 1990.
5. *Daubechies, I.* Ten Lectures on Wavelets. Philadelphia, SIAM, 1992.
6. *Chui, C.K.* An Introduction to Wavelets. New York, Academic Press, 1992.
7. *Rvachev, V.L. and Rvachev, V.A.*, Ob odnoi finitnoi funktsii (On One Compactly Supported Function). DAN of the USSR, 1971, ser. A, no. 8, pp. 705–707.
8. *Rvachev, V.A.* Predstavleniye eksponentsial'nykh funktsii finitnymi (Representation of Exponential Functions by Compactly Supported Functions). DAN of the USSR, 1973, ser. A, no.9, pp. 821–824.
9. *Rvachev, V.A.* Odna ortonormirovannaya sistema na osnove funktsii $up(x)$ (One Orthonormal System on the Basis of Function up(x). Matematicheskii analiz i teoriya veroyatnosti, Kiev, Inst. Matematiki Akad. Nauk UkrSSR, 1978, pp. 146–150.
10. *Rvachev, V.A.* Atomarnye funktsii (Atomic Functions). Matematicheskii analiz i teoriya veroyatnosti, Kiev: Inst. Matematiki AN UkrSSR, 1978, pp. 143–146.
11. *Rvachev, V.A.* O priblizhenii s pomoshch'u funktsii $up(x)$ (On Approximation with the Use of the Function up(x)). DAN of the SSSR, 1977, vol. 233, no. 2, pp. 295–296.
12. *Rvachev, V.A.* Primeneniye funktsii $up(x)$ v variatsionno-raznostnykh metodakh (The Use of Function up(x) in Variational Difference Methods). Preprint of V.Ts. Sib. Otd. Akad. Nauk SSSR, Novosibirsk, 1975, no. 16.
13. *Rvachev, V.L. and Rvachev, V.A.* Neklassicheskiye metody teorii approksimatsii v krayevykh zadachakh (Nonclassical Methods of Approximation Theory in Boundary Value Problems). Kiev, Naukova Dumka, 1979.
14. *Rvachev, V.A.* Atomarnye funktsii i teoriya priblizhenii (Atomic Functions and Approximation Theory). Trudy Matem. Inst. Akad. Nauk SSSR im. V.A. Steklova, 1987, vol. 180, pp. 186–187.
15. *Rvachev, V.A.* Finitnye resheniya funktsional'no-differentsial'nykh uravnenii i ikh primeneniya (Compactly Supported Solutions of Functional-Differential Equations and Their Applications). Uspekhi Matem. Nauk, 1990, vol. 45, issue 1, pp. 77–103.
16. *Kolodyazhny V.M. and Rvachev, V.A.* On the Construction of Meyer Wavelets Using $up(x)$ Function. Dokl. Akad. Nauk Ukrainy, 1993, no. 10, pp. 18–24.
17. *Kolodyazhny, V.M. and Rvachev, V.A.* Construction of Wavelet System. Dokl. Akad. Nauk Ukrainy, 1995, no. 2, pp. 77–81.
18. *Kravchenko, V.F., Rvachev, V.A., and Pustovoit, V.I.* Algoritm postroeniya "wavelet" sistem dlya obrabotki signalov (An Algorithm of Construction of Wavelet System for Signal Processing). DAN RAN, 1996, vol. 346, no. 1, pp. 31–32.

19. *Rvachev*, *V.A.* Pro pobudovu lokalizovannykh ortogonal'nykh sistem tipu "wavelet" (On the Construction of Localized Orthogonal Systems of the Wavelet Kind). Abstracts of papers, Mezhdunar. Konf. Teoriya Priblizhenii v Zadachakh Vychisl. Matem., Dnepropetrovsk, 1993, p.159.
20. *Kravchenko*, *V.F. and Rvachev*, *V.A.* Primeneniye atomarnykh funktsii dlya sinteza KIKh fil'trov (The Use of Atomic Functions for the Synthesis of Short Impulse Responces of Filters). Radiotekhnika, 1995, no.10 (Elektromagn. Volny, no. 3), pp. 80–82.
21. *Gulyaev*, *Yu.V.*, *Kravchenko*, *V.F.*, *and Rvachev*, *V.A.* Doklady Mathematics, 1995, vol. 51, no. 3, pp. 456–458.
22. *Kravchenko*, *V.F.*, *Rvachev*, *V.A.*, *and Rvachev*, *V.L.* DAN of the SSSR, 1989, vol. 306, no. 1, p. 78.
23. *Kravchenko*, *V.F.*, *Rvachev*, *V.A.*, *and Rvachev*, *V.L.* Mathematical Methods for Signal Processing Based on Atomic Functions. Journal of Communications Technology and Electronics, 1995, vol. 40(12), pp. 118–137
24. *Crandall*, *R.E.* Projects in Scientific Computation. New York: Springer Verlag, 1994.
25. *Harris*, *F.J.* On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. Proc. of the IEEE, 1978, vol. 66, no.1, pp. 51–83.
26. *Tikhomirov*, *V.M.* Poperechniki mnozhestv v funktsional'nykh prostranstvakh i teoriya nailuchshikh priblizhenii (Sets Widths in Functional Spaces and the Theory of Best Approximation). Uspekhi Matematicheskikh Nauk, 1960, vol. 15, issue 3, pp. 81–120.
27. *Haar*, *A.* Theorie der orthogonalen Funktionen–Systeme. Math. Annalen, 1910, vol. 69, pp. 331–374.
28. *Franklin*, *P.A.* Set of Continuous Orthogonal Functions. Math. Annalen, 1928, vol. 100, pp. 522–529.

Chapter 5

MODELS OF IMAGE AND NOISE

5.1. Noise

Real-world images are usually affected by random fluctuations in intensity, color, texture, object boundary, or shape. There is a lot of different and complex reasons for these fluctuations, often due to factors such as non-uniform lighting, random fluctuations in object's surface orientation and texture, sensor's limitations, etc. The processing of such images can be treated as a problem of statistical inference, which requires the definition of a statistical model corresponding to the image and noise pixels. Although simple image models can be obtained from image statistics such as the mean, variance, histogram, and correlation function, the most general approach is to use random field models. Combined with various frameworks for statistical inference such as maximum likelihood (ML) and Bayesian estimation, random field models are used in many applications of statistical image processing. These include image restoration, enhancement, classification, segmentation, compression, and synthesis [1–5].

The most general model of image-noise representation consists in definition of the random process (field) representing the multidimensional signal and the random process representing the corrupted multidimensional image along with the joint density that models the corruption mechanism [1, 2, 4–6].

The simplest additive noise corruption in an image is assumed in the form of *zero-mean additive white noise model*.

Images are also generally thought of as relatively broadband signals. Important visual information may reside at mid-to-high spatial frequencies, since visually significant image details, such as edges, lines, and textures typically contain higher frequencies. Nevertheless, the higher image frequencies are visually significant.

In many cases, the classical approach to noise suppression using linear filtering algorithms is equivalent to image enhancement by low-pass filtering. For a given filter type, different quality of smoothing can be attained by adjusting the bandwidth of a filter. The use of a narrower bandwidth low-pass filter can reject high-frequency content of an image, carrying significant information but, in the same time, gives the possibility to decrease the additive noise corruption.

Image sharpening refers to any enhancement technique that highlights edges and fine details in an image. In principle, image sharpening consists in adding to the original image a signal proportional to a high-pass filtered version of the original image.

When a multidimensional signal (two-, 2D, or three-dimensional, 3D) is observed, it is usually corrupted by noise of different nature. The noise may be additive, multiplicative, or of a more general nature, and some assumptions should be made on its statistical properties. The goal is to estimate the original uncorrupted multidimensional signal with a good accuracy and good preservation quality for edges and fine details [1, 5, 7–9].

Digital image enhancement and analysis have played and will play an important role in scientific, industrial, and military applications. In addition to these applications,

image enhancement and analysis are increasingly used in consumer electronics and other applications; for instance, by the Web users, who rely not only on the built-in image processing protocols, such as JPEG2000, MPEG format, etc., and interpolation.

Image enhancement refers to processes seeking to improve the visual appearance of an image. As an example, image enhancement might be used to emphasize the edges and fine details within the image. Such edge-enhanced image would be more visually pleasing to the naked eye or perhaps could serve as an input to a machine that would detect the edges and make measurements of shape and size of the edges of the objects.

The leading role mainly belongs to the following factors: the mechanism of generating multidimensional signals, the nature of corruption, and the accuracy of solution according to the made assumptions.

5.2. Additive Noise

Optimal methods of linear filtering theory can be useful only when the corruption can be represented as a Gaussian process and the criterion of accuracy is the mean-square error (MSE). This assumption is not true in most applications, for example in digital systems, where the errors are often caused by bit changes and the distribution is far from Gaussian. On other hand, for the visual quality, the MSE is not a realistic criterion [1, 3, 4, 9].

In image processing the assumption of additive white noise rarely holds. The intensity of an image formed by an image acquisition system is usually multiplicative with respect to illumination and the reflectivity of the observed surface.

Some of the noises are naturally occurring, e.g., Gaussian noise; some are sensor induced, e.g., photon counting noise and speckle one; and some result from various processing, e.g., quantization and transmission.

Noise is usually defined as an unwanted component of the image. Gaussian noise is a part of almost any signal, for example, the white noise on a weak television station is well modeled as Gaussian. Since image sensors have to count photons — especially in low light situations — and the number of photons counted is a random quantity, images often have photon counting noise usually modeled as a Poisson process [1–4].

Photographic grain noise is a characteristic of photographic films. It limits the effective magnification that one can obtain from a photograph. This noise in photographic films is sometimes modeled as Gaussian and sometimes as Poisson.

Gaussian noise is usually considered to be an additive component.

The additive model is most appropriate when the noise in the model is independent of an image. There are many applications of the additive model: thermal noise, photographic noise, and quantization noise, etc.

Also known are quantization noise and speckle in coherent light situations.

5.3. Speckle Image Noise

In this work, we touch on other important applications of image processing, such as ultrasound (US) [1, 10, 11] and SAR imaging [12, 13]. US visualization is one of the most efficient methods of medical diagnostics, also widely used in other fields. On other hand, SAR imaging systems are used in applications of remote sensing. A significant limitation on the quality of these systems is the effect of multiplicative (speckle) noise.

Speckle is one of the more complex image noise models. It is signal dependent, usually non-Gaussian, and spatially dependent. The physical nature of this noise is in some variations in phase and amplitude that can effect in such a way: some of these

variations in phase add constructively, resulting in strong intensities, and others add deconstructively, resulting in low intensities. These variations are called speckle.

For understanding the speckle corruptions, it is common to use the point-spread function (PSF). Depending on sensor type and system (US, SAR, laser, etc.), there can be three cases.

The PSF is so narrow that the individual variations in surface roughness can be resolved, which is uncommon in most applications.

The PSF is broad in comparison with typical size of the surface roughness, but, in the same time, small in comparison with the details in the image. This is a common case and the speckle noise is exponentially distributed and uncorrelated on the scale of the details in the image. This gives a multiplicative noise model.

The PSF is broad compared to both the characteristic object's size and for surface roughness. The speckle noise is correlated and its distribution can be determined by the PSF.

Usually, it is assumed that the surface is very rough on the scale of the wavelengths, which means that each microscopic reflector has a random height and random orientation with respect to the incoming polarization field. The random changes are reflected in the amplitude, phase, and polarization of the signal, which can be approximated as independent from each other and from changes at any other point. Because the system cannot resolve variations in roughness, it results in speckle noise. All coherent systems including lasers, SAR, and US sensing systems are subjected to an effect of multiplicative (speckle) noise. This noise arises in sensing by such coherent sensors of objects having nonhomogeneities. In this case, the filtration of speckle noise is an obligatory pre-processing procedure, making it possible to improve image characteristics and, as a consequence, the quality of recognition and diagnostics in remote sensing, medical and other applications [10, 11].

Human perception is highly sensitive to edges and fine details of an image, so the visual quality of an image can be enormously degraded if the high frequencies are attenuated or completely removed. In contrast, enhancing the high-frequency components of an image leads to an improvement in the visual quality.

5.4. Impulsive Image Noise

In many applications of image processing, the noise is far more complex and the mixture of noise and signal is very complicated. Linear methods largely fail in analysis of impulsive noise. It is assumed that a noise process is impulsive if many of the signal values do not change at all or change slightly and some signal values change dramatically, i.e., the change is clearly visible [1, 2, 4, 10].

In practice, the same number of bits will be used to represent the noisy and the noise-free signal, usually 8 bits or 256 levels 0, 1, ... , 255. There are many models for impulsive noise. For example, the impulses may have different amplitude values. Common for the models of impulsive noise in images is the appearance of noise as black and/or white (for color images, color) spots in the images, i.e., the noisy pixels have either a very small or very large values. This type of noise is often called salt-and-pepper noise because one could create it by sprinkling salt-and-pepper on an image. Formally, pure salt-and-pepper noise is very easy to remove from images because maximum values rarely occur in actual images and, thus, just checking whether the pixel has a maximum or minimum value reveals if it is corrupted or not. As a rule, the realistic impulsive noise is modeled as bit errors in signal values. Typical sources for this kind of noise are channel errors in communication or storage. For example, such noise arises in transmission of images over noisy digital links. Let each a pixel be quantized to B bits

in a usual way. Assume that the channel is binary symmetric with a given crossover probability. Then it is easy to show that the contribution to the MSE from the most significant bit is approximately 3 times that of all the other bits. This noise is an example of (very) heavy-tailed noise [4].

Impulses are also referred to as *outliers*. Another approach presented below can also be effective and is connected with robust statistics based on rank (R) and generalized maximum likelihood statistics (M) [14, 15]. Details of this approach will be explained below in this work.

Several types of impulsive models usually can be used. Some of them need the detail *a priori* information on the degradation process in each channel for multichannel (or color) multidimensional image. In our opinion, complex models that need several parameters that must be determined *a priori* or during the processing stage have low tolerance and therefore such models can produce confusion in the interpretation of filtering results [7–9, 16, 17].

Below, we use a simple and, in the same time, the most severe model of impulsive noise from the viewpoint of image degradation. This model needs only prior information about the probability p of random spikes appearance, which are independent in each channel. Additionally, the amplitude of impulsive noise is modeled as a uniformly distributed random value within the interval of given values (0–255) for each channel in the case of color (multichannel) images.

5.5. Mathematical Solutions Applied in Noise Models

There exist different models of noise that are dependent on physical noise nature or different representation of multidimensional signals [1–5, 7, 8, 16, 17]. According to the discussion presented above, the simplest model is the model of additive Gaussian noise degradation

$$u(i,j) = Y(i,j) + n(i,j), \tag{5.1}$$

where $Y(i,j)$ is an original image, $u(i,j)$ is a degraded image and $n(i,j)$ is a Gaussian additive noise.

Also, we use the following model for noise influence in the case of impulse noise degradation [18, 19]:

$$\begin{gathered} Y(i,j) = n_i\left(Y(i,j)\right), \\ n_i\left(Y(i,j)\right) = \begin{cases} \text{random values with probability } P, \\ Y(i,j) \text{ another case,} \end{cases} \end{gathered} \tag{5.2}$$

where $Y(i,j)$ is an original image, $u(i,j)$ is a degraded image and $n_i\left(Y(i,j)\right)$ is the function presented above.

There is another, more complicated, model that uses information about corruption in each channel [20, 21]:

$$\overrightarrow{Y} = \begin{cases} (n_1, Y_G, Y_B) & \text{with probability } pp_1, \\ (Y_R, n_2, Y_B) & \text{with probability } pp_2, \\ (Y_R, Y_G, n_3) & \text{with probability } pp_3, \\ (n_1, n_2, n_3) & \text{with probability } p(1 - \sum_{i=1}^{3} p_i), \end{cases} \tag{5.2a}$$

where n_i are independent random values for each channel, uniformly distributed in the interval (0, 255) for every pixel or voxel. The main drawback of this model is that *a priory* information about values p and p_i is needed to implement the filtering algorithm.

In the case of multiplicative noise degradation, model (5.1), (5.2) can be represented in the form [22, 23]:

$$Y(i,j) = n_i\left(\varepsilon_m(i,j)\cdot Y(i,j)\right), \tag{5.3}$$

where $n_m(i,j)$ denotes multiplicative noise.

Equations (5.1)–(5.3) present the basic models in noise degradations.

For mulichannel images, it is necessary to apply equation (5.2) for each channel.

In the case of multidimensional image representation, model (5.2)–(5.3) is changed and, for 3D discrete image, can be rewritten as follows [17]:

$$Y(i,j,k) = n_i\left(Y(i,j,k)\varepsilon_m(i,j,k)\right), \tag{5.4}$$

where $n_i\left(Y(i,j,k)\right)$ is the functional

$$n_i\left(Y(i,j,k)\right) = \begin{cases} \text{noise } n_i \text{ with probability } p, \\ Y(i,j,k), \text{otherwise}, \end{cases}$$

$Y_{speckle}(i,j,k)$ is a noisy observation (i.e., the recorded image) of the 3-D function $Y(i,j,k)$ (i.e., the noise-free image to be recovered), and $\varepsilon_m(i,j,k)$ is the corrupting multiplicative (speckle) noise component.

5.6. Objective and Subjective Criteria

To evaluate different filters and compare their performances against the performance of reference filtering techniques presented in literature, several criteria are used, such as the *peak signal-to-noise ratio* (*PSNR*) and *normalized mean-square error* (*NMSE*) for the evaluation of noise suppression, the *mean absolute error* (*MAE*) for quantization of edges and fine detail preservation, and the *normalized color difference* (*NCD*) for the quantization of the color (multichannel) perceptual error [1, 4, 8, 16, 17, 20]:

$$PSNR = 10\cdot\log\left[\frac{(255)^2}{MSE}\right],\ \text{dB}, \tag{5.5}$$

$$NMSE = \frac{\sum\limits_{i=1}^{M_1}\sum\limits_{j=1}^{M_2}\|y(i,j)-y_0(i,j)\|_{L_2}^2}{\sum\limits_{i=1}^{M_1}\sum\limits_{j=1}^{M_2}\|y_0(i,j)\|_{L_2}^2}, \tag{5.6}$$

$$MAE = \frac{1}{M_1M_2}\sum_{i=1}^{M_1}\sum_{j=1}^{M_2}\|y(i,j)-y_0(i,j)\|_{L_1}, \tag{5.7}$$

where $MSE = \frac{1}{M_1M_2}\sum\limits_{i=1}^{M_1}\sum\limits_{j=1}^{M_2}\|y(i,j)-y_0(i,j)\|_{L_2}^2$ is the *mean-square error*, M_1, M_2 are the image dimensions, $y(i,j)$ is the 3D vector value of the pixel (i,j) in the filtered color image, $y_0(i,j)$ is the corresponding 3D vector value of the pixel in the original uncorrupted image, and $\|\cdot\|_{L_1}$ and $\|\cdot\|_{L_2}$ are the L_1- and L_2-vector norms, respectively;

$$NCD = \frac{\sum\limits_{i=1}^{M_1}\sum\limits_{j=1}^{M_2}\|\Delta E_{Luv}(i,j)\|_{L_2}}{\sum\limits_{i=1}^{M_1}\sum\limits_{j=1}^{M_2}\|E^*_{Luv}(i,j)\|_{L_2}}. \tag{5.8}$$

Here, $\|\Delta E_{Luv}(i,j)\|_{L_2} = \left[(\Delta L^*(i,j))^2 + (\Delta u^*)^2 + (\Delta v^*)^2\right]^{1/2}$ is the norm of color (or multichannel) error; ΔL^*, Δu^*, and Δv^* are the difference in the L^*, u^*, and v^* components, respectively, between the two color vectors that present the filtered image and uncorrupted original one for each pixel (i,j) of an image; and $\|E^*_{Luv}(i,j)\|_{L_2} = \left[(L^*)^2 + (u^*)^2 + (v^*)^2\right]^{1/2}$ is the L_2 norm or magnitude of the uncorrupted original image pixel vector in the $L^*u^*v^*$ space. It has been proved that the *NCD* objective measure expresses well the color distortion (for example, see [8, 9]).

It should be noted that, in the case of 3D imaging when the observation is presented as in equation (5.2), all the criteria should be treated similarly in calculating all the measures (*PSNR*, *NMSE*, *MAE*, and *NCD*) for 3D image, for example, as for MAE criterion:

$$MAE = \frac{1}{M_1 M_2 M_3} \sum_{i=0}^{M_1-1} \sum_{j=0}^{M_2-1} \sum_{k=0}^{M_3-1} \left| Y_{speckle}(i,j,k) - \widehat{Y}(i,j,k) \right|, \tag{5.9}$$

where $Y(i,j,k)$ is the original free noise 3-D image voxel (in the case of color image, 3D vector), $\widehat{Y}(i,j,k)$ is the restored 3-D image voxel (3D vector for color image), and M_1, M_2, M_3 are the sizes of the 3-D image.

What weight is given to each error criterion depends on particular applications where the filter is used. Since it is difficult to define the error criteria for an accurate quantization of image distortion, we also use a subjective measure of the image distortion in form of subjective visual criterion presented by error image — the absolute difference between the original and filtered image. So, subjective visual comparison of the images provides information about the spatial distortion and artifacts introduced by different filters, as well as the noise suppression quality of the algorithm, and present performance of the filter when filtering images are observed by the human visual system.

References

1. *Bovik*, *A*. Handbook of Image and Video Processing, Academic Press, San Diego CA, 2000.
2. *Pitas*, *I*. Digital image processing algorithms and applications. New York: Wiley; 2000.
3. *Gonzalez*, *R.C. and. Woods R.E*. Digital Image Processing. Reading, MA/Englewood Cliffs, NJ: Addison Wesley/Prentice Hall, 1992.
4. *Astola*, *J. and Kuosmanen*, *P*. Fundamentals of Nonlinear Digital Filtering, CRC Press, Boca Raton-New York, 1997.
5. *Barner*, *K.E. and Arce*, *G.R*. Eds. Nonlinear Signal and Image Processing: Theory, Methods, and Applications. Boca Raton, FL: CRC, 2004.
6. *Jan*, *J*. Medical Image Processing, Reconstruction and Restoration: Concepts and Methods, New York Taylor & Francis, CRC Press, 2006.
7. *Lukac*, *R*., *Smolka*, *B*., *Martin*, *K*., *Plataniotis*, *K.N*., *and Venetsanopoulos*, *A.N*. Vector filtering for color imaging. IEEE Signal Processing Magazine, Special Issue on Color Image Processing 2005, vol. 22 no. 1, pp. 74–86.
8. *Trahanias*, *P.E. and Venetsanopoulos*, *A.N*. Directional Processing of Color Images: Theory and Experimental Results. IEEE Transactions on Image Processing, 1996, vol. 5., no. 6.
9. *Plataniotis*, *K.N. and Venetsanopoulos*, *A.N*. Color Image Processing and Applications, Berlin, Springer Verlag, 2000.
10. *Webb*, *A.G*. Introduction to Biomedical Imaging. Hoboken, New Jersey: Wiley–IEEE Press. 2002.

11. *Jan, J.* Medical Image Processing, Reconstruction and Restoration: Concepts and Methods, New York Taylor & Francis, CRC Press, 2006.
12. *Skolnik and Merril, I.* Radar handbook, McGraw Hill, 1990.
13. *Natashon, F.E. and Patrick Reilly, J.* Radar design principles, Scitech Publ., INC, 1999.
14. *Hampel, F.R., Ronchetti, E.M., Rouseew, P.J., and Stahel, W.A.* Robust Statistics: The approach based on influence function, Wiley, New York, 1986.
15. *Huber, P.J.* Robust Statistics, Wiley, New York, 1981.
16. *Ponomaryov, V.I., Gallegos-Funes, F.J., and Rosales-Silva, A.* Real-time color imaging based on RM-filters for impulsive noise reduction. Journal of Imaging Science and Technology, 2005, vol. 49 no. 3. pp. 205–219.
17. *Ponomaryov, V.I.* Real-time 2D-3D filtering using order statistics based algorithms. Survey paper. Journal of Real-Time Image Processing, Springer Edit. 2007, vol. 1, no. 3, pp. 173–194.
18. *Ponomaryov, V.I. and Pogrebnyak, O.B.* Novel robust RM filters for radar image preliminary processing. Journal of Electronic Imaging, October 1999, vol. 08, no. 4, pp. 467–477.
19. *Kravchenko, V.F, Ponomaryov, V.I., Pustovoit, V.I., Nino-De-Rivera, L., and Gallegos Funes, F.* Robust Image Fitration with Fine Detail Preservation in presence of Multiplicative and Impulsive Noise. Doklady Physics, Edit. «Nauka/Inerperiodica». N.Y. 2001. vol. 46, no. 2. pp. 97–102.
20. *Lukac, R., Smolka, B., Plataniotis, K.N., and Venetsanopoulos, A.N.* Selection weighted vector directional filters. Computer Vision and Image Understanding 2004, vol. 94, pp. 140–67.
21. *Lukac, R.* Adaptive vector median filtering. Pattern Recognition Letters 2003, vol. 24, no. 12, pp. 1889–1899.
22. *Volosyuk, V.K., Kravchenko, V.F., and Ponomaryov, V.I.* Mathematical Methods Physical Processes Modeling in Remote Sensing Problems of the Earth. Uspekji sovremennoi electroniki, 2000, no 12, pp. 3–74 (in Russian).
23. *Kravchenko, V.F, Ponomaryov, V.I., Pogrebnyak, A.B., Pustovoit, V.I., and Nino-de-Rivera, L.* Robust Image processing filters preserving small features and rejecting impulse noise. Journal of Communications Technology and Electronics, 2001, vol. 46, no. 4, pp. 450–455.

Chapter 6

TYPES OF STATISTICAL ESTIMATORS

6.1. Maximum Likelihood and M estimators

Huber has proposed the M estimators as the maximum likelihood (ML) function estimations [1–6]. Its definition is given via function ρ, $\{\rho(X) = \ln(F(X))\}$, related to the density probability function $F(X)$ of input data sample set X_i, $i = 1, 2, \ldots, N$, as

$$\widehat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{i=1}^{N} \rho(X_i - \theta). \tag{6.1}$$

The point estimation of parameter θ can be found by calculating the derivative ψ with respect to θ [1, 4, 5]:

$$\sum_{i=1}^{N} \psi(X_i - \theta) = 0, \tag{6.2}$$

where θ is unknown bias of parameter.

Standard technique consists in the usage of the iterative Newton procedure [3, 4] and permits one to write the following equation:

$$\widehat{\theta}^{(q+1)} = \widehat{\theta}^{(q)} + S^{(q)} \frac{\sum \psi \left[\dfrac{X_i - \widehat{\theta}^{(q)}}{S_0} \right]}{\sum \psi' \left[\dfrac{X_i - \widehat{\theta}^{(q)}}{S_0} \right]}, \tag{6.3}$$

where $\widehat{\theta}^{(q)}$ is the M-estimation for parameter θ and S_0 is a scale estimation. Usually, $\widehat{\theta}^{(0)}$ is chosen as the median of primary data and

$$S_0 = MED \left\{ \left| y_i - \widehat{\theta}^{(0)} \right| \right\} = MAD \{y_N\} \tag{6.4}$$

is the median of the absolute deviations from the median [1, 2].

If $0 < \psi' \leqslant 1$, the estimate $\widehat{\theta}^{(q)}$ converges.

One of the limitations on deviations of $\psi(X)$ could be chosen by applying the function $\widetilde{\psi}(X)$:

$$\widetilde{\psi}_b(X) = [\psi(X)]_b^a = \begin{cases} -b, & \text{if} \quad \psi(X) - a < b, \\ \psi(X), & \text{if} \quad -b \leqslant \psi(X) - a < b, \\ b, & \text{if} \quad \psi(X) - a < b. \end{cases} \tag{6.5}$$

The simplest variant of equation (6.5) is the range of $\psi(X)$ in the form of the limiting Huber M estimator [1, 2, 4, 5] for normal distribution:

$$\widetilde{\psi}_b(X) = MIN\,(b, MAX(X, -b)) = MIN\left(1, \frac{b}{|X|}\right) = \\ = [X]_{-b}^{b} = \begin{cases} X & |X| < b, \\ b \cdot \operatorname{sgn}(X) & |X| \geqslant b. \end{cases} \tag{6.6}$$

Another way to determine the function $\widetilde{\psi}(X)$ is to illuminate tailed values of data, for example (as it was done by Hampel [2]), by applying the skipped median function:

$$\psi_{\mathrm{med}(r)}(X) = \operatorname{sgn}(X) \cdot 1_{[-r,r]}(X) = \begin{cases} \operatorname{sgn}(X), & |X| \leqslant r, \\ 0, & X > r. \end{cases} \tag{6.7}$$

Another possibility is to simplify the procedure to the simplest cut (skipped) function:

$$\psi_{cut(r)}(X) = X \cdot 1_{[-r,r]}(X) = \begin{cases} X, & |X| \leqslant r, \\ 0, & |X| > r. \end{cases} \tag{6.8}$$

There is a lot of different influence functions presented in literature [1, 2, 4, 5]: Hampel's three-part redescending function

$$\psi_{\alpha,\beta,r}(X) = \begin{cases} X, & 0 \leqslant |X| \leqslant \alpha, \\ \alpha \cdot \operatorname{sgn}(X), & \alpha \leqslant |X| \leqslant \beta, \\ \alpha \dfrac{r - |X|}{r - \beta}, & \beta \leqslant |X| \leqslant r, \\ 0, & r \leqslant |X|, \end{cases} \tag{6.9}$$

the Andrews sine function

$$\psi_{sin(r)}(X) = \begin{cases} \sin(X/r), & |X| \leqslant r\pi, \\ 0, & |X| > r\pi, \end{cases} \tag{6.10}$$

the Tukey biweight function

$$\psi_{bi(r)}(X) = \begin{cases} X^2\left(r^2 - X^2\right), & |X| \leqslant r, \\ 0, & |X| > r, \end{cases} \tag{6.11}$$

and the Bernoulli function

$$\psi_{ber(r)}(X) = X^2\sqrt{r^2 - X^2} \cdot 1_{[-r,r]}(X), \tag{6.12}$$

It has been shown that these functions can provide good suppression of impulsive and multiplicative noise [3, 4, 13–16]. Other influence functions are also used in literature: the Smith function, the Huber-Collins function, the scaled logistic ML function, the median type tangent hyperbolic function, etc.

6.2. R and L Estimators

Other types of estimators are the R and L estimators. R estimators belong to nonparametric robust estimators based on rank calculations [1, 2, 5]. Let use two samples for rank tests $x_1, \ldots, x_m$ and $y_1, \ldots, y_n$ as two samples with distributions $H(x)$ and $H(x + \Delta)$, where Δ is an unknown change for bias. Suppose that R_i is the rank of X_i in the sample of size $N = m + n$. The rank test of $\Delta = 0$ for $\Delta > 0$ is based on statistics test:

$$S = \frac{1}{m} \sum_{i=1}^{m} a(R_i). \tag{6.13}$$

To find coefficients or scores a_i, it is possible to use the function J:

$$a_i = J\left(\frac{i}{m + n + 1}\right). \tag{6.14}$$

There are another possibilities to derive the scores or coefficients a_i of J, for example,

$$a_i = J\left(\frac{i-\frac{1}{2}}{m+n}\right), \tag{6.15}$$

or

$$a_i = (m+n)\int\limits_{(i-1)/(m+n)}^{i/(m+n)} J(s)ds. \tag{6.16}$$

The last version is more preferable. The function $J(s)$ is symmetrical in the sense that $J(1-s) = -J(s)$ and satisfies the condition $\int J(s)ds = 0$, and coefficients a_i satisfy the condition $\sum\limits_{i=1}^{n} a_i = 0$. The statistical test S is the rank test for the localization of changes. If the observations $X_1, \ldots, X_n$ and its mirror images $2T_n - X_1, \ldots, 2T_n X_n$ have the same localization, the statistical test S detects the change of location and its values are close to zero. For every J and F, all coefficients yield the asymptotic test. For the Wilcoxon test, we find $J(t) = \left|t - \frac{1}{2}\right|$.

The test T_n based on *R estimator* (6.13) corresponds to the functional $T(G)$[1, 2]:

$$\int J\left[\frac{1}{2}G(y) + \frac{1}{2}\left(1 - G\left(2T(G) - y\right)\right)\right] dG(y) = 0. \tag{6.17}$$

The influence function $IF\,(x; T, F)$ can be calculated via G in the form $F_{t,x} = (1 - -t)F + t\delta_x$ in (6.17), and the derivation gives

$$IF\,(x; T, F) = \frac{U(x) - \int U(x)f(x)dx}{\int U'(x)f(x)dx}, \tag{6.18}$$

where $U(x) = \int_0^x J'\left[\frac{1}{2}\left(F(y) + 1 - F\left(2T(F) - y\right)\right)\right] f\left(2T(F) - y\right) d\lambda(y)$.

If F is a symmetric function, the following relations take place: $T(F) = 0$ and $U(x) = J\left(F(x)\right)$. Then,

$$IF\,(x; T, F) = \frac{J\left(F(x)\right)}{\int J'\left(F(x)\right) f(x)^2 dx}. \tag{6.19}$$

In the case of the distribution $\frac{1}{\sqrt{2\pi}}e^{-x^2/2}$, the $R-$estimator is the median $T_n = \mathrm{med}\,\{X_i\}$ if n is odd with the functional $T(G) = G^{-1}\left(\frac{1}{2}\right)$.

This equation for $T(G)$ is the classical definition of median: the median is the point x where $G(x) = \frac{1}{2}$. The influence function in this case with

$$J(t) = \begin{cases} -1 & t < 1/2, \\ 1 & t > 1/2 \end{cases} \text{ and coefficients } a_i = \begin{cases} 1 & i = (N+1)/2, \\ 0 & \text{another case.} \end{cases}$$

From (6.16) has the form

$$IF\,(x; T, F) = \frac{\mathrm{sgn}\,(x)}{2f(0)}. \tag{6.20}$$

So the corresponding rank estimator of X_i is the well known simplest R estimator, which can be written as [1, 2]

$$\widehat{\theta}_{\text{med}} = \begin{cases} \frac{1}{2}\left(X_{n/2} + X_{1+n/2}\right) & \text{for } n \text{ even}, \\ X_{(1+n/2)} & \text{for } n \text{ odd}, \end{cases} \tag{6.21}$$

where $X_{(j)}$ is the element with the rank j. Estimator (6.21) is known as the *median* of sample of data. It is the best estimator if there is no any *a priori* information about the sample distribution and moments of X_i [2, 4]. The Hodges-Lehmann estimator $J(t) = \left|t - \frac{1}{2}\right|$ is related to the Wilcoxon test $\Delta_n = \text{med}\{y_i - x_j\}$ and $T_n = \text{med}\left\{\frac{1}{2}(x_i + x_j)\right\}$. For a symmetric probability data distribution, this test is known as the most powerful asymptotically [7–9]. The Wilcoxon test has the influence function $IF(x; T, F) = \dfrac{F(x) - \frac{1}{2}}{\int f^2(y)dy}$ [2, 5] and the coefficients $a_i = \dfrac{2i - N - 1}{2N}$, where N equals the number of values in a sample.

The correspondent rank estimator is the Wilcoxon R-estimator [4, 5, 7, 9]:

$$\widehat{\theta}_{Wil} = \underset{i \leqslant j}{\text{med}}\left\{\frac{1}{2}\left(X_{(i)} + X_{(j)}\right), i, j = 1, \ldots N\right\}, \tag{6.22}$$

where $X_{(i)}$, $X_{(j)}$ are elements with ranks i and j, respectively. Equation (6.22) is robust and is the best estimator when data distribution function has a symmetrical form [1, 2].

In a similar way, using different $J(t)$, one can obtain other R estimations [10, 11].

The Ansari–Bradley–Siegel–Tukey function $J(t) = \left|t - \frac{1}{2}\right| - \frac{1}{4}$ has the coefficients $a_i = \dfrac{2i - \frac{3}{2}N - 1}{2N}$, and the corresponding R estimator can be written in the form

$$\theta_{ABST} = \text{med}\left\{\begin{array}{ll} X_{(i)} & i \leqslant N/2 \\ \frac{1}{2}\left(X_{(i)} + X_{(j)}\right) & i > N/2 \end{array}\right\}. \tag{6.23}$$

This estimator represents the use of two aforementioned estimators (6.21) and (6.22). Suppose that the sample has the values X_i with the size $N = 1, \ldots, 9$, the first four ranks $X_{(i)}$ are defined according to (6.21), and the other ones are defined as in the Wilcoxon estimator (6.22).

The Mood function is $J(t) = \left(t - \frac{1}{2}\right)^2 - \frac{1}{12}$ with $a_i = \dfrac{i^2 - i - \frac{1}{3}}{N^3} - \dfrac{2i - \frac{1}{3}N - 1}{2N^2}$ and the R estimator is presented in the form

$$\theta_{MOOD} = \text{med}\left\{\begin{array}{ll} \frac{1}{2}\left(X_{(i)} + X_{(j)}\right) & i \leqslant 3 \\ X_{(i)} & i > 3 \end{array}\right\}$$

for all $i = 1, \ldots, N$.

6.3. Robust Properties of Estimators

Robust properties of estimators are determined by the definition of robustness as a property to retain the effectiveness of estimator when noise characteristics could be changed during the experiments. Different aspects of robustness can be described by

the *influence functions* (IF), c*hange of variance function* (CVF), and the *breakdown point* ε^* [1, 2, 4, 7]. Hampel [2] studied robustness by means of the influence function IF. The IF describes the infinitesimal stability effect for asymptotic values of point x for the estimator. The influence function IF of T in F is given by the equation:

$$IF\,(x;T,F) = \lim_{t\to 0} \frac{T\,((1-t)F + t\Delta_x) - T(F)}{t}. \tag{6.24}$$

The *gross-error sensitivity* of T in F can be presented in the following form:

$$\gamma^* = \sup_x |IF\,(x;T,F)|\,. \tag{6.25}$$

It is desirable that γ^* be finite. In this case, one can say that T is B robust, where B means the bias. Typically, establishing that γ^* is finite is the first step to finding out whether the estimator is robust. This cannot have any conflict with the asymptotic efficiency.

Another variable for calculating the robustness is the *breakdown point* ε^* of the estimator sequences $\{T_n; n \geqslant 1\}$ in F and is defined as [1, 2, 4]

$$\varepsilon^* = \sup\{\varepsilon\,|b\,(\varepsilon) < b(1)\}\,, \tag{6.26}$$

where $b\,(\varepsilon) = \lim\limits_n \sup\limits_{F\in P} |M\,(F,T_n)|$ and $M\,(F,T_n)$ is the median of the distributions of $[T_n - T\,(F_0)]$.

6.3.1. Asymptotic Efficiency. The selection of function for the M, L and Restimators is somewhat heuristic. Usually, the Frechet differentiation is used [1, 2].

It is supposed that probability distributions $(F_\theta)_{\theta\in\Theta}$ are from a parametric family and the functional T is the uniform Fisher estimation for θ, defined as $T\,(F_\theta) = \theta$, for all θ. In this case, T is a differential in the Frechet form for F. The corresponding estimation is asymptotically efficient in F_θ, and their influence function satisfies the equation

$$IF\,(x;F_\theta,T) = \frac{1}{I\,(F_\theta)} \frac{\partial}{\partial\theta}\,(\log\, f_\theta)\,, \tag{6.27}$$

where f_θ is the probability function of F_θ and $I\,(F_\theta) = \int \left(\frac{\partial}{\partial\theta}\,\log\, f_\theta\right)^2 dF_\theta$ is the Fisher information.

For every function of the M, L or R estimation it is necessary to solve equation (6.27) for the bias parameter $f_\theta(x) = f_0(x-\theta)$.

(1) For M estimator, it is sufficient to find

$$\psi(x) = -c\frac{f_0'(x)}{f_0(x)}, \quad c \neq 0 \tag{6.28}$$

and compare it with the influence function

$$IF\,(x;F,T) = \frac{\psi\,[x - T(F)]}{\int \psi'\,[x - T(F)]\,F(dx)}.$$

(2) For L estimator, it is necessary to use $h(x) = x$ and a sufficient m as [5]

$$m\,(F_0(x)) = \frac{-1}{I\,(F_0)}\,(\log\, f_0(x))^n\,. \tag{6.29}$$

(3) For R estimations and symmetrical F_0, it is sufficient to find the function J as [2]

$$J(F_0(x)) = -c\frac{f_0'(x)}{f_0(x)}, \quad c \neq 0 \tag{6.30}$$

6.3.2. Some Examples of Robustness. Below we present different functions to which one can apply some criteria of robustness.

Distribution: Normal

$$f_0(x) = \frac{1}{\sqrt{2\pi}}\mathrm{e}^{-x^2/2}.$$

Estimation: M: $\psi(x) = x$ mean, not robust with $\gamma^* = \infty$ and $\varepsilon^* = 0$,

L: $m(t) = 1$ mean, but not robust,

R: $J(t) = \Phi^{-1}(t)$ estimator of normal score, robust with $\gamma^* = \infty$ and $\varepsilon^* = 2\Phi\left(-\sqrt{\ln 4}\right) \approx 0.239$.

Distribution: Logistic

$$F_0(x) = \frac{1}{1+e^{-x}}.$$

Estimation: M: $\psi(x) = \tanh\left(\frac{x}{2}\right)$ robust;

L: $m(t) = 6t(1-t)$ not robust, $\gamma^* < \infty$;

R: $J(t) = t - \frac{1}{2}$ Hodges–Lehmann, robust with $\varepsilon^* = 1 - \frac{1}{\sqrt{2}} \approx 0.293$.

Distribution: Cauchy Distribution

$$f_0(x) = \frac{1}{\pi\left(1+x^2\right)}.$$

Estimation: M: $\psi(x) = \frac{2x}{1+x^2}$ robust,

L: $m(t) = 2\cos(2\pi t)\left[\cos(2\pi t) - 1\right]$ not robust,

R: $J(t) = -\sin(2\pi t)$ robust.

Distribution: Minimum Information

$$f_0(x) = \begin{cases} Ce^{-x^2/2} & |x| \leqslant c, \\ Ce^{-c|x|+c^2/2} & |x| > c. \end{cases}$$

Estimation: M: $\psi(x) = \max\left[-c, \min(c, x)\right]$ is Huber estimator, robust with $\gamma^* = 1.037$ and $\varepsilon^* = \frac{1}{2}$,

L:

$$m(t) = \begin{cases} \frac{1}{1-2\alpha}, & \alpha < t < 1-\alpha, \alpha = F_0(-c), \\ 0, & \text{another case} \end{cases}$$

is α-trimmed mean,

robust with $\gamma^* = 1.167$ and $\varepsilon^* = \alpha$,

R: The corresponding estimator [2], but with complicated description, robust.

The R estimation with the normal scores in the case of Gaussian distribution has unlimited influence curve and has *gross-error sensitivity* $\gamma^* = \infty$. This estimator is robust but in practice it is necessary to modify it because its indicators of robustness $b(\varepsilon)$ and $v(\varepsilon)$ (*maximum asymptotic bias and variance*) increase rapidly.

6.4. RM Estimators

Sometimes, equation (6.3) can be simplified to the following one-step estimator [3–5]:

$$\theta_{\mathbf{M}} \cong \frac{\sum\limits_{i=1}^{N} y_i \widetilde{\psi}\left(y_i - MED\left\{Y_N\right\}\right)}{\sum\limits_{i=1}^{N} 1_{[-r.r]} \widetilde{\psi}'\left(y_i - MED\left\{Y_N\right\}\right)}, \tag{6.31}$$

where

$$\cdot 1_{[-r,r]}(X) = \begin{cases} 1, & |X| \leqslant r, \\ 0, & |X| > r, \end{cases}$$

but the optimal estimator is defined in equation (6.3) and it will be used below in the proposed RM filtering scheme.

It is evident that formula (6.31) represents the arithmetic average of $\sum\limits_{i=1}^{n} \psi\left(y_i - MED\left\{y_N\right\}\right)$ evaluated on the interval $[-r, r]$, where the parameter r is connected with restrictions on the range of $\psi(y)$ as it was done in the case of the simplest Huber's limiter type M-estimator $\widetilde{\psi}_r(y) = \min\left(r, \max(y, -r)\right) = [y]_{-r}^{r}$ for the normal distribution contaminated by another one with a heavy 'tails'[1, 2].

Another way to derive the function $\widetilde{\psi}(y)$ is to cut the outliers from the primary sample. This leads to the so-called lowered M-estimates. Below we use the different influence functions defined in equations (6.9)–(6.12).

The proposal for enhancement of the robust properties of M-estimators by using the rank estimates consists in the application of the procedure similar to the median operator instead of arithmetic average. We present here the next iterative RM-estimators that follow from equation (6.3) [4-6, 10]:
the Median M-type estimator

$$\theta_{M\mathbf{M}}^{(q)} = MED\left\{y_i \widetilde{\psi}\left(y_i - \theta^{(q-1)}\right), i = 1, 2, \ldots, N\right\}, \tag{6.32}$$

the Wilcoxon M-type estimator

$$\theta_{W\mathbf{M}}^{(q)} = \underset{i \leqslant j}{MED} \left\{ \frac{1}{2} \left[y_i \widetilde{\psi}\left(y_i - \theta^{(q-1)}\right) + y_j \widetilde{\psi}\left(y_j - \theta^{(q-1)}\right) \right], \right.$$
$$\left. i = 1, 2, \ldots, N; j = 1, 2, \ldots, N \right\}, \tag{6.33}$$

the Ansari–Bradley–Siegel–Tukey estimator

$$\theta_{ABST\mathbf{M}}^{(q)} = MED \left\{ \begin{array}{ll} y_i \widetilde{\psi}\left(y_i - \theta^{(q-1)}\right), & 1 \leqslant i \leqslant \left[\frac{N}{2}\right], \\ \frac{1}{2}\left[y_i \widetilde{\psi}\left(y_i - \theta^{(q-1)}\right) + \right. & \\ \left. + y_j \widetilde{\psi}\left(y_j - \theta^{(q-1)}\right) \right], & \left[\frac{N}{2}\right] < i, j \leqslant N; i \leqslant j \end{array} \right\}, \tag{6.34}$$

where y_i and y_j are input data samples, initial estimate is $\widehat{\theta}^{(0)} = MED\left\{y_N\right\}$, and $\{y_N\}$ is the primary data set. Equations (6.32)–(6.34) can be also applied in processing of the 2D and 3D data [4, 5, 11–17]. Here, an input sample is formed by pixels in a sliding window (or cube for 3D data) used in the image processing. The presented estimators are the iterative combined RM-estimators.

Additional way to enhance the robust properties of the *RM*-estimators (6.32)–(6.34) considered here is the use of the iterative form of such estimators that follows from equation (6.3).

References

1. *Huber, P.J.* Robust Statistics, Wiley, New York, 1981.
2. *Hampel, F.R., Ronchetti, E.M., Rouseew, P.J., and Stahel, W.A.* Robust Statistics: The approach based on influence function, Wiley, New York, 1986.
3. *Astola, J., Haavisto, P., and Neuvo, Y.* Vector median filters. Proc. of the IEEE 1990; 78(4): 678–89.
4. *Ponomaryov, V.I. and Pogrebnyak, O.B.* Novel robust RM filters for radar image preliminary processing. Journal of Electronic Imaging, October 1999, vol. 08, no. 4, pp. 467–477.
5. *Gallegos-Funes, Francisco, J. and Ponomaryov, V.I.* Real-time image filtering scheme based on robust estimators in presence of impulsive noise. Real Time Imaging, 8(2), pp. 69–80, Elsevier, 2004.
6. *Ponomaryov, V.I., Gallegos-Funes F.J., and Nino-de-Rivera, L.* RM-Filters Applicable in Image Real-Time Processing. Electromagnetic Waves and Electronic Systems, 2003, vol. 8, no 7–8, pp. 81–89.
7. *Hodges, J.L. and Lehmann, E.L.* Estimates of location based on rank test. Ann. Math. Statist., 963, vol. 34, pp. 598–611.
8. *Kundu, A. and Wu, W.R.* Double-window Hodges-Lehmann (D) filter and hybrid D-median filter for robust image smoothing. IEEE Trans. Acoust., Speech, and Signal Processing, 1989, vol. 37, pp. 1293–1298.
9. *Crinon, R.J.* The Wilcoxon filter: a robust filtering scheme. Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing 85, 1985, pp. 668–671.
10. *Kravchenko, V.F., Pogrebnyak, A.B., and Ponomarev, V.I.* Robust algorithms of noise filtration for radar image processing. Telecommunications and Radio Engineering, Begell Hose Edit... vol. 56 no. 12, pp. 44–55, 2001.
11. *Ponomaryov, V., Gallegos-Funes, F., and Rosales-Silva, A.* Adaptive multichannel non parametric median M-type K-nearest neighbour (AMN-MMKNN) filter to remove impulsive noise from color images. Electronics Letters. 2004, vol. 40, no. 13, pp. 796–798.
12. *Ponomaryov, V., Gallegos-Funes, F., and Rosales-Silva A.* Real-Time Color Image Processing Using Order Statistics Filters. Journal of Mathematical Imaging and Vision. Springer. 2005, vol. 23, no. 3, pp. 315–319.
13. *Kravchenko, V.F., Ponomaryov, V.I., and Pustovoit V.I.* Algorithms of Robust Image Filtering with the Preservation of Fine Details in the Presence of Noise. Doklady Physics, «Nauka/Inerperiodica»... N.Y. Sept. 2001, vol. 46, Issue 9 pp. 642–646.
14. *Kravchenko, V.F, Ponomaryov, V.I., Pustovoit, V.I., and Nino-de-Rivera, L.* New Robust Nonlinear Image-Filtering Algorithms Preserving Small Features in the Presence of Noise. Journal of Communications Technology and Electronics, 2002. vol. 47, no. 1, pp. 67–74.
15. *Gallegos-Funes F.J., Ponomaryov V.I., Sadovnychiy S., and Nino-de-Rivera L.* Median M-type K-nearest neighbour (MM-KNN) filter to remove impulse noise from corrupted images. Electronics Letters, 2002, vol. 38, no. 15, pp. 786–787.
16. *Kravchenko, V.F., Ponomaryov, V.I., Pustovoit, V.I., and Sanchez-Garcia, J.C.* Nonlinear robust filtration of image sequences degraded by noise. Doklady Physics, «Nauka/Inerperiodica». N.Y. 2003. vol. 48, no. 6.
17. *Ponomaryov, V.I. and Nino-de-Rivera, L.* Order statistics — M Method in Image and Video sequence Processing Applications. Electromagnetic Waves and Electronic Systems, 2003, vol. 8, no 7–8, pp. 99–107.

Chapter 7

LINEAR AND NONLINEAR FILTERING TECHNIQUES

In this chapter, a short review of different kinds of frequently used algorithms is presented. In most cases, we analyze only the algorithms connected with order statistics as the most promising approach.

7.1. Trimmed Mean Filters

Usually, this type of filters can be utilized to suppress both Gaussian and impulsive noises [1, 2].

7.1.1. Filter α-trimmed mean. This filter can be designed by ordering the data sample and cutting the tailed values before number r and after $n-s$, i.e., $X_{(1)}$, $X_{(2)}$, ..., $X_{(r)}$ and $X_{(n-s+1)}$, $X_{(n-s+2)}, \ldots, X_{(n)}$ [3, 4]. Then,

$$\theta_{\alpha TM} = \frac{1}{n-r-s} \sum_{i=r+1}^{n-s} X_{(i)}. \tag{7.1}$$

The α-*trimmed mean filter* is one compromise between median and classical filters.

7.1.2. Filter KNN. This *KNN* filter (*K-nearest neighbor filter*) [5] is the modification of the *trimmed mean filter* and is the mean of Kelements for $1 \leqslant K \leqslant n$ with the values nearest to central pixel X^* in the sliding window. This helps to preserve edges and fine small details in an image. This filter can also help to suppress impulsive noise. One adaptation of such a filter is [4]:

$$\theta_{KNN} = \frac{\sum_{i=1}^{n} a_i X_i}{\sum_{i=1}^{n} a_i}, \tag{7.2}$$

where θ_{KNN} represents the *KNN* modified filter and

$$a_i = \begin{cases} 1, & \text{if} \quad |X_i - X^*| \leqslant T, \\ 0, & \text{another case.} \end{cases}$$

If threshold T is equal to twice of noise deviation σ, the *KNN* modified filter is the same as *Sigma* filter [6].

7.2. L Filters

These filters are usually employed to suppress complex noises (impulsive, Gaussian, and multiplicative).

7.2.1. Filter L. This filter has the structure very similar to the lineal *FIR* filters [7]:

$$\theta_L = \sum_{i=1}^{n} a_i \cdot X_{(i)}, \tag{7.3}$$

where $X_{(i)}$, $i = 1,\ldots,n$, are a sequence of ordered data and a_i, $i = 1,\ldots,n$, are the filter coefficients, e.g., as in following equation:

$$a_i = \frac{\int_{\frac{i-1}{n}}^{\frac{i}{n}} h(\lambda)d\lambda}{\int_{0}^{1} h(\lambda)d\lambda}, \tag{7.4}$$

where h: $[0, 1] \rightarrow \Re$ satisfies the condition integral $\int_0^1 h(\lambda)d\lambda \neq 0$ and $\sum_{i=1}^{n} a_i = 1$.

For every known distribution, it is easy to find the filter's coefficients optimizing the MSE of the filter. For the uniform, Gaussian, and Laplacian distributions, the filter coefficients are presented in Table 7.1 [1].

Table 7.1. Optimal coefficients for L filter 3×3.

	Uniform	Gaussian	Laplacian
a_1	0.5	0.11	0.019
a_2	0.0	0.11	0.0291
a_3	0.0	0.11	0.0697
a_4	0.0	0.11	0.2380
a_5	0.0	0.11	0.3647
a_6	0.0	0.11	0.2380
a_7	0.0	0.11	0.0697
a_8	0.0	0.11	0.0291
a_9	0.5	0.11	0.019

7.2.2. C- and Lλ -Filter [8]. This filter is a special case of the $L\lambda$ filter [9] and includes *FIR* (*finite impulse response*) and L filter in the following way:

- It is an L filter with the coefficients of X_i depending on the temporal position in the filter window.
- It is a C *FIR* filter with the coefficients of X_i depending on the rank in the window.

So, for filter C $n \times n$ and data $\left(X_{(1)}, X_{(2)}, \ldots, X_{(n)}\right)$, it can be found that

$$\theta_C = \frac{\sum_{i=1}^{n} c\left(R\left(X_i\right), i\right) X_i}{\sum_{i=1}^{n} c\left(R\left(X_i\right), i\right)}, \tag{7.5}$$

where $R(X_i)$ is the ranked X_i, $c(R(X_i), i)$ is filter's coefficient according to the rank of $R(X_i)$,

$$C = \left(\overbrace{a^T, \ldots, a^T}^{12\text{times}}, b^T, \overbrace{a^T, \ldots, a^T}^{12\text{times}} \right)$$

is the matrix of coefficients for a 5×5 window and coefficients of vectors $\mathbf{a}$ and $\mathbf{b}$ are defined below in Tables 7.2 and 7.3.

Table 7.2. Coefficients of vector $\mathbf{a}$.

a_1, a_{25}	0.00550	a_7, a_{19}	0.00314
a_2, a_{24}	0.00335	a_8, a_{18}	0.01064
a_3, a_{23}	−0.00427	a_9, a_{17}	0.02907
a_4, a_{22}	−0.00101	a_{10}, a_{16}	0.06499
a_5, a_{21}	−0.00008	a_{11}, a_{15}	0.11835
a_6, a_{20}	0.00065	a_{12}, a_{14}	0.17195
a_{13}	0.19541		

Table 7.3. Coefficients of vector $\mathbf{b}$.

b_1, b_{25}	0.00550	B_5, b_{21}	−0.00008
b_2, b_{24}	0.00335	b_6, b_{20}	0.00065
b_3, b_{23}	−0.00427	b_7, b_{19}	0.00314
b_4, b_{22}	−0.00101	$b_{8,\ldots,18}$	6.0

7.2.3. Filter LMS-L (Least Mean-Square L-Filter). The *LMS-L* filter [10] is an extension of the L filter for nonstationary signals.

The *location-invariant LMS-L filter* [10] was designed for the cases when signals are contaminated by additive white noise:

$$\widehat{\mathbf{a}}'(i+1) = \widehat{\mathbf{a}}'(i) + \mu e(i) \mathbf{x}'_L(i), \tag{7.6}$$

where $e(i) = x(i) - \theta_L(i)$ is the error of estimation in the pixel i, x is the original image, and $\theta_L = \mathbf{a}^T(i)\mathbf{x}_L(i)$ is the output of filter L. Parameter μ is chosen in the interval $0 < \mu < \dfrac{2}{3 \cdot \text{tr}\,[\mathbf{R}_L]}$, where tr $[\mathbf{R}_L]$ is the trace of the correlation matrix of sample $\mathbf{R}_L = E\left\{\mathbf{x}_L(i)\mathbf{x}_L^T\right\}$ of ordered pixel values.

The coefficients form the vector

$$a(i) = \left(\mathbf{a}_1^T(i) \left| a_v(i) \right| \mathbf{a}_2^T(i)\right)^T, \tag{7.7}$$

where $v = (n+1)/2$, and $\mathbf{a}_1(i)$ and $\mathbf{a}_2(i)$ are vectors of the dimension $(n-1)/2 \times 1$: $\mathbf{a}_1(i) = (a_1(i), \ldots, a_{v-1}(i))^T$ and $\mathbf{a}_2(i) = (a_{v+1}(i), \ldots, a_n(i))^T$. The coefficient for the central pixel is $a_v(i) = 1 - \mathbf{1}_{v-1}^T \mathbf{a}_1(i) - \mathbf{1}_{v-1}^T \mathbf{a}_2(i)$.

The input vector is defined as $\mathbf{x}_L(i) = \left(\mathbf{x}_{L1}^T(i) \left| x_{(v)}(i) \right| \mathbf{x}_{L2}^T(i) \right)^T$. So $\mathbf{a}'(i)$ is the vector of the L filter's coefficients

$$\mathbf{a}'(i) = \left(\mathbf{a}_1^T(i) \left| \mathbf{a}_2^T(i) \right.\right)^T \tag{7.8}$$

and $\mathbf{x}'_L(i)$ is the $(n-1) \times 1$ vector

$$\mathbf{x}'_L(i) = \begin{bmatrix} \mathbf{x}_{L1}(i) - x_{(v)}(i)\mathbf{1}_{v-1} \\ \mathbf{x}_{L2}(i) - x_{(v)}(i)\mathbf{1}_{v-1} \end{bmatrix}. \tag{7.9}$$

In practice, the pure image (free of noise) usually cannot be observed, so equation (7.6) is transformed to the following one:

$$\widehat{\mathbf{a}}'(i+1) = \widehat{\mathbf{a}}'(i) + \mu y(i)\mathbf{x}'_L(i). \tag{7.10}$$

Another version of the *LMS-L* filter is the *unconstrained LMS-L filter,* where the coefficients are defined as

$$\widehat{\mathbf{a}}(i+1) = \widehat{\mathbf{a}}(i) + \mu e(i)\mathbf{x}_L(i). \tag{7.11}$$

Normalized LMS-L filter [10] uses a modified equation (7.11) where parameter μ is changed as follows:

$$\widehat{\mathbf{a}}(i+1) = \widehat{\mathbf{a}}(i) + \frac{\mu_0}{\|\mathbf{x}_L(i)\|^2} e(i)\mathbf{x}_L(k), \tag{7.12}$$

where μ_0 must be taken in interval $0 < \mu_0 \leqslant \frac{2}{3}$.

The *signed error LMS-L filter*, according to the MAE (mean absolute value) criterion, can be written as

$$\widehat{\mathbf{a}}(i+1) = \widehat{\mathbf{a}}(i) + \mu \operatorname{sgn}\left[e(i)\right] \mathbf{x}_L(i). \tag{7.13}$$

The equation presented above uses the vector of ordered sample. The estimation of error can be realized in the form

$$e'(i) = X(i) - \widehat{\mathbf{a}}^T(i+1)\mathbf{x}_L(i) = e(i) - \mu \operatorname{sgn}\left[e(i)\right] \mathbf{x}_L^T(i)\mathbf{x}_L(i), \tag{7.14}$$

where $\mu(i) = \dfrac{\mu_0 |e(i)|}{\mathbf{x}_L^T(i)\mathbf{x}_L(i)}$, $0 < \mu_0 < 1$, and $|e'(i)|$ is $(1-\mu_0) \cdot |e(i)|$.

7.3. Weighted Median and Order Statistics Filters

7.3.1. Weighted Median Filters. The weighted median filter has better preservation of edges in comparison with the standard median filter but a worse impulse noise suppression.

The definition of this filter is

$$\theta_{WM} = MED\left\{a_i \diamondsuit X_i, \quad i = 1, \ldots, n\right\}, \tag{7.15}$$

where operator $\diamondsuit$ means that the element X_i is used a_i times and a_i are the filter coefficients $a_i \geqslant 0$ for $i = 1, 2, \ldots, n$. Examples of weighted filters are [11, 12] the filter with matrix a_1 and the *Center Weighted Median Filter* with matrix a_2:

$$a_1 = \begin{bmatrix} 1 & 1 & 2 & 1 & 1 \\ 1 & 3 & 4 & 3 & 1 \\ 2 & 4 & 11 & 4 & 2 \\ 1 & 3 & 4 & 3 & 1 \\ 1 & 1 & 2 & 1 & 1 \end{bmatrix} \quad \text{and} \quad a_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 13 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{7.16}$$

7.3.2. Ranked-Order and Weighted Order Statistics Filters. *Ranked-Order filters* are the modifications of median filter. The output of the filter is presented by the window of the following type:

$$RO\left(X_1, X_2, \ldots, X_n; r\right) = X_{(r)}. \tag{7.17}$$

Weighted Order Statistics Filters can be defined in a similar way:

$$WOS\left(X_1, X_2, \ldots, X_n; a, r\right) = order \text{ statistics } r \text{ of } \left\{a_1 \diamondsuit X_1, a_2 \diamondsuit X_2, \ldots, a_n \diamondsuit X_n\right\}. \tag{7.17a}$$

7.3.3. Multistage Median Filter. This filter type allows one to realize fast calculation of median. It uses different stages of median filters to calculate the median of medians.

S*eparable two-dimensional median filter* [13] is a modification of the median filter and can calculate the output very fast. It has two stages. In the first stage, it calculates horizontal medians, and in second stage, it calculates the median in the vertical direction. Multistage median filters are presented as follows:

$$\begin{aligned}
MSM_1 &= MED\left\{X_{m+1,m+1}, h - \text{med}, v - \text{med}\right\},\\
MSM_2 &= MED\left\{X_{m+1,m+1}, d45 - \text{med}, d135 - \text{med}\right\},\\
MSM_3 &= MED\left\{h - \text{med}, v - \text{med}, d45 - \text{med}, d135 - \text{med}\right\},\\
MSM_4 &= MED\left\{X_{m+1,m+1}, h - \text{med},\right.\\
&\qquad \left. v - \text{med}, d45 - \text{med}, d135 - \text{med}\right\},\\
MSM_5 &= MED\left\{X_{m+1,m+1}, MED\left\{X_{m+1,m+1}, h - \text{med}, v - \text{med}\right\},\right.\\
&\qquad \left. MED\left\{X_{m+1,m+1}, d45 - \text{med}, d135 - \text{med}\right\}\right\},\\
MSM_6 &= MED\left\{X_{m+1,m+1}, c - \text{med}, x - \text{med}\right\},
\end{aligned} \tag{7.18}$$

where m is found from the equality $z = 2m + 1$ and $z \times z = n$.

7.3.4. Adaptive Center Weighted Median Filter. This filter uses an adaptive operator to estimate differences between an actual pixel and outputs of the central median filter with variations of weights in the central pixel. The filter is defined as [14]:

$$\theta_{ACWM} = \begin{cases} \theta^1_{CWM} & d_k > T_k, \\ X^* & \text{another case}, \end{cases} \tag{7.19}$$

where $d_k = |\theta^m_{CWM} - X^*|$ are the differences between the median filter and the central weighted median filter, $\theta^m_{CWM} = MED\left\{X_i, m \diamondsuit X^*\right\}$ is the central weighted median filter, $T_k = s \cdot MAD + \delta_k$ are the thresholds of the filter, and $MAD = MED\left\{\left|X_i - \theta^1_{CWM}\right|\right\}$. Parameter s determines the necessary robustness and varies in the interval $0 \leqslant s \leqslant 0.6$; $\delta_k = [\delta_0, \delta_1, \delta_2, \delta_3] = [40, 25, 10, 5]$ [14].

7.3.5. LUM Filter. The *LUM* (*Lower-Upper-Middle*) filter is a filter with selections of ranks and has been proposed to preserve edges [15]:

$$\theta_{LUM} = \begin{cases} X_{(s)} & \text{if} \quad X^* < X_{(s)}, \\ X_{(t)} & \text{if} \quad X_{(t)} < X^* \leqslant t_L, \\ X_{(n-t+1)} & \text{if} \quad t_L < X^* \leqslant X_{(n-t+1)}, , \\ X_{(n-s+1)} & \text{if} \quad X_{(n-s+1)} < X^*, \\ X^* & \text{another case}, \end{cases} \tag{7.20}$$

where $t_L = \dfrac{X_{(t)} + X_{(n-t+1)}}{2}$ for $1 \leqslant t \leqslant k+1$, parameters s and t are found in the interval $1 \leqslant s \leqslant t \leqslant k+1$, and X^* is the median of the data sample.

7.3.6. Rank-Ordered Mean (ROM) Filter. The ROM filter is designed to suppress a noise of complex mixture. For a 3×3 window, the central element is X_5, so the output of ROM filter for other ordered samples is defined as:

$$ROM(X_i) = \frac{X_4 + X_5}{2}. \tag{7.21}$$

Also, an impulse detector that determines noisy pixels is used, which calculates the differences

$$d_k = \begin{cases} X_{(k)} - X^*, & X^* \leqslant ROM(X_i), \\ X^* - X_{(9-k)} & X^* > ROM(X_i). \end{cases} \tag{7.22}$$

Four thresholds $T_1 < T_2 < T_3 < T_4$ are used for comparing with them the differences:

$$d_k > T_k, \quad k = 1, \ldots, 4. \tag{7.23}$$

Finally, *Signal-Dependent Rank-Ordered Mean Filter* operates according to following equation:

$$SDROM(X_i) = \begin{cases} ROM(X_i) & d_k > T_k, \\ X^* & \text{another case.} \end{cases} \tag{7.24}$$

The optimal values T_k can be chosen in the intervals $T_1 \leqslant 15$, $15 \leqslant T_2 \leqslant 25$, $30 \leqslant T_3 \leqslant \leqslant 50$, and $40 \leqslant T_4 \leqslant 60$.

Training can be used to obtain the optimal parameters according to the MSE criterion, which gives

$$\alpha_i = \frac{-\sum\limits_{i:s(i)=n} (X(i) - ROM(i))\,(ROM(i) - V(i))}{\sum\limits_{i:s(i)=n} (X(i) - ROM(i))^2}, \tag{7.25}$$

where $i = 1, 2, \ldots, M$, $V(i)$ is the training image, $\beta_i = 1 - \alpha_i$, and $M = 1296$ is number of samples...

Finally, the *SDROM* filter [16] can be defined by the formula

$$SDROM(X_i) = \alpha_{s(i)} X(i) + \beta_{s(i)} ROM(X_i). \tag{7.26}$$

7.3.7. Vector Median Filter. This filter has been designed for multichannel images, e.g., color ones.

The output of the filter is defined as a vector with the minimum distance sum and can be written as [17]

$$\mathbf{x}_{VM}^{(l)} = F_{VM}\left(\mathbf{X}^{(l)}\right) = \mathbf{x}_1^{(l)}, \tag{7.27}$$

where $\mathbf{x}_1^{(l)}$ is a value with rank *one* found after calculating all the distances $d_i^{(l)}$ according to the scheme $d_{(1)}^{(l)} \leqslant d_{(2)}^{(l)} \leqslant \cdots d_{(n)}^{(l)}$, and $\mathbf{x}_1^{(l)}$ corresponds to the distance $d_{(1)}^{(l)}$. The distance $\mathbf{x}_i^{(l)}$ is calculated as

$$d_i^{(l)} = \sum_{j=1}^{N} F_d\left(\mathbf{x}_i^{(l)}, \mathbf{x}_j^{(l)}\right), \tag{7.28}$$

where F_d is usually a L_2 norm. Another type of such filters, the *directional* filter uses the F_d norm defined as the angles between vectorial pixels:

$$F_d\left(\mathbf{x}_i^{(l)}, \mathbf{x}_j^{(l)}\right) = \arccos\left(\frac{\mathbf{x}_i^{(l)^T}\mathbf{x}_j^{(l)}}{\left\|\mathbf{x}_i^{(l)}\right\|\left\|\mathbf{x}_j^{(l)}\right\|}\right). \tag{7.29}$$

This technique and its modifications will be used below in filtering of multichannel multidimensional images.

7.4. Examples of Data-Dependent Filters

7.4.1. Lee filter. Lee [18] proposed the method of local statistics in the *LLMMSE* (*local linear minimum mean square error estimator)* [18, 19]:

$$LLMMSE = \left(1 - \frac{\sigma_n^2}{\sigma_s^2}\right) X^* + \frac{\sigma_n^2}{\sigma_s^2} m, \tag{7.30}$$

where σ_s is the RMS of the original signal X_i, σ_n is the RMS of noise, X^* — a central pixel, m is the mean of a sample, and $0 \leqslant \sigma_n^2/\sigma_s^2 \leqslant 1$. For homogeneous areas, this filter operates as a mean- squares filter, but, if the changes are large near the central pixel, it is not practically changed. The previous equation can be transformed as follows:

$$LLMMSE = \frac{\sigma_s^2}{\sigma_s^2 + \sigma_n^2} X^* + \left(1 - \frac{\sigma_s^2}{\sigma_s^2 + \sigma_n^2}\right) m, \tag{7.31}$$

where the *MAD* can be used to estimate σ_s of the signal part.

7.4.2. Modified Frost Filter. The *Frost* filter has been designed as a filter to suppress multiplicative noise [20]:

$$X_{FROST} = \sum_{p,q} \theta s_{ij}^2 exp\left\{-a s_{ij}^2 (|p-i| + |q-j|)\right\} X_{pq}, \tag{7.32}$$

where p and q show the pixels in a sliding filtering window, X_{pq} is a pixel contaminated by the noise, $a = 4/\left(\sigma_\mu^2\sqrt{N}\right)$, σ_μ^2 is the variance of multiplicative noise, N is the number of pixels, $s_{ij}^2 = \sigma_{ij}^2/\overline{I}_{ij}^2$, $\overline{I}_{ij}$ and σ_{ij}^2 are the mean and local variance, and θ is normalization factor.

The equation can be written in another form as

$$X_{FROST} = \sum_{pq} b_{\Delta p \Delta q}\left(s_{ij}^2\right) X_{pq}, \tag{7.33}$$

where $b_{\Delta p \Delta q}\left(s_{ij}^2\right) = \theta s_{ij}^2 \exp\left\{-a s_{ij}^2 (\Delta p + \Delta q)\right\}$.

Modified Frost filter is based on the previous formula but depends of sub filters and thresholds:

$$X_{MFROST} = \begin{cases} \dfrac{1}{M}\displaystyle\sum_{p,q} B^0_{\Delta p \Delta q} X_{pq}, 0 \leqslant s_{ij}^2 < t_1, \\ \dfrac{1}{M}\displaystyle\sum_{p,q} B^1_{\Delta p \Delta q} X_{pq}, t_1 \leqslant s_{ij}^2 < t_2, \\ \dfrac{1}{M}\displaystyle\sum_{p,q} B^2_{\Delta p \Delta q} X_{pq}, t_2 \leqslant s_{ij}^2 < t_3, \\ \dfrac{1}{M}\displaystyle\sum_{p,q} B^3_{\Delta p \Delta q} X_{pq}, t_3 \leqslant s_{ij}^2 < \infty, \end{cases} \tag{7.34}$$

where $M = 64$ for a 5x5 window, $t_1 = 0.41$, $t_2 = 1.02$, $t_3 = 1.52$, and the coefficients are defined as follows:

$$B^0_{\Delta p \Delta q} = \begin{bmatrix} 2 & 2 & 3 & 2 & 2 \\ 2 & 3 & 3 & 3 & 2 \\ 3 & 3 & 4 & 3 & 3 \\ 2 & 3 & 3 & 3 & 2 \\ 2 & 2 & 3 & 2 & 2 \end{bmatrix}, \quad B^1_{\Delta p \Delta q} = \begin{bmatrix} 1 & 1 & 3 & 1 & 1 \\ 1 & 3 & 5 & 3 & 1 \\ 3 & 5 & 8 & 5 & 3 \\ 1 & 3 & 5 & 3 & 1 \\ 1 & 1 & 3 & 1 & 1 \end{bmatrix},$$
$$B^2_{\Delta p \Delta q} = \begin{bmatrix} 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & 6 & 2 & 1 \\ 2 & 6 & 16 & 6 & 2 \\ 1 & 2 & 6 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \end{bmatrix}, \quad B^3_{\Delta p \Delta q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & 0 & 0 \\ 0 & 8 & 32 & 8 & 0 \\ 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{7.35}$$

7.5. RM Filtering Technique

7.5.1. Some classical filters. It is easy to define the *R filter*, using the genetic equation for the mean value in a sliding window in the image [2, 21]:

$$\widehat{e}(i,j) = \frac{1}{(2K+1)^2} \sum_{m=-K}^{K} \sum_{n=-K}^{K} x(i-m, j-n), \tag{7.36}$$

where $\widehat{e}(i,j)$ is the estimation of the image e, $x(i,j)$ is the image corrupted by Gaussian noise, $(2K+1)^2$ is the filter window size with $i = 1, \ldots, M$, $j = 1, \ldots, N$, and MxN is the image size. Filter (7.36) is known as a standard mean filter or lineal standard filter [21].

The R filter can be found by changing the arithmetic mean by the rank operation. By applying equation (6.21) to (7.36), it is easy to obtain the standard median filter [21–25]

$$\widehat{e}_{\text{med}} = \text{med}\left\{q(i+m, j+n), \quad m,n = -K, \ldots, K\right\}, \tag{7.37}$$

where $\text{med}\{\cdot\}$ denotes the median of all pixels in the filter window of size $(2K+1)^2$.

Similarly, the Wilcoxon filter could be found from estimator (6.22) [2, 21] as

$$\widehat{e}_{Wil} = \text{med}\left\{\frac{q(i+m, j+n) + q\,(i+m_1, j+n_1)}{2}\right\}, \tag{7.38}$$

where $m \leqslant m_1$, $n \leqslant n_1$ and m, n, m_1, $n_1 = -K, \ldots, K$.

Median filter (7.37) has sufficient properties in applications of image processing when any *a priory* information is absent. On the contrary, *Wilcoxon* filter (7.38) can give a good estimation when the data distribution has a symmetric form. It is known [2] that its robust properties are not sufficient in the case of impulsive noise since it cannot suppress it sufficiently. We propose to modify this filter in order to increase its robustness [2, 22, 25, 26]. The standard technique in this case is the operation of *trimming* or *winsorization* [2, 22]. So, in the ordering of data in the sample X_j with $j = 1, \ldots, K$, the elements that have the rank number less than αK and more than $K - \alpha K$ should be eliminated. The volume of eliminated data is defined by the cut of «trimming» parameter α, varying from 0 to 0.5. Applying this concept to median filter (7.36) it is easy to write the α-trimmed mean (α-TM) filter [10]

$$\widehat{e}_{\alpha-TM}(i,j) = \frac{1}{L - 2\alpha L} \sum_{k=\alpha L}^{L-\alpha L} R_q(k), \tag{7.39}$$

where $L = (2K+1)^2$ and $R_q(k)$ represents the value of the pixel having the rank k among all elements in the filter window $q(i+m, j+n)$, $m, n = -K, \ldots, K$.

By applying sample censorization to Wilcoxon estimator (6.22) and rank operation to generic filter (7.36), one can find the *Wilcoxon α-TM filter* [25]

$$\widehat{e}_{W\alpha-TM}(i,j) = \underset{k \leqslant l}{\text{med}} \left\{ \frac{R_k + R_l}{2}, \quad k, l = \alpha L, \ldots, L - \alpha L \right\}, \tag{7.40}$$

where $L = (2K+1)^2$.

7.5.2. Examples of RM Filters. The design of the M and *RM* filters is based on genetic equation (7.36) and equations (6.32), (6.33), and (6.34). The standard$M$ filter (STM) [2, 22, 25] can be found by using the influence function $\psi(X)$ defined in (6.6) and applying the winsorization:

$$\widehat{e}_{STM}(i,j) = \frac{1}{2K+1} \sum_{m=-K}^{K} \sum_{n=-K}^{K} \widetilde{\psi}\left[g(i+m, j+n)\right], \tag{7.41}$$

where

$$\widetilde{\psi}\left(g(i,j)\right) = \begin{cases} \widehat{e}_{\text{med}}(i,j) - b, & g(i,j) < b, \\ q(i,j), & |g(i,j)| \leqslant b, \\ \widehat{e}_{\text{med}}(i,j) + b, & g(i,j) > b, \end{cases}$$

and $g(i+m, j+n) = q(i+m, j+n) - \widehat{e}_{\text{med}}(i,j)$.

Applying the *RM* estimator from (6.32) in equation (7.41), one can find the median standard filter. Employing *RM* estimator (6.33) again, we obtain the *Wilcoxon M* type filter (W-STM) [25]:

$$\begin{gathered} \widehat{e}_{W-STM}(i,j) = \text{med}\left\{ \frac{y_k + y_l}{2}, k \leqslant l \right\}, \\ \overrightarrow{y} = \left\{ \widetilde{\psi}\left[g(i+m, j+n)\right], m, n = -K, \ldots, K \right\}, \end{gathered} \tag{7.42}$$

where $k, l = -K, \ldots, K$, the vector $\overrightarrow{y}$ denotes the data intermediate sample, and $\widetilde{\psi}\left[g(i,j)\right]$ is the influence function from equation (7.42).

Applying the simple cut influence function (6.6) in the equation (6.31) and making some modifications in equation (7.41), we obtain the *cut M filter*:

$$\begin{aligned} \widehat{e}_{CM}(i,j) = \frac{1}{\sum \psi_{cut}\left[g(i+m, j+n)\right]} \times \\ \times \sum_{m=-K}^{K} \sum_{n=-K}^{K} \psi_{cut}\left[g(i+m, j+n)\right] \cdot q(i+m, j+n), \end{aligned} \tag{7.43}$$

where

$$\psi_{cut}\left[g(i,j)\right] = \begin{cases} 1, & g(i,j) = |q(i,j) - \text{med}\left\{q(i+m, j+n) \leqslant b\right\} \\ & \quad m, n = -K, \ldots, K, \\ 0, & \text{another case.} \end{cases}.$$

Using *RM* estimator (6.33) in equation (7.43), we obtain the *Wilcoxon cut filter* (WCM) [25]

$$\widehat{e}_{WCM}(i,j) = \text{med}\left\{ \frac{y_k + y_l}{2}, k \leqslant l \right\}. \tag{7.44}$$

Here,

$$\overrightarrow{y} = \{q(i+m, j+n) : |q(i+m, j+n) - \operatorname{med}\{q\,(i+m_1, j+n_1)\}| \leqslant b, \\ m, n, m_1, n_1 = -K, \ldots, K\},$$

where the vector $\overrightarrow{y}$ denotes a sample of the intermediate data.

Finally, applying *RM* estimator (6.32) to equation (7.44), we obtain the *median cut filter* (MCM) [28-30]

$$\widehat{e}_{MCM}(i,j) = \operatorname{med}\{\overrightarrow{y}\}, \tag{7.45}$$

where the vector $\overrightarrow{y}$ is defined as in the previous equation. The same results can be obtained if we use function (6.8) in (6.32) and make necessary substitutions in (7.36).

7.5.3. RM-KNN Type Filters. Robustness can be increased by different methods: censorization or others [22, 26, 28]. A well-known way to enhance the filtration quality by preservation of image details consists in the use of K sample elements with the values close to that of the central pixel in a sliding filtration window. In this case, the aforementioned filtering algorithm (*K-nearest neighbor*) can be used:

$$\widehat{e}_{KNN}(i,j) = \frac{1}{K} \sum_{m=-L}^{L} \sum_{n=-L}^{L} \psi\,(x(i+m, j+n))\, x(i+m, j+n), \tag{7.46}$$

where

$$\psi\,(x(i+m, j+n)) = \begin{cases} 1, & \text{if } x(i+m, j+n) \text{ is one of the K valures} \\ & \text{the most closed to x(i, j) in the window,} \\ 0, & \text{another case,} \end{cases}$$

and $m, n = -L, \ldots, L$.

In order to improve robustness of KNN standard filter (7.46), we substitute the calculated of arithmetic mean by *RM* algorithms (6.32) and (6.33) as the first step of estimation. The following two RM-KNN filters can be derived [29–31]:
Median type-M KNN filter (MM–KNN)

$$\widehat{e}_{MM-KNN}(i,j) = \operatorname{med}\{x_{KNN}(i+m, j+n)\}, \tag{7.47}$$

and *Wilcoxon type-M KNN filter* (WM–KNN).

$$\widehat{e}_{WM-KNN}(i,j) = \operatorname{med}\left\{\frac{x_{KNN}(i+m, j+n) + x_{KNN}\,(i+m_1, j+n_1)}{2}\right\}, \tag{7.48}$$

where $x_{KNN}(i+m, j+n)$ and $x_{KNN}(i+m_1, j+n_1)$ are the K pixels in the filter window that have the nearest values to the central pixel $x(i,j)$, and $m, n, m_1, n_1 = -$ $-L, \ldots, L$. It is easy to derive similar algorithms in the cases when other R-estimators are used: the *Mood* and *Ansari–Bradley–Siegel–Tukey* filters.

To improve the robustness of the *KNN* filter, near the edges of image objects, we should make some adjustment: first, in the number of the nearest pixels, which have to be chosen in accordance to the local deviation of sample data in window, and second, in the suppression of impulsive noise. To do this, an iterative procedure has been proposed.

The resulting filter, named lineal type-M KNN (LM–KNN) [30, 31], is defined as follows:

$$\widehat{e}^{(q)}_{LM-KNN}(i,j) = \frac{1}{K_s} \sum_{m=-L}^{L} \sum_{n=-L}^{L} \psi^{(q)}\,(x(i+m, j+n))\, x(i+m, j+n), \tag{7.49}$$

where

$$\psi^{(q)}\left(x(i+m,j+n)\right)=\begin{cases}1, & \text{if } x(i+m,j+n) \text{ is one of } K_s\\ & \text{valures closed to } \widehat{e}_{KNN}^{(q-1)}(i,j),\\ 0, & \text{another case,}\end{cases}$$

and

$$\begin{gathered}K_s=K_{\min}+a\cdot S\left(x(i,j)\right)\leqslant K_{\max},\\ S\left(x(i,j)\right)=\operatorname{med}\left\{|x(i,j)-x(i+m.j+n)|\right\}.\end{gathered}\tag{7.50}$$

The initial estimation is $\widehat{e}_{KNN}^{(0)}(i,j)=x(i,j)$; a controls the filter sensitivity for a better detection of edges; $K_{\min}$ is the minimum neighbor number for elimination of noise; $K_{\max}$ is the number for restoration of the edges and fine details; $S\left(x(i,j)\right)$ is the pulse detector based on the median of the differences between the central pixel and another pixels containing in the filter window. If the value of $S\left(x(i,j)\right)$ is small, the pixel can be recognized as clear of noise; in the opposite case, if $S\left(x(i,j)\right)$ is large, the pixel is recognized as contaminated by impulsive noise. The iterations should be ended when $\widehat{e}^{(q)}(i,j)=\widehat{e}^{(q-1)}(i,j)$. One can see that filter (7.53) is the extended form of the standard *KNN* filter.

Applying *RM* estimation (6.32) in (7.49), we can find the final version of the *RM–KNN* filter for preservation of details and impulse noise suppression. This filter is named the *Median type-M KNN* filter (*MM–KNN*) and determined as follows [31]:

$$\widehat{e}_{MMKNN}^{(q)}(i,j)=\operatorname{med}\left\{g^{(q)}(i+m,j+n)\right\},\tag{7.51}$$

where $g^{(q)}(i+m,j+n)$ are some K_{close} pixels weighted according to function $\psi(X)$ from (7.6) and having nearest values in the sliding filtering window to the estimate $\widehat{e}_{MMKNN}^{(q-1)}(i,j)$ in the previous step; the initial estimate is $\widehat{e}_{MMKNN}^{(0)}(i,j)=x(i,j)$; $x(i,j)$ is the original image pixel degraded by impulsive noise or the central pixel of the filtering window; $(2L+1)^2$ is the filtering window size with $m,n=-L,\ldots,L$; $\widehat{e}_{MMKNN}^{(q)}$ denotes the estimate in the iteration q; and q is the iteration index of actual iteration. The iterations end when the actual estimate $\widehat{e}_{MMKNN}^{(q)}$ equals the previous estimation $\widehat{e}_{MMKNN}^{(q-1)}(i,j)$. Usually, as it has been found in simulations, it takes 3 or 4 steps to satisfy this condition.

The value $K_{close}(i,j)$ is the actual number of the nearest neighboring pixels. It reflects the local data activity and the presence of impulsive noise [31]:

$$K_{close}(i,j)=\left[K_{\min}+a\cdot D_s\left(x\left(i,j\right)\right)\right]\leqslant K_{\max},\tag{7.52}$$

$$D_s\left(x(i,j)\right)=\frac{\operatorname{med}\left\{|x(i,j)-x(i+m,j+n)|\right\}}{MAD\left\{x(i,j)\right\}}+0.5\frac{MAD\left\{x(i,j)\right\}}{\operatorname{med}\left\{x(i+k,j+l)\right\}},\tag{7.53}$$

where a controls the filter sensitivity in order to have satisfactory capability of detecting fine details; $K_{\min}$ is minimum number of neighboring pixels to remove noise; $K_{\max}$ is maximum number of neighboring pixels for restoring edges and fine details; $D_S\left(x(i,j)\right)$ is the proposed pulse detector, which has a better capability of detection as compared to that given by (7.50); and MAD is the median of the absolute deviations of the median defined in Chapter 6: $MAD\left\{x(i,j)\right\}=\operatorname{med}\left\{|x(i+m,j+n)-\operatorname{med}\left\{x(i+k,j+l)\right\}|\right\}$.

Pulse detector $D_S\left(x(i,j)\right)$ (7.53) was proposed in accordance with the following reasoning: the MAD is the most robust estimator for standard deviation, usually used as a scale estimator and sometimes as a local estimator of the standard deviation of the signal. If the value of MAD is small, the pixel is not contaminated by noise. In the

opposite case, when the value MAD is large, the pixel is degraded by impulse noise. The actual number of the nearest neighboring pixels $K_{close}(i,j)$ (7.52) is determined with the only purpose to give satisfactory detail preservation for the filter, so it was proposed to take into account the minimum number of neighboring pixels $K_{\min}$ used in calculations to remove impulse noise. The operation $a \cdot D_s\left(x(i,j)\right)$ represents a quantity that allows one to identify whether the pixel is of the impulsive noise or is from details of an image.

In a similar matter, as in case of the MM–KNN filter (7.51), the *Wilcoxon M-type KNN* filter (WM–KNN) can be obtained from estimation *RM* (6.33) in (7.49) as follows [30, 31]:

$$\widehat{e}^{(q)}_{WMKNN}(i,j) = \operatorname{med}\left\{\frac{g^{(q)}(i+m,j+n)+g^{(q)}(i+m_1,j+n_1)}{2}\right\}, \tag{7.54}$$

where $g^{(q)}(i+m,j+n)$ and $g^{(q)}(i+m_1,j+n_1)$ are the sets of K_{close} pixels with the weight according to the influence function $\psi(X)$, nearest in their values to the estimate at the previous step $\widehat{e}^{(q-1)}_{WMKNN}(i,j)$; the initial estimate is $\widehat{e}^{(0)}_{WMKNN}(i,j) = x(i,j)$; $x(i,j)$ is the original image; $\widehat{e}^{(q)}_{WMKNN}$ denotes the estimate on iteration q; and q is the index of the current iteration. The iterations end when the current estimate $\widehat{e}^{(q)}_{WMKNN}$ equals the previous estimate $\widehat{e}^{(q-1)}_{WMKNN}(i,j)$; $K_{close}(i,j)$ is defined by (7.52); and $(2L+1)^2$ is the size of the filter with $m \leqslant m_1$, $n \leqslant n_1$ y $m,n,m_1,n_1 = -L,\ldots,L$.

MM-KNN filter (7.51) has a good capability in reduction of impulse noise and fine detail preservation, due to its robustness and the choice of the influence function, but it cannot suppress Gaussian multiplicative noise well when the variations are large. To cope with this problem, the *M-filter* has been proposed as a second-stage filter [32, 33].

7.5.4. Robust M Filter. The *M-filter* provides good fine detail preservation and is capable to remove spikes and suppress multiplicative noise [32, 33]. This filter uses the following influence function in the M estimator:

$$\psi(X) = \begin{cases} X, & \left|X-\theta^{(0)}\right| \leqslant b \cdot \operatorname{med}\{X\}, \\ 0, & \text{another case}, \end{cases} \tag{7.55}$$

where X is a data vector in a filter window, $\theta^{(0)} = \widehat{e}^{(q)}_{MMKNN}(i,j)$, and $\operatorname{med}\{X\}$ is the median of the pixels in the window.

The M type filter using the influence function defined above can be written as [32, 33]:

$$\widehat{e}_M(i,j) = \frac{\displaystyle\sum_{m=-L}^{L}\sum_{n=-L}^{L} x(i,j)\psi'\left\{x(i+m,j+n) - \widehat{e}^{(0)}(i,j)\right\}}{\displaystyle\sum_{k=-L}^{L}\sum_{l=-L}^{L} \psi'\left\{x(i+m,j+n) - \widehat{e}^{(0)}(i,j)\right\}}, \tag{7.56}$$

where

$$\psi'\left\{x(i+m,j+n) - \widehat{e}^{(0)}(i,j)\right\} = \begin{cases} 1, & \left|x(i+m,j+n) - \widehat{e}^{(0)}(i,j)\right| \leqslant \\ & \leqslant b \cdot \operatorname{med}\left\{x(i+m,j+n)\right\}, \\ 0, & \text{another also}, \end{cases}$$

is the influence function in the case when the simple cut function (6.8) is used. The influence function $\psi(X)$ from (6.8) should be changed in *M filter* (7.56) if other influence functions (6.7), (6.9)–(6.12) are used. The value $\operatorname{med}\{x(i+m,j+n)\}$ is the median of the pixels nearest to the central one in the window and b controls the suppression

of multiplicative noise. It has been found in simulations that the optimal value is 2. Also in (7.56), $(2L+1)^2$ is the filter window size with $m, n = -L, \ldots, L$ and $\widehat{e}^{(0)}(i,j) = \widehat{e}^{(q)}_{MMKNN}(i,j)$ is the initial estimation.

The *M filter* (7.56) has a good noise suppression performance in the case of multiplicative noise but, at the same time, detail preservation is satisfactory only when the noise level is not too high. Finally, we have proposed an adaptive scheme, similar to that used in the local statistics *Lee filter* [4, 6], combining two filters:

$$\widehat{e}_{LEE} = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_n^2} X^* + \left(1 - \frac{\sigma_x^2}{\sigma_x^2 + \sigma_n^2}\right) m, \tag{7.57}$$

where the following outputs of filter estimates $\widehat{e}_{MMKNN}$ and $\widehat{e}_M$ are used [32, 33]:

$$\widehat{e}_{RM-CAS}(i,j) = \widehat{e}_{MMKNN}(i,j) \cdot \widehat{\theta}_W(i,j) + \; + \left[1 - \widehat{\theta}_W(i,j)\right] \cdot \widehat{e}_M(i,j), \tag{7.58}$$

$$\widehat{\theta}_W(i,j) = 1 - \left(c \frac{\widehat{e}_M(i,j)}{\operatorname{med}\left\{|\widehat{e}_M(i,j) - x(i+m, j+n)|\right\}}\right)^2. \tag{7.59}$$

In (7.59), $\widehat{\theta}_W(i,j)$ is a robust estimator of local data activity; c controls the fine detail preservation, and $\widehat{e}_{RM-CAS}(i,j)$ is the output of the proposed two-stage filter.

The resulting image filter (7.58) is named the *RM cascade filter* and represents two filters connected in cascade to preserve fine details: MM-KNN filter (7.51) to remove impulsive noise and M filter (7.56) [33], which suppresses multiplicative noise. So, this filter is very similar to the well-known *Sigma filter* [6] for detail preservation with an exception that it uses local weighted adaptive data. The outputs of these filters are mixed in a manner similar to the *Lee filter* [4], which takes the relation of local data activity of the image to realize better preservation of fine details in the image.

In order to determine the noise suppression properties and compare the qualitative characteristics of various filters, namely, the 3x3 Cascaded RM-filters with simple cut, Hampel's three part redescending, Andrew's sine, Tukey biweight, Bernoulli influence functions, 3x3 normalized least-mean-squares L (*NLMS-L*) [10], 3x3 rank-order mean (*ROM*) [16], 5x5 *modified Frost* [20, 31], and 3x3 vector median rational hybrid (*VMRH*) [34] filters were simulated. The reason of choosing these filters was to compare the performance of the filters proposed above with that of various known filters and demonstrate their advantages.

The commonly used 256×256 grayscale test images «Airfield», «Bridge», «Mandrill», and «Lena» were corrupted by an impulse noise with 20% of spike occurrence mixed with multiplicative Gaussian noise with the variance of 0.1. The choice of these images is justified by different texture character of them. Thus, whereas the images «Bridge» and «Mandrill» have large areas similar to noise structure, the other images such as «Airfield» have geometrically restricted objects and «Lena» is a portrait type image. Therefore, a good performance of the proposed filters in filtering these images could justify the possibility to decrease noise influence and preserve fine details in other images with a similar structure.

The PSNR and MAE performances for all investigated filters are presented in Table 7.4, which shows that the Cascaded RM-filters are preferable because of better PSNR and MAE performances in comparison with the NLMS-L and ROM filters.

The performances of the filters are almost similar to those of the modified Frost and VMRH filters but with better detail preservation. Figure 7.1 presents the processed images for «Airfield» image, demonstrating better detail preservation attained by the proposed filters in comparison with other techniques.

Table 7.4. PSNR (in dB) and MAE values for different images degraded by impulsive noise with 20% of spike occurrence mixed with multiplicative noise having variance σ_ε^2 of 0.1 for different filters.

Nonlinear Filters	Airfield		Bridge		Mandrill	
	PSNR	MAE	PSNR	MAE	PSNR	MAE
3x3 NLMS-L	14.94	37.20	14.35	42.77	14.63	38.96
3x3 ROM	17.41	26.63	16.89	28.57	16.04	31.03
5x5 Modified FROST	20.03	18.92	19.21	21.60	18.24	24.53
3x3 VMRH	20.14	18.21	19.30	20.43	18.21	23.74
3x3 Cas. MM (Simple)	19.80	17.30	18.74	20.49	17.21	24.84
3x3 Cas. WM (Simple)	20.37	17.23	19.37	20.38	18.02	24.43
3x3 Cas. ABSTM (Simple)	20.44	16.99	19.53	20.05	18.10	24.10
3x3 Cas. MM (Hampel)	20.77	16.37	**19.86**	19.38	18.29	23.54
3x3 Cas. MM (Andrew)	20.86	16.24	19.85	**19.31**	18.31	23.54
3x3 Cas. MM (Tukey)	20.88	**16.18**	19.85	19.33	**18.32**	**23.51**
3x3 Cas. MM (Bernoulli)	**20.89**	**16.18**	19.85	19.32	18.31	23.52

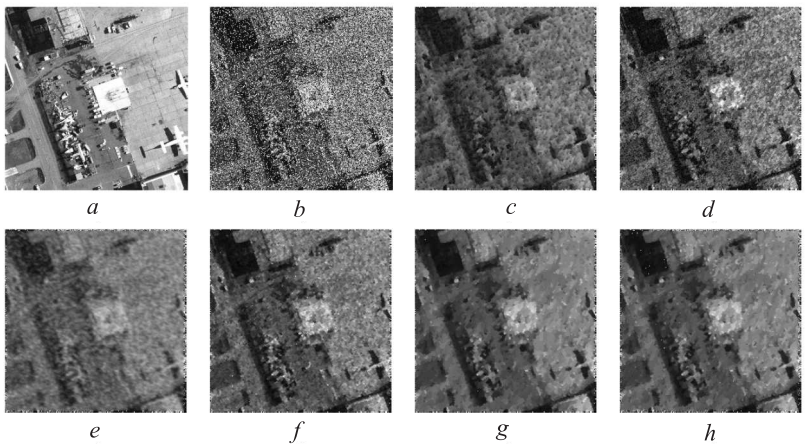


Fig. 7.1. Comparative results of suppression of impulsive and multiplicative noise in the «Airfield» image. a) Original image, b) Degraded image by mixture of impulsive noise with 20% of spike occurrence and multiplicative noise of variance 0.1, c) Restored image with the NLMS–L filter, d) Restored image with the ROM filter, e) Restored image with the Modified Frost filter, f) Restored image with the VMRH filter, g) Restored image with the Cascaded MM-filter (Bemoulli), h) Restored image with the Cascaded WM-filter (Bemoulli).

To investigate the performance of the proposed filters in the presence of noise with different intensity level, the 256×256 standard test grayscale image «Lena» was corrupted by impulsive noise with occurrence rate ranging from 1 to 20% mixed with multiplicative Gaussian noise having variances σ_ε^2 equal to 0.05, 0.1 and 0.25.

The PSNR and MAE performances for the comparative NLMS-L, ROM, modified Frost, VMRH filters and Cascaded RM- (MM- and WM-KNN) filter with Andrew's sine influence function are presented in Table 7.5.

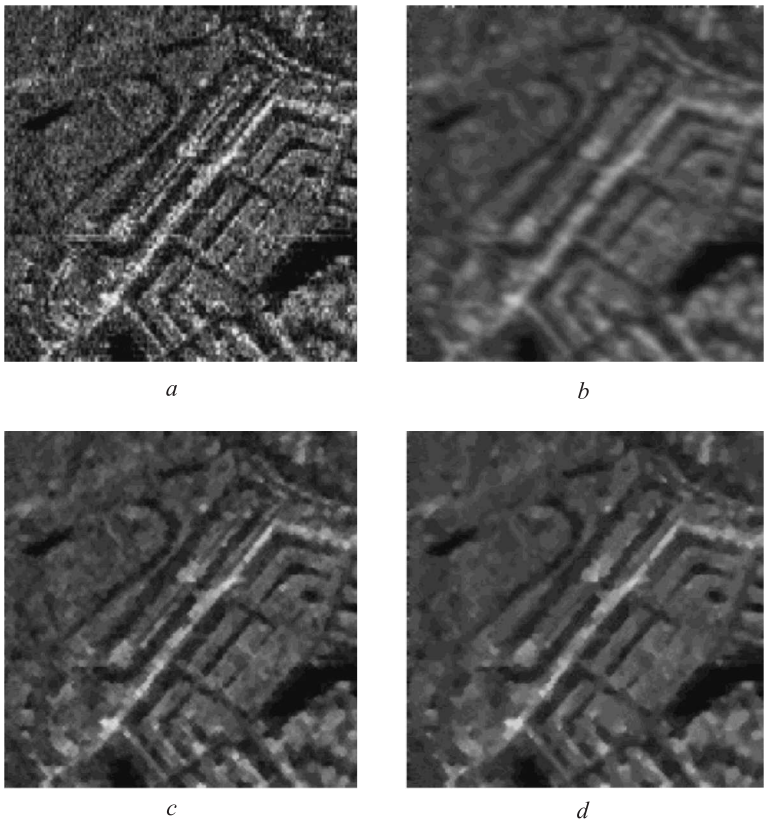


Fig. 7.2. Comparative results of despeckling for real-life SAR image. a) Original part of «Pentagon» image, resolution 1 m, source: Sandia National Lab., b) Despeckled image with the Modified Frost filter, c) Despeckled image with the VMRH filter, d) Despeckled image with the Cascaded ABSTM-filter (Bernoulli).

Here, we present only the use of one influence function in the proposed two filters, since the PSNR and MAE performances are very similar when other influence functions and other filters are used. In this test, we found out that the modified Frost filter is the best filter in comparison with the NLMS-L, ROM, and VMRH filters since it provides better PSNR and MAE performances when the impulsive noise percentage is 5% or less, approximately. The advantage of the proposed algorithm is that it does not use training data and parameters of the proposed filter can be treated as constants. It has been found out that the parameters of the proposed algorithm provide the optimal values of the PSNR and MAE criteria. Different test images have been degraded by mixture of impulsive and multiplicative noise.

Finally, we have standardized these parameters as the constants in order to realize the real-time implementation of the proposed filters [32, 33].

So, the proposed filter can suppress a mixture of complex noise and can preserve small-size details well as compared with other nonlinear filters described in literature, when impulsive noise percentages are 5% or more (Table 7.5).

To demonstrate the performances of the proposed filtering scheme, we applied it to filtering of the real-life SAR image, which naturally has multiplicative speckle noise.

Table 7.5. PSNR (in dB) and MAE values for «Lena» image degraded by impulsive noise with occurrence rate ranging from 1 to 20% mixed with multiplicative noise variance σ_ε^2 having of 0.05, 0.1 and 0.25 for different filters.

Mixed Noise		**Nonlinear Filters**					
multiplicative noise variance	impulsive noise percentage	3x3 NLMS-L		3x3 ROM		5x5 Modified FROST	
		PSNR	MAE	PSNR	MAE	PSNR	MAE
σ_ε=0.05	1	21.76	16.38	22.76	13.89	**24.56**	**10.37**
	5	21.43	16.89	22.51	14.43	23.56	11.90
	10	20.83	17.62	22.11	15.12	22.84	13.33
	20	19.60	19.73	21.03	17.08	21.06	17.03
σ_ε=0.1	1	20.41	18.86	21.40	16.52	**23.36**	**12.15**
	5	20.00	19.32	21.07	17.19	**22.71**	**13.29**
	10	19.55	20.06	20.71	17.98	21.92	14.95
	20	18.66	22.22	19.89	19.86	20.43	18.35
σ_ε=0.25	1	18.12	23.54	18.98	22.03	**20.04**	**17.02**
	5	17.96	23.95	18.85	22.40	19.65	18.06
	10	17.61	25.20	18.44	23.50	19.15	19.57
	20	16.74	27.69	17.79	25.48	18.30	21.92
		3x3 VMRH		3x3 Cas. MM (Andrew)		3x3 Cas. WM (Andrew)	
		PSNR	MAE	PSNR	MAE	PSNR	MAE
σ_ε=0.05	1	23.90	12.21	24.39	10.91	24.52	10.87
	5	23.45	12.84	23.92	11.53	**23.96**	**11.49**
	10	23.12	13.22	23.60	11.89	**23.67**	**11.86**
	20	21.89	15.24	22.50	13.33	**22.55**	**13.32**
σ_ε=0.1	1	21.47	16.23	22.49	13.66	22.55	13.60
	5	21.16	16.77	22.23	14.05	22.25	14.05
	10	20.78	17.63	21.95	14.55	**21.99**	**14.54**
	20	19.83	19.50	21.10	16.13	**21.12**	**16.11**
σ_ε=0.25	1	18.19	23.60	19.85	17.66	19.88	18.58
	5	18.09	24.09	**19.78**	**17.80**	**19.78**	18.81
	10	17.57	25.73	**19.38**	**19.13**	19.37	20.11
	20	16.75	28.26	18.70	**20.66**	**18.71**	21.64

The results of such a filtering are presented in Figure 7.2 for «Pentagon» image. As seen from the analysis of the zoom part of the filtering images, speckle noise can be efficiently suppressed, while the sharpness and fine feature are preserved well using the proposed Cascade RM-filters.

7.6. Model of Multichannel (Color) Image

7.6.1. RGB Model. In the case of multichannel images, in particular, color images, each pixel can be represented as a vector in 3D color space as shown in Fig. 7.3. For example, in *RGB*, the color image pixels are considered as vectors in the color cube, where the points marked with a cross «X» are in the intersection with the *Maxwell triangle* (the triangle drawn between the three primaries *R, G, B)* as is seen in Fig. 7.3. Angle θ represents the angles between two vectors U and V.

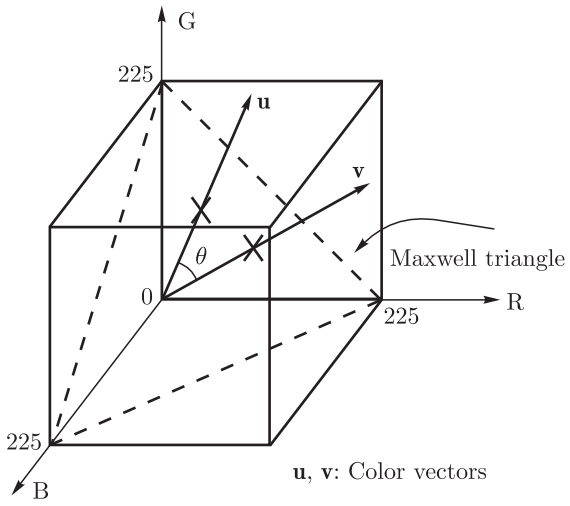


Fig. 7.3. Maxwell triangle.

Color images can be also modeled in other color spaces YIQ, HIS, HSV or L*a*b* [21, 35].

7.6.2. YIQ Model. This model is used in TV applications and is the linear transformation of the *RGB* model:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{7.60}$$

Usually, representation of the component (luminance) Y takes more bits than that of the components I and Q.

7.6.3. HSI (hue, saturation, intensity) Model. The component I presents intensity and the other two components: tone and saturation are based on human perception of color properties:

$$r = \frac{R}{(R+G+B)}, \quad g = \frac{G}{(R+G+B)}, \quad b = \frac{B}{(R+G+B)}, \tag{7.61}$$
$$r + g + b = 1.$$

It is easy to find the intensity as $I = \frac{1}{3}(R+G+B)$, and the equations for the other two components are

$$H = \cos^{-1}\left\{\frac{1/2[(R-G)+(R-B)]}{\left[(R-G)^2+(R-B)(G-B)\right]^{1/2}}\right\}, \qquad S = 1 - \frac{3}{(R+G+B)}\left[\min(R,G,B)\right]. \tag{7.62}$$

The conversion can be written as follows:
for $0° < H < 120°$ we have:

$$b = \frac{1}{3}(1-S), \quad r = \frac{1}{3}\left[1 + \frac{S\cos H}{\cos(60° - H)}\right], \quad g = 1-(r+b); \tag{7.62a}$$

for $GB(120° < H \leqslant 240°)$:

$$H = H - 120°,$$
$$r = \frac{1}{3}(1-S), \quad g = \frac{1}{3}\left[1 + \frac{S\cos H}{\cos(60° - H)}\right], \quad b = 1-(r+g); \tag{7.62b}$$

for $BR(240° < H \leqslant 360°)$:

$$H = H - 240°,$$
$$g = \frac{1}{3}(1-S), \quad b = \frac{1}{3}\left[1 + \frac{S\cos H}{\cos(60° - H)}\right], \quad r = 1-(g+b). \tag{7.62c}$$

The tone is the color attribute of pure, and saturation gives the estimates of the grade of dilution of pure color with white light. It is possible to separate the intensity component I from the color information of an image. This can be done in a similar manner as by the human eye. The model uses cylindrical coordinates.

The saturation is proportional to the radial distance, the tone (H) (Fig. 7.4) is the function of angle of coordinate, and intensity is the distance on the axis perpendicular to the polar coordinates [21].

7.6.4. HSV (hue, saturation, value) Model. This model is similar to the HSI one and it states that intensity is changed from black to white in one prism (Fig.7.5).

7.6.5. Models: L*u*v*and L*a*b*. The first model is represented using RGB space and reference point is white. This is equivalent to [1, 1, 1]. Lightness L* is defined as a cubic raise of luminance (Y):

The definition of L* is applied for one segment near black for $(Y/Y_{nis}) \leqslant 0.008856$ and is changed in the interval [0, 100].

$$L^* = \begin{cases} 116(Y/Y_n)^{1/3} - 16 & \text{if} \quad Y/Y_n > 0.008856, \\ 903.3(Y/Y_n)^{1/3} & \text{otherwise.} \end{cases} \tag{7.63}$$

To calculate $u*$ and $v*$, the following equations for intermediate parameters u', v', u'_n, v'_n is commonly used:

$$u' = \frac{4X}{X+15Y+3Z}, \quad v' = \frac{9Y}{X+15Y+3Z}, \qquad u'_n = \frac{4X_n}{X_n+15Y_n+3Z_n}, \quad v'_n = \frac{9Y_n}{X_n+15Y_n+3Z_n}. \tag{7.64}$$

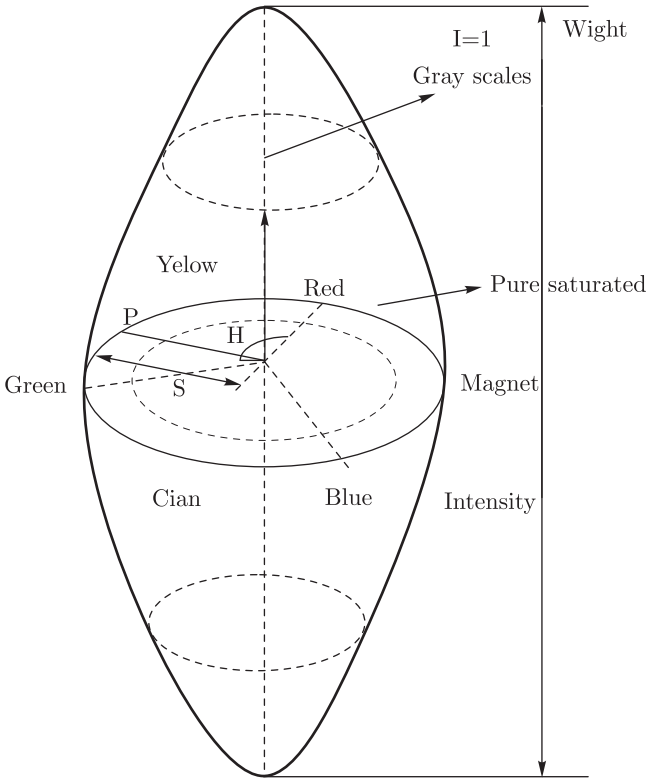


Fig. 7.4. HSI model.

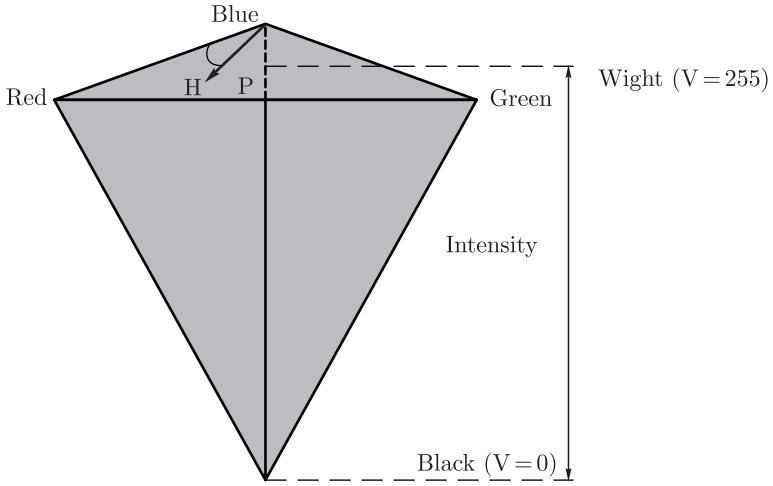


Fig. 7.5. HSV Model.

Finally, for $u*$ ad $v*$, we can write

$$\begin{cases} u* = 13L*(u' - u'_n), \\ v* = 13L*(v' - v'_n). \end{cases} \tag{7.65}$$

For $L^*a^*b^*$model, the equations are written as

$$\begin{cases} L^* = 116(Y/Y_n)^{1/3} - 16, \\ a^* = 500[(x/x_n)^{1/3} - (y/y_n)^{1/3}], \\ b^* = 200[(y/y_n)^{1/3} - (z/z_n)^{1/3}] \end{cases} \tag{7.66}$$

under the conditions x/x_n, y/y_n, $z/z_n > 0.01$.

7.6.6. False Color Model. The change of monochromatic images into color ones can be done according to gray levels, for example, as presented in Fig. 7.7, where the image is interpreted as a 2D function. Different colors are assigned there to different parallel planes. So, each pixel of a plane is represented by the same color.

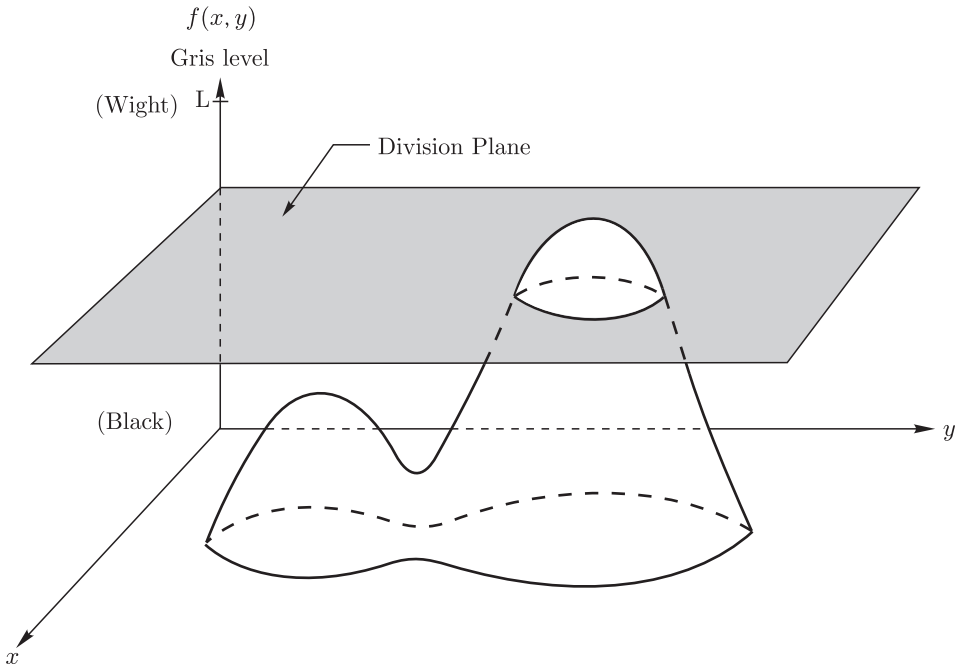


Fig. 7.6. Color codification.

7.7. Wavelet Functions in Multidimensional Signal Processing

7.7.1. Wavelet Analysis. The mathematical theory of the wavelet systems is discussed in Chapter 4 of this book. Here, before introducing the wavelet transform, we will review some of the concepts concerning such transforms. A transform can be thought of as a remapping of a signal that provides more information than the original.

The Fourier transform fits this definition quite well because the frequency information that it provides often leads to new insights in the original signal. However, the inability of the Fourier transform to describe both time and frequency characteristics of the waveform led to a number of different approaches. None of these approaches was able to solve completely the time–frequency problem. The wavelet transform can be used as another way to describe the properties of a waveform changing in time, but, in this case, the waveform is divided not into sections of time but segments of scale.

In the Fourier transform, the waveform is compared to the sine function, more precisely, a whole family of sine functions at harmonically related frequencies. This comparison is carried out by multiplying the waveform by the sinusoidal functions and averaging the product (using either integration in the continuous domain or summation in the discrete domain), i.e., $\mathrm{X}(\omega_m) = \int\limits_{-\infty}^{+\infty} x(t)e^{-j\omega_m t}dt$, which represents the continuous Fourier transform.

Almost any family of functions could be used to probe the characteristics of a waveform, but sinusoidal functions are particularly popular because of their unique frequency characteristics: they contain energy at only one specific frequency. Naturally, this feature makes them ideal for probing the frequency makeup of a waveform, for example, its frequency spectrum.

Other probing functions can be used, functions chosen to evaluate some particular behavior or characteristic of the waveform. If the probing function is of finite duration, it would be appropriate to translate, or slide, the function over the waveform, x(t), as is done in convolution and the *short-term Fourier transform* (STFT), given by the following equation:

$$STFT(t,f) = \int\limits_{-\infty}^{+\infty} x(\tau)\left(w(t-\tau)\,e^{-2j\pi f\tau}\right)d\tau. \tag{7.67}$$

Here, f is the frequency and also serves as an indication of family member, and w(t-τ) is some sliding window function, where t acts to translate the window over x. More generally, a translated probing function can be written as

$$X(t,m) = \int\limits_{-\infty}^{+\infty} x(\tau)\,f_m(t-\tau)\,d\tau. \tag{7.68}$$

Here, $f_m(t)$ is some family of functions and m specifies the family number.

If the family of functions $f_m(t)$ is sufficiently large, then it should be able to represent all aspects of the waveform $x(t)$. This would then allow $x(t)$ to be reconstructed from $X(t,m)$ making this transform bilateral.

Often the family of basis functions is so large that $X(t,\ m)$ forms a redundant set of descriptions, more than sufficient to recover $x(t)$. This redundancy can sometimes be useful, serving to reduce noise or acting as a control, but may be simply unnecessary. Note that while the Fourier transform is not redundant, most transforms represented by (7.68) (including the STFT and all the distributions) would be, since they map a variable of one dimension (t) into a variable of two dimensions (t, m).

7.7.2. Wavelet Transform. The Wavelet transform introduces an intriguing twist to the basic concept defined by equation (7.68). In wavelet analysis, a variety of different probing functions may be used, but the family always consists of enlarged or compressed

versions of the basic function, as well as translations. This concept leads to the defining equation for *continuous wavelet transform* (CWT) [35, 36]:

$$W(a,b) = \int_{-\infty}^{+\infty} x(t) \frac{1}{\sqrt{|a|}} \psi * \left(\frac{t-b}{a} \right) dt, \tag{7.69}$$

where b acts to translate the function across $x(t)$ just as t does in the equations above, and the variable a acts to vary the time scale of the probing function ψ. If value a is greater than one, the wavelet function ψ is stretched along the time axis, and if it is less than one (but still positive) it contacts the function. Negative values of a simply flip the probing function on the time axis. While the probing function ψ could be any of a number of different functions, it always takes on an oscillatory form, hence the term «wavelet». If b=0 and a=1, then the wavelet is in its natural form, which is termed the *mother wavelet*; that is, $\psi_{1,0}(t) \equiv \psi(t)$. A mother wavelet is shown in Fig.7.7 along with some of its family members produced by dilation and contraction. The wavelet shown is the popular *Morlet wavelet*, named after a pioneer of wavelet analysis, and is defined by the equation:

$$\psi(t) = e^{-t^2} \cos\left(\pi \sqrt{\frac{2}{\ln 2}}\, t \right). \tag{7.70}$$

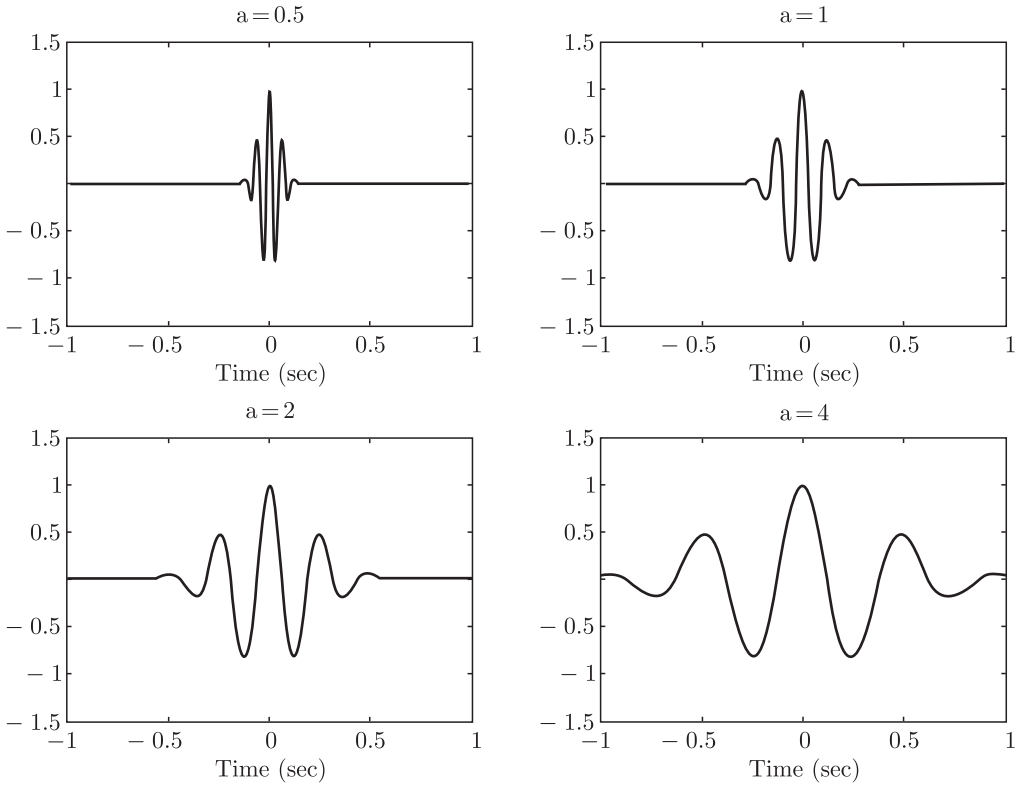


Fig. 7.7. A mother wavelet (a = 1) with two dilations (a = 2, 4) and one contraction (a = 0.5).

The wavelet coefficients $W(a, b)$ describe the correlation between the waveform and the wavelet at various translations and scales: the similarity between the waveform and the wavelet at a given combination of scale and position, a and b. Stated another way, the coefficients provide the amplitudes of a series of wavelets over a range of scales and translations, that would need to be added together to reconstruct the original signal. From this perspective, wavelet analysis can be thought of as a search over the waveform of interest for activity that most clearly approximates the shape of the wavelet.

This search is carried out over a range of wavelet sizes: the time span of the wavelet varies although its shape remains the same. Since the net area of a wavelet is always zero by design, a waveform that is constant over the length of the wavelet would give rise to zero coefficients. Wavelet coefficients respond to changes in the waveform, more strongly to changes on the same scale as the wavelet, and most strongly, to changes that resemble the wavelet. Although a redundant transformation, it is often easier to analyze or recognize patterns using the CWT.

If the wavelet function $\psi(t)$ is appropriately chosen, then it is possible to reconstruct the original waveform from the wavelet coefficients just as in the Fourier transform. Since the CWT decomposes the waveform into coefficients of two variables, a and b, a double summation in discrete case (or integration in continuous case) is required to recover the original signal from the coefficients [37, 38]:

$$x(t) = \frac{1}{C} \int_{a=-\infty}^{+\infty} \int_{b=-\infty}^{+\infty} W(a,b)\psi_{a,b}(t)da\,db, \tag{7.71}$$

where $C = \int_{-\infty}^{+\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega$ and $0 < C < -\infty$ (so called *admissibility condition*).

In fact, reconstruction of the original waveform is rarely performed using the CWT coefficients because of its redundancy. When recovery of the original waveform is desired, the more parsimonious discrete wavelet transform is used

Wavelet Time-Frequency Characteristics.

Wavelets, such as that shown in Fig. 7.7 do not exist at a specific time or a specific frequency. In fact, wavelets provide a compromise in the battle between time and frequency localization: they are well localized in both time and frequency but not precisely localized in either. A measure of the time range of a specific wavelet, Δt_ψ, can be specified by the square root of the second moment of a given wavelet about its time center:

$$\Delta t_\psi = \sqrt{\int_{-\infty}^{+\infty} (t - t_0)^2 |\psi(t/a)|^2\, dt \Big/ \int_{-\infty}^{+\infty} |\psi(t/a)|^2\, dt}\,,$$

where t_0 is the center time, or the first moment of the wavelet, and is given by

$$t_0 = \int_{-\infty}^{+\infty} t\,|\psi(t/a)|^2\, dt \Big/ \int_{-\infty}^{+\infty} |\psi(t/a)|^2\, dt.$$

Similarly the frequency range, $\Delta\omega_\psi$ is given by

$$\Delta\omega_\psi = \sqrt{\frac{\int_{-\infty}^{+\infty} (\omega - \omega_0)^2 |\Psi(\omega)|^2 \, d\omega}{\int_{-\infty}^{+\infty} |\Psi(\omega)|^2 \, d\omega}},$$

where $\Psi(\omega)$ is the frequency domain representation of $\psi(t/a)$, and ω_0 is the center frequency of $\Psi(\omega)$. The center frequency is given by the equation

$$\omega_0 = \frac{\int_{-\infty}^{+\infty} \omega \, |\Psi(\omega)|^2 \, d\omega}{\int_{-\infty}^{+\infty} |\Psi(\omega)|^2 \, d\omega}.$$

The time and frequency ranges of a given family can be obtained from the *mother wavelet*. Dilation by the variable a changes the time range simply by multiplying Δt_ψ, by a. Accordingly, the time range of $\psi_{a,0}$ is defined as $\Delta t_\psi(a) = |a| \Delta t_\psi$. The inverse relationship between time and frequency is shown in Fig.7.8, which was obtained for the *Mexican hat wavelet*. This wavelet is given by equation:

$$\psi(t) = (1 - 2t^2) e^{-t^2}. \tag{7.72}$$

The frequency range, or bandwidth, would be the range of the mother Wavelet divided by a: $\Delta\omega_\psi(a) = \Delta\omega_\psi / |a|$. If we multiply the frequency range by the time range, the a's cancel and we are left with a constant that is the product of the constants produced by Δt_ψ and $\Delta\omega_\psi$, given by the following equation:

$$\Delta\omega_\psi(a) \cdot \Delta t_\psi(a) = \Delta\omega_\psi \Delta t_\psi = \text{constant}_\psi. \tag{7.73}$$

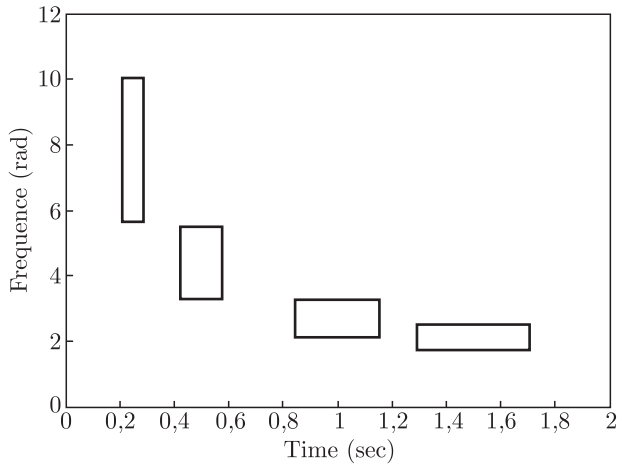


Fig. 7.8. Time-frequency boundaries of the *Mexican hat wavelet* for various values of a. The area of each of these boxes is constant.

Equation (7.73) shows that the product of the ranges is invariant to dilation (changes in the variable b do alter either the time or frequency resolution; hence, both time and frequency resolution, as well as their product, are independent of the value of b) and that the ranges are inversely related; increasing the frequency range, $\Delta\omega_\psi(a)$ decreases the time range, $\Delta t_\psi(a)$. These ranges correlate to the time and frequency resolution of CWT. Decreasing the wavelet time range (by decreasing a) provides a more accurate assessment of time characteristics (the ability to separate the close events in time) at the expense of frequency resolution, and vice versa.

Since the time and frequency resolutions are inversely related, the CWT can provide better frequency resolution when a is large and the length of the wavelet (and its effective time window) is long. Conversely, when a is small, the wavelet is short and the time resolution is maximum, but the wavelet only responds to high frequency components. Since a is variable, there is a built-in trade-off between time and frequency resolution, which is the key to the CWT and makes it well suited in signal analysis with rapidly varying high frequency components superimposed on slowly varying low frequency components.

The CWT has one serious problem: it is highly redundant. The CWT provides an oversampling of the original waveform: many more coefficients are generated than are actually needed to uniquely specify the signal. This redundancy is usually not a problem in analysis applications, such as described above, but will be costly if the application calls for recovery of the original signal. For recovery, all of the coefficients will be required and the computational effort could be excessive. In applications that require bilateral transformations, we would prefer a transform that produces the minimum number of coefficients required to recover accurately the original signal. The *discrete wavelet transform* (DWT) achieves this parsimony by restricting the variation in translation and scale, usually to powers of 2. When the scale is changed in powers of 2, the discrete wavelet transform is sometimes termed the *dyadic wavelet transform*. The DWT may still require redundancy to produce a bilateral transform unless the wavelet is carefully chosen such that it leads to an orthogonal family. In this case, the DWT will produce a non-redundant, bilateral transform.

The basic analytical expressions for the DWT will be presented here; however, the transform is easier to understand and easier to implement using filter banks [38–41]. The DWT is often introduced in terms of its recovery transform

$$x(t) = \sum_{k=-\infty}^{\infty} \sum_{\ell=-\infty}^{\infty} d(k,\ell) 2^{-k/2} \psi(2^{-k}t - \ell). \tag{7.74}$$

Here, k is related to a as $a = 2k$; b is related to ℓ as $b = 2^k\ell$; and $d(k,\ell)$ is a sampling of $W(a,b)$ at discrete points k and ℓ.

In the DWT, the *scaling function* is introduced, i.e., a function that facilitates computation of the DWT. To implement the DWT efficiently, the finest resolution is computed first. The computation then proceeds to coarser resolutions, but rather than start over on the original waveform, the computation uses a smoothed version of the fine resolution waveform. This smoothed version is obtained with the help of the scaling function. In fact, the scaling function is sometimes referred to as the *smoothing function*. The definition of the scaling function uses a *dilation* or a *two-scale difference equation*:

$$\varphi(t) = \sum_{n=-\infty}^{\infty} \sqrt{2}\, c(n) \varphi(2t - n). \tag{7.75}$$

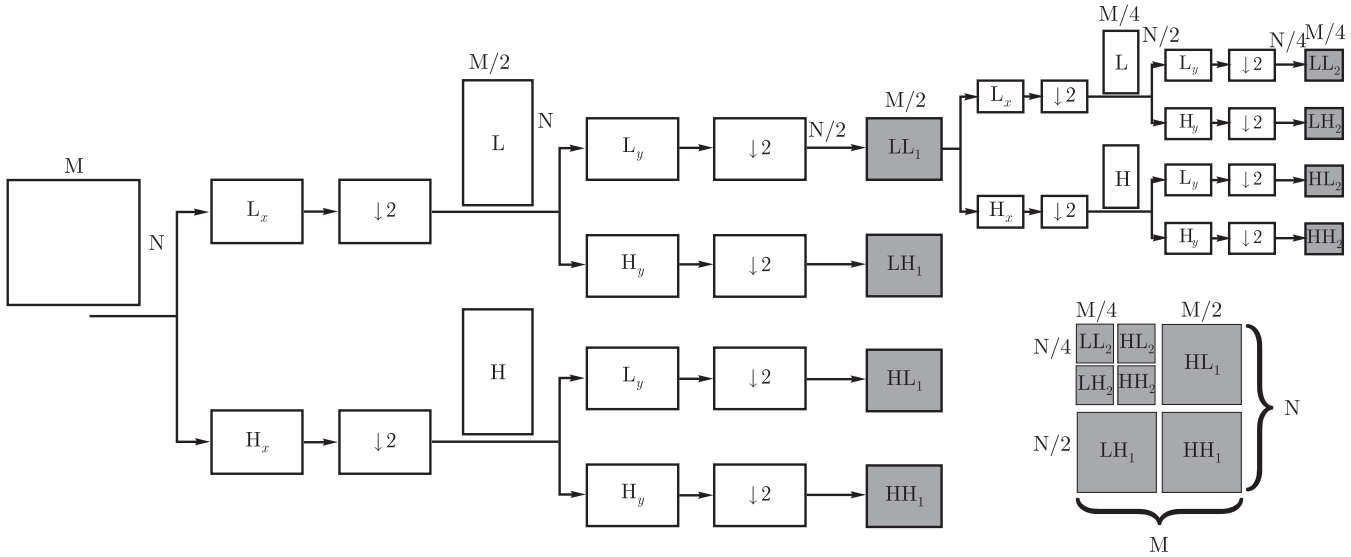


Fig. 7.9. Structure of the analysis filter bank for 2-D image with two levels of decomposition.

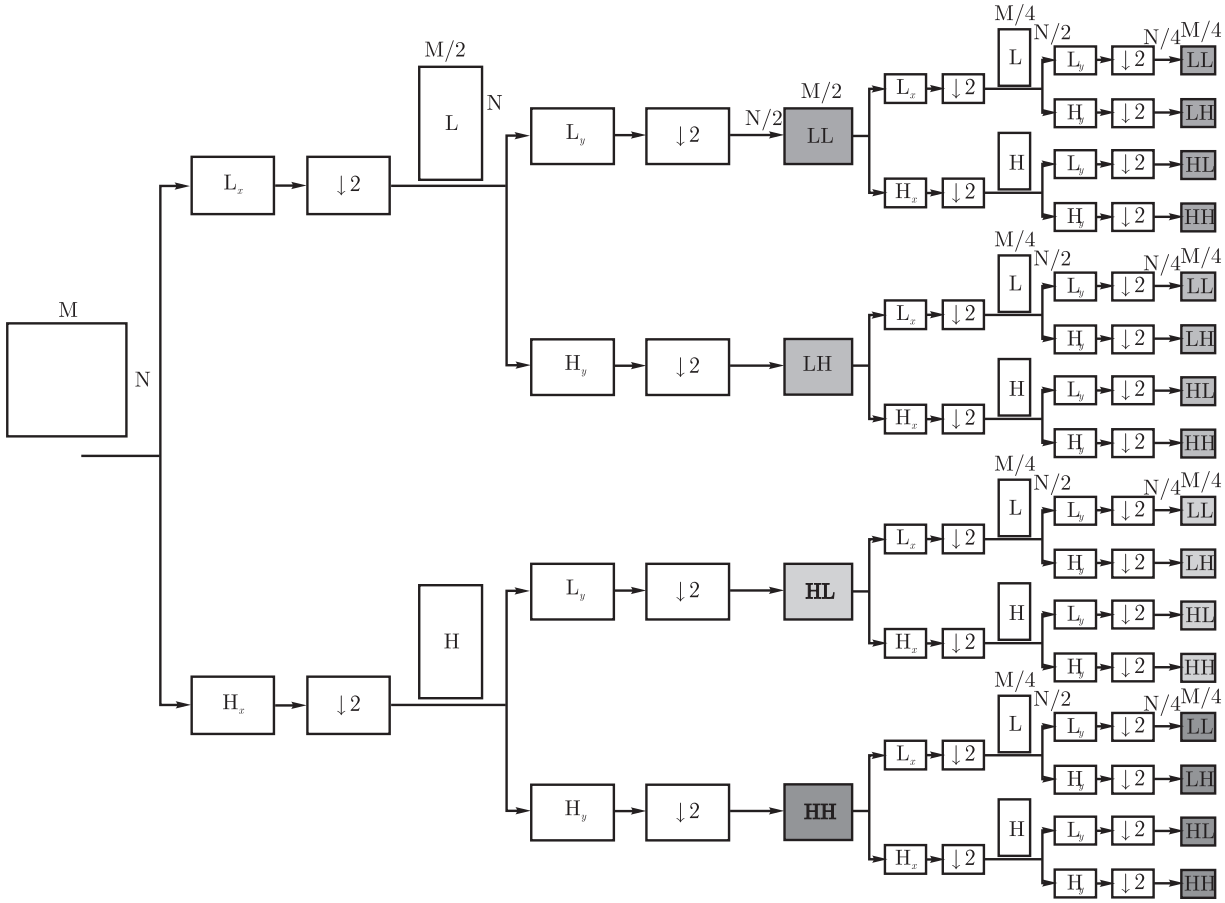


Fig. 7.10. Structure of the decomposition filters using Wavelets Packets for 2-D image with two levels of decomposition.

Where $c(n)$ are the series of scalars that define the specific scaling function. This equation involves two time scales (t and $2t$) and can be quite difficult to solve.

In the DWT, the wavelet itself can be defined from the scaling function [40]:

$$\psi(t) = \sum_{n=-\infty}^{\infty} \sqrt{2}\, d(n)\varphi(2t - n). \tag{7.76}$$

where $d(n)$ are the series of scalars that are related to the waveform $x(t)$ (7.74) and that define the discrete wavelet in terms of the scaling function. While the DWT can be implemented using the above equations, it is usually implemented using filter bank techniques.

For most signal and image processing applications, DWT-based analysis is best described in terms of filter banks. The use of a group of filters to divide up a signal into various spectral components is termed subband coding. The most used implementation of the DWT for 2-D signal applies only two filters for rows and columns, as in the filter bank, which is shown in Fig. 7.9.

The DWT can also be applied to construct useful descriptors of a waveform. Since the DWT is a bilateral transform, all of the information in the original waveform must be contained in the subband signals. These subband signals, or some aspect of the subband signals, such as their energy over a given time period, could provide a succinct description of some important aspect of the original signal.

In the decompositions described above, only the low-pass filter subband signals were sent on for further decomposition. This decomposition structure is also known as a logarithmic tree.

However, other decomposition structures are valid, including the complete or balanced tree structure shown in Fig. 7.10. In this decomposition scheme, both high-pass and low-pass subbands are further decomposed into high-pass and low-pass subbands up till the terminal signals. Other, more general, tree structures are possible, where a decision on further decomposition (whether or not to split a subband signal) depends on the activity of a given subband. The scaling functions and wavelets associated with such general tree structures are known as wavelet packets.

References

1. *Pitas, A.N. and Venetsanopoulos, A.N.* Nonlinear Digital Filters: Principles and Applications, Kluwer Academic Publisher, 1990.
2. *Astola, J. and Kuosmanen, P.* Fundamentals of Nonlinear Digital Filtering, CRC Press, Boca Raton-New York, 1997.
3. *Bednar, J.B. and Watt, T.L.* Alpha-trimmed means and their relationship to median filters. IEEE Trans. Acoust., Speech, and Signal Processing, 1984, vol. ASSP-32, pp. 145–153.
4. *Lee, Y.H. and Kassam, S.A.* Generalized median filtering and related nonlinear filtering techniques. IEEE Trans. Acoust., Speech, and Signal Processing, 1985, vol. ASSP-33, pp. 672–683.
5. *Davis, L.S. and Rosenfeld, A.* Noise cleaning by iterated local averaging. IEEE Trans. Syst. Man Cybern., 1978, vol. SMC-8, pp. 705–710.
6. *Lee, J.S.* Digital image smoothing and the sigma filter. Computer Vision, Graphics, and Image Processing, 1983., vol. 24, pp. 255–269.
7. *Longbotham, H.G. and Bovik, A.C.* Theory of order statistic filters and the relationships to linear FIR filters. IEEE Trans. Acoust., Speech and Signal Processing, 1989, vol. ASSP-37, No. 2, pp.275-287.

8. *Kassam, S.A. and Peterson S.R.* Nonlinear finite moving window filters for signal restoration. Proc. IEEE Pacific RIM Conf. on Comm. Computer, and Signal Processing, Victoria, Canada, 17–20, June 1987.
9. *Palmieri, F. and Boncelet, C.G.* L*l*–filters — a new class of order statistic filters. IEEE Trans. Acoust., Speech, Signal Processing, 1989, vol. 37, pp. 691–701.
10. *Kotropoulos, C. and Pitas, I.*, Adaptive LMS *L*-filters for noise suppression in images. IEEE Trans. Image Process. 1996, vol. 5, no. 12, pp. 1596–1609.
11. *Brownrigg, D.R.K.* Generation of representative members of an RrSst weighted median filter class. IEE Proc. Part. F, Commun., Radar, Signal Process., 1986, vol. 133, pp. 445–448.
12. *Nieweglowsky, J., Gabbouj, M., and Neuvo Y.* Weighted medians — positive Boolean function conversion. *Signal Processing*, 1993. vol. 34, pp. 149–162.
13. *Narenda, P.M.* A separable median filter for image noise smoothing. IEEE Trans. Pattern Anal. Mach., Intell., 1981, vol. PAMI-3, pp. 20–29.
14. *Ko, S.J. and Lee, Y.H.* Center Weighted Median Filters and their applications to image enhancement. IEEE Trans. Circuits and Systems, 1991, vol. 38, pp. 984–993.
15. *Hardie, R.C. and Boncelet, C.G.* LUM filters: a class of rank order based filters for smoothing and sharpening. IEEE Trans. Signal Processing, 1993, vol. 41, pp. 1061–1076.
16. *Abreu, E., Lightstone, M., Mitra, S.K., and Arakawa, K.* A new efficient approach for the removal of impulse noise from highly corrupted images. IEEE Trans. Image Process. 5(6), 1996, p. 1025.
17. *Astola, J., Haavisto, P., and Neuvo, Y.* Vector median filter. Proc. IEEE, 1990, vol. 78(4), pp. 678–689.
18. *Lee, J.S.* Digital image enhancement and noise filtering by use of local statistic. IEEE Trans. Pattern Anal. Mach. Intell., 1980, vol. PAMI-2, pp. 165-168.
19. *Kuan, D.T., Sawchuk, A.A., Strand T.C., and Chavel P.* Adaptive noise smoothing filter for images with signal-dependent noise. IEEE Trans. Pattern Anal. Mach. Intell., 1985, vol. PAMI-7, pp. 165–177.
20. *Frost, V.S, Stiles, J.A., Shanmugam, K.S., Holtzman, J.C., and Smith, S.A.* Adaptive filter for smoothing noisy radar images. Proc. of the IEEE, 69(1), 1981, pp. 133–135,
21. *Bovik, A.* Handbook of Image and Video Processing, Acadamic Press, San Diego CA, 2000.
22. *Huber, P.J.* Robust Statistics, Wiley, New York, 1981.
23. *Brownrigg, D.R.K.* The Weighted Median Filter. Commun. Assoc. Comput. Mach, 1984, vol. 27, pp. 887–818.
24. *Yin L., Yang R, Gabbouj, M., and Neuvo, Y.* Weighted median filters: a tutorial. IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, 1996, vol. 43, pp. 157–192, March.
25. *Ponomaryov, V.I. and Pogrebnyak, O.B.* Novel robust RM filters for radar image preliminary processing. Journal of Electronic Imaging, October 1999, vol. 8, no. 4. pp. 467–477.
26. *Hampel, F.R., Ronchetti, E.M, Rouseew, P.J., and Stahel, W.A.* Robust Statistics: The approach based on influence function, Wiley, New York, 1986.
27. *Hodges, J.L. and Lehmann, E.L.* Estimates of location based on rank test. Ann. Math. Statist., 1963, vol. 34, pp. 598–611.
28. *Volosyuk, V.K., Kravchenko, V.F., and Ponomaryov, V.I.* Mathematical Methods Physical Processes Modeling in Remote Sensing Problems of the Earth. Uspekji sovremennoi electroniki, 2000, no 12, pp. 3–74 (in Russian).
29. *Kravchenko, V.F., Ponomaryov, V.I., and Pustovoit V.I.* Algorithms of Robust Image Filtering with the Preservation of Fine Details in the Presence of Noise. Doklady Physics, «Nauka/Inerperiodica». N.Y. Sept. 2001, vol. 46, Is. 9, pp. 642–646.
30. *Kravchenko, V.F., Ponomaryov, V.I., Pogrebnyak, A.B., Pustovoit V.I., and Niño-de-Rivera L.* Robust Image processing filters preserving small features and rejecting impulse noise. Journal of Communications Technology and Electronics, April 2001, vol. 46, no. 4, pp. 450–455.

31. *Gallegos-Funes, F.J., Ponomaryov, V.I., Sadovnychiy S., and Nino-de-Rivera L.* Median M-type K-nearest neighbour (MM-KNN) filter to remove impulse noise from corrupted images. Electronics Letters, 2002, vol. 38, no. 15, pp. 786–787.
32. *Kravchenko, V.F., Ponomaryov, V.I., Pustovoit, V.I., Nino-De-Rivera, L., and Gallegos Funes F.* Robust Image Filtration with Fine Detail Preservation in presence of Multiplicative and Impulsive Noise. Doklady Physics, Edit. «Nauka/Inerperiodica». N.Y. 2001, vol. 46, no. 2, pp. 97–102.
33. *Kravchenko, V.F., Ponomaryov, V.I., Pustovoit, V.I., and. Nino-de-Rivera, L.* New Robust Nonlinear Image-Filtering Algorithms Preserving Small Features in the Presence of Noise. Journal of Communications Technology and Electronics, 2002, vol. 47, no. 1, pp. 67–74.
34. *Khriji, L. and Gabbouj, M.* Vector median-rational hybrid filters for multichannel image processing. IEEE Signal Processing Letters, 1999, vol. 6, no. 7, pp. 186–190.
35. *Plataniotis, K.N. and Venetsanopoulos, A.N.* Color Image Processing and Applications, Berlin, Springer Verlag, 2000.
36. *Jan, J.* Medical Image Processing, Reconstruction and Restoration: Concepts and Methods — N.Y.: Taylor & Francis, CRC Press, 2006.
37. *Iyengar, S.S., Cho, E.C., and Phoha, V.V.* Foundations of Wavelet Networks and Applications. Chapman & Hall/CRC, 2002.
38. *Mallat, S.A.* Theory for multiresolution signal decomposition: the wavelet representation. IEEE Anal Pattern. and Machine Intell., 1989, vol. 11, no. 7, pp. 674–693.
39. *Jähne, B.* Practical Handbook on Image Processing for Scientific and Technical Applications, Second Ed., CRC Press, 2004.
40. *Semmlow, J.L.* Biosignal and Biomedical Image Processing: MATLAB-Based Applications. Signal processing and Communications Series, Ed. Marcel Dekker, 2004.
41. *Misiti, M., Misiti, Y., and Oppenheim, G.* Wavelet toolbox User's Guide, N.Y.USA: Matlab, 1999.

Chapter 8

VECTOR ORDER STATISTICS IN MULTICHANNEL AND VIDEO PROCESSING

8.1. Vector Order Statistics

Different applications of multichannel image processing, such as multisensor remote sensing, color imaging, robot control, etc. can be sufficiently realized using concepts of robust vector order statistics. The final outputs of such rank procedures usually depend on the type of data that can be, for example, multichannel satellite sensor's images, or color channels ones. So, a special function Δ should be selected to evaluate the norm or distance between two vectors y_i and y_j that represent two images. Different techniques can be used, operated on the magnitude space, angular domain, and, finally, combined in usual representation or more complex fuzzy language ones.

The vector ordering of each multichannel sample or vector y, $i = 1, 2, \ldots, N$, should be reduced to a scalar representative that is obtained using the aggregated distances or similarities as follows [1–3]:

$$\Delta(y_i) = \sum_{j=1}^{N} d(y_i, y_j), \tag{8.1}$$

where $d(.)$ denotes the norm applied or dissimilarity measure. Usually, the Minkowski metrics (L_q) for p-channel signal y is used:

$$d_M(y_i, y_j) = \left(\sum_{k=1}^{p} | \, (y_i^k - y_j^k) \, |^q \right)^{1/q}.$$

Special cases of such a norm are: City-Block (norm L_1)

$$d_{L_1}^{(} y_i, y_j) = \sum_{k=1}^{p} |(y_i^k - y_j^k|$$

and Euclidean (norm L_2)

$$d_{L_2}(y_i, y_j) = \left(\sum_{k=1}^{p} | \, (y_i^k - y_j^k) \, |^2 \right)^{1/2}.$$

If it is necessary to realize order operation for vectors y_i located inside the supporting window, cube, or more general sample group. The scalar measures $\Delta(y_i)$ should be ranked in the order of its values and thus burn a novel ordered sample of $\Delta(y_i)$ and initial vectors y_i [2, 3]:

$$\Delta_{(1)} \leqslant \Delta_{(2)} \leqslant \ldots \leqslant \Delta_{(N)}, \tag{8.2}$$

$$y_{(1)}(\Delta_{(1)}) \leqslant y_{(2)}(\Delta_{(2)}^{)} \leqslant \ldots \leqslant y_{(N)}(\Delta_{(N)}). \tag{8.3}$$

Finally, the ordered vectors represent a one-dimensional sample and have a one-to-one correspondence with the original vector sample in the selected window, cube or more general sample group.

The output of ranking is determined by the type of the initial sample and the function $\Delta(y_i)$, which can be based on magnitude, directional, or fuzzy membership or maybe more general ones. Figure 8.1 illustrates the ordered vectors in the population of five vectors.

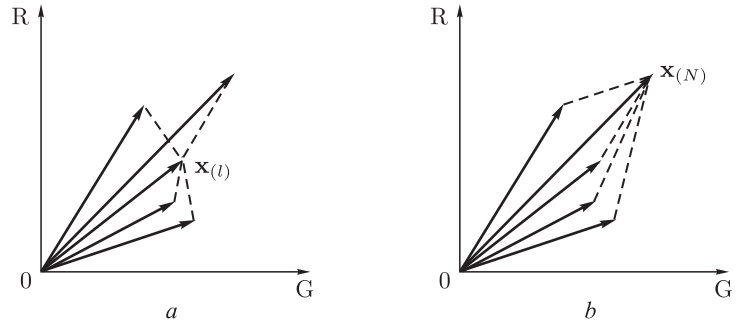


Fig. 8.1. Extreme vector order statistics expressed in the population of five vectors. (*a*) Lowest vector order statistic $x_{(1)}$. (*b*) Uppermost vector order statistic $x_{(N)}$.

8.1.1. Vector Median Ordering and Filtering. In this case, multichannel sample y_i is associated with the norm

$$\Delta L_i = \sum_{j=1}^{N} \|y_i - y_j\|_q = \sum_{j=1}^{N} \left(\sum_{m=1}^{K} |y_{ik} - y_{jk}|^q \right)^{\frac{1}{q}}, \tag{8.4}$$

where $\|y_i - y_j\|_q$ is the distance between two $K-$channel samples in the Minkowski norm in the particular cases: the city-block norm ($q = 1$), the Euclidean norm ($q = 2$), and the chess-board one ($q = \infty$). The sample associated with the minimum vector norm ΔL_1 defines the output of the *Vector Median Filter,* which minimizes the distance to all other vectors inside the sliding window, cube, or another group of samples, i.e., $\Delta L_1 \leqslant \Delta L_2 \leqslant, \ldots, \leqslant \Delta L_N$.

The weighted VMF (*WVMF*) is the vector Y_{WVM} with the components $\{y_1, y_2, \ldots, y_N\}$ satisfying the equation [1, 2, 4]

$$\sum_{i=1}^{N} w_i \|a_i \otimes (y_{WVM} - y_i\|_Q \leqslant \sum_{i=1}^{N} w_i \|a_i \otimes (y_i - y_i\|_Q, \quad j = 1, 2, \ldots, N. \tag{8.5}$$

The point-wise multiplication is denoted by $\otimes$.

8.1.2. Basic Directional Ordering and Filtering. These terms mean the standard directional processing applied in color imaging to each input vector y, $i = 1, 2, \ldots, N$ and associated with the angular norm, for example, as follows [1, 2, 5, 6]:

$$\alpha(y_i) = \sum_{j=1}^{N} A(y_i, y_j), \tag{8.6}$$

where $A(y_i, y_j)$ is the angle between two K dimensional vectors y_i and y_j .The final ordered sample $\alpha_{(1)}$, which minimizes the angle with all other vectors, forms

the associated output vector called *Basic Vector Directional Filter* output, where $\alpha_{(1)} \leqslant \alpha_{(2)} \leqslant, \ldots, \leqslant \alpha_{(N)}$. The principal drawback of this filter is its computational complexity due to the calculation of all the angles.

Figure 8.2 expresses the quantification of differences between the 2D vectors.

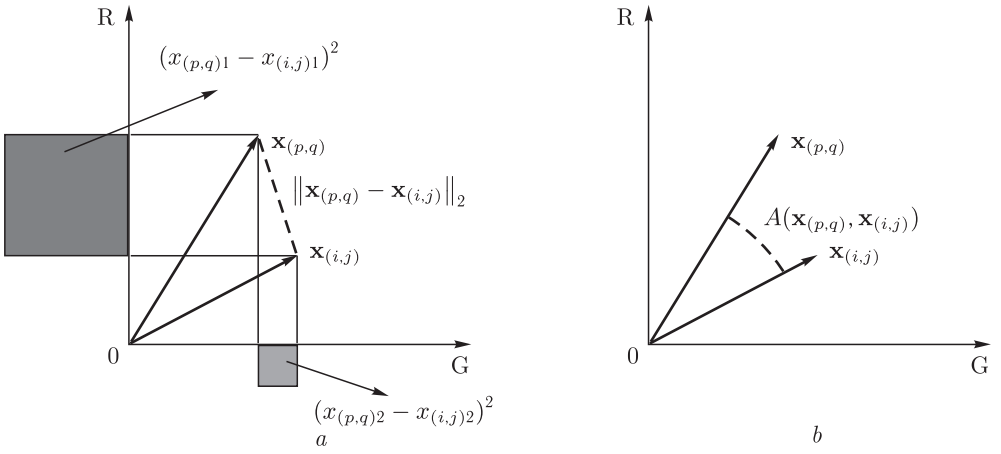


Fig. 8.2. Quantification of differences between the 2D vectors $x_{(p,q)} = [x_{(p,q)1}, x_{(p,q)2}]$ and $x_{(i,j)} = [x_{(i,j)1}, x_{(i,j)2}]$ expressed in red-green (RG) color space. (*a*) Expressed through the Euclidean metrics $\|x_{(p,q)} - x_{(i,j)}\|_2$ in the magnitude domain. (*b*) Expressed through the angle $A(x_{(p,q)}, x_{(i,j)})$ in the directional domain.

8.1.3. Directional Distance Filter. This filter is formed as a combination of the vector median filter and vector directional one and is expressed as a hybrid ordering criterion [6–8]:

$$\Omega = \left(\sum_{j=1}^{N} \|y_i - y_j\|_Q\right)^P \left(\sum_{j=1}^{N} A(x_i, x_j)\right)^{1-P}, \quad \text{for} \quad i = 1, 2, \ldots, N. \tag{8.7}$$

DDF output is the sample $y_{(1)}$ that minimizes the above criterion and is associated with $\Omega_{(1)}$ that satisfied the condition $\Omega_{(1)} \leqslant \Omega_{(2)} \leqslant \ldots \leqslant \Omega_{(N)}$. If $P = 1$, then DDF is equivalent to the VMF; if $P = 0$, then the DDF is exactly the BVDF.

The filters presented above are operating using fixed support window and can introduce excessive smoothing and blur image details or illuminate fine image details. There are several possibilities to improve the quality of filtering, but the principal idea is that, if the sample is free of noise, it should be leaved unchanged. One of the promising approaches is the switching scheme, when the filter is to decide whether it will use nonlinear robust procedure or switch to the identity operator.

8.2. Kernel Density Estimation Method

8.2.1. Adaptive Kernel Multichannel Filter. This approach uses smooth nonparametric estimate of distribution density. So, the similarity measure of two estimates of color distribution is the distance between two surfaces of 2D kernel density estimations in the normalized *RGB* color space. The data are represented by a set of sample values from an unknown density distribution to be estimated.

The multichannel kernel density estimate in the q-dimensional case is defied as [1, 2]

$$f(\overrightarrow{Y}) = \frac{1}{nh^q} \sum_{i=1}^{N} K\left(\frac{\|Y - Y_i\|}{h}\right). \tag{8.8}$$

The shape of approximated density depends heavily on chosen parameter h :small values of h lead to spiked density estimates; on other hand, big values can produce over-smoothed estimates and hide fine image details.

It is common to choose the Gaussian or exponential kernels, i.e., $K(z) = = \exp(-0.5\, z^{\tau} z)$ or $K(z) = \exp(-|z|)$ [1, 2], respectively.

This approach gives the following solution:

$$\widehat{Y}(y)_{np} = \sum_{l=1}^{n} x_l \left(\frac{h_l^{-M} K\left(\frac{y - y_l}{h_l}\right)}{\sum\limits_{l=1}^{n} h_l^{-M} K\left(\frac{y - y_l}{h_l}\right)} \right) = \sum_{l=1}^{n} Y_l \omega_l(y), \tag{8.9}$$

where $y_l \in W$ and $\omega_l(y)$ is a weight function in the interval [0,1].

The choice of the Gaussian kernel function makes it possible to find the optimal value of $h_{opt} = [4/(q+2)]^{\frac{1}{q+4}}\, \widehat{\sigma} n^{-\frac{1}{q+4}}$, where $\widehat{\sigma}$ is the standard deviation. In the case of the 2D model, $q = 2$.

Figure 8.3 presents a block diagram of the nonparametric algorithm [9, 10].

8.2.2. Adaptive Multichannel Nonparametric Filtering (AMNFs). In real applications of image processing, the vectors of the actual image are not available for observation, so it usually provides a suboptimal solution. One of such approaches is the adaptive multichannel nonparametric filtering (**AMNF**), which uses contaminated pixels to estimate the kernel:

$$\widehat{Y}(y)_{AMNF} = \sum_{l=1}^{n} y_l \left(\frac{h_l^{-M} K\left(\frac{y - y_l}{h_l}\right)}{\sum\limits_{l=1}^{n} h_l^{-M} K\left(\frac{y - y_l}{h_l}\right)} \right). \tag{8.10}$$

The block diagram is the same as presented in Fig.8.3 but with one difference: the reference vectors are contaminated ones. The expression $(x_l \rightarrow y_l)$ signifies that it is not necessary to use any reference filter output. Another possibility is to use the previously formed vector as the reference one, for example, the *VMF* output vectors [1, 2] are defined as

$$\widehat{x}(y)_{AMNF\_VM} = \sum_{l=1}^{n} x_l^{VM} \left(\frac{h_l^{-M} K\left(\frac{y - y_l}{h_l}\right)}{\sum\limits_{l=1}^{n} h_l^{-M} K\left(\frac{y - y_l}{h_l}\right)} \right). \tag{8.11}$$

So, the block diagram presented above is changed at the stage of forming the reference vector.

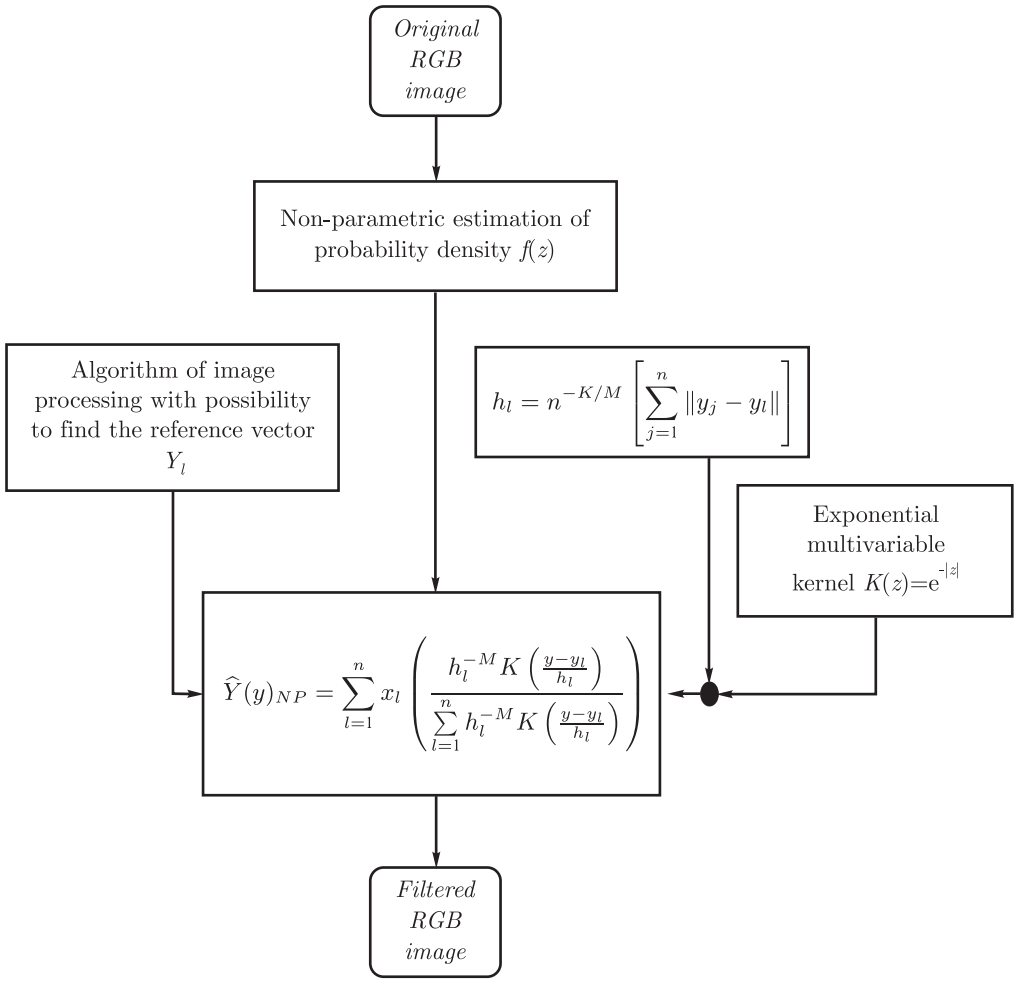


Fig. 8.3. Block diagram of nonparametric algorithm.

Other possibilities are also presented in this work. Besides *VMF*, other filter outputs are used, for example, the *MM-KNN* filter vectors [9–11] are defined as

$$\widehat{x}(y)_{AMN\_MMKNN} = \sum_{l=1}^{n} x_l^{MMKNN} \left(\frac{h_l^{-M} K\left(\frac{y-y_l}{h_l}\right)}{\sum_{l=1}^{n} h_l^{-M} K\left(\frac{y-y_l}{h_l}\right)} \right). \tag{8.12}$$

The quality of filtering depends on the window size, chosen values h_l, type of the kernel approximating the real one, and the reference vector, which usually should be obtained using another filtering scheme.

8.2.3. Generalizad Vector Directional Filter with Double Window (GVDF_DW). This filter employs two operations used in the different windows. The small window is used to preserve the details of the image, and larger one is used

in order to have more vector values and better approximation of the output vector. For example, when the VDF processing is applied, the directional operation and the magnitude one can be realized in two windows [1, 10].

Assume that W_1 and W_2 are two windows such that $W_1 \subset W_2$. Consider $y_{1_i}, i = 1, 2, \ldots, n$, as the vectors of an image in W_1, $y_{1_i} \in W_1$ and $f_{2_j}, j = 1, 2, \ldots, l$ are the vectors in W_2 window image outside W_1, i.e., $f_{2_j} \in W_2 - W_1$. The GVDF applied to $y_{1i}, i = 1, 2, \ldots, n$, produces an output set $\left\{y_1^{(1)}, y_1^{(2)}, \ldots, y_1^{(k)}\right\}$ according to the order of $\alpha_i' s$: $\alpha_1^{(1)} \leqslant \alpha_1^{(2)} \leqslant \cdots \leqslant \alpha_1^{(k)} \leqslant \cdots \leqslant \alpha_1^{(n)}$. The set $\left\{y_1^{(1)}, y_1^{(2)}, \ldots, y_1^{(k)}\right\}$ should be complemented by vectors of the sub-window $W_2 - W_1$, and these vectors are used to calculate the final output $\widehat{Y}$ as

$$\widehat{Y} = \Im\left\{y^{(1)}, y^{(2)}, \ldots, y^{(k)}\right\} = \Im\left\{GVDF\left[y_1, y_2, \ldots, y_n\right]\right\}. \tag{8.13}$$

For the vectors $Y_{2_j} \in W_2 - W_1$, one should obtain α'_{2_j} that relates to Y_{2_j} defined according to angles as

$$\alpha'_{2_j} = \sum_{i=1}^{n} A(Y_{2_j}, Y_{1_i}). \tag{8.14}$$

The application of Y_{2_j} changes the set $\left\{y_1^{(1)}, y_1^{(2)}, \ldots, y_1^{(k)}\right\}$ according to the following condition: $\alpha'_{2_j} \leqslant \alpha_1^{(k)}$.

This definition uses the internal window (W_1) to realize the directional ordering, and the vectors of external sub window ($W_2 - W_1$) are used in the magnitude processing if its vectors are close to the vector median of the sample. Figure 8.4 exposes the GVDF_DW algorithm with a double window: the first window is of 3×3 and the second one is of 5×5.

8.3. Fuzzy Logic Definitions and Properties

Fuzzy transformation (FZT) theory is briefly introduced in this section and the definitions of the fuzzy ranks are given. We explain two fundamental properties of the fuzzy ranks, i.e., the order invariant and spread sensitive properties. These properties show how fuzzy ranks jointly represent the rank order and spread information [12–15].

8.3.1. Fuzzy Logics Definitions. FZT theory addresses the relationship between an observation crisp sample vector $x_l = [x_1, x_2, \ldots, x_N]$, within which the samples are indexed according to their spatial locations, and the order statistic vector $x_L = [x_{(1)}, x_{(2)}, \ldots, x_{(N)}]$, which is generated by crisp rank ordering of the observed samples, so that $x_{(1)} \leqslant x_{(2)} \leqslant \ldots \leqslant x_{(N)}$. The crisp relationship between a spatial sample x_i and an order statistic $x_{(j)}$ can be represented by a binary spatial-ranking (SR) relation [15]

$$\begin{gathered} R = \{(x_i, x_{(j)}, \mu(x_i, x_{(j)})/x_i, \quad x_{(j)} \in X\}, \\ \mu(x_i, x_{(j)}) = \begin{cases} 1, & \text{for} \quad x_i \leftrightarrow x_{(j)} \\ 0, & \text{otherwise} \end{cases}, \end{gathered} \tag{8.15}$$

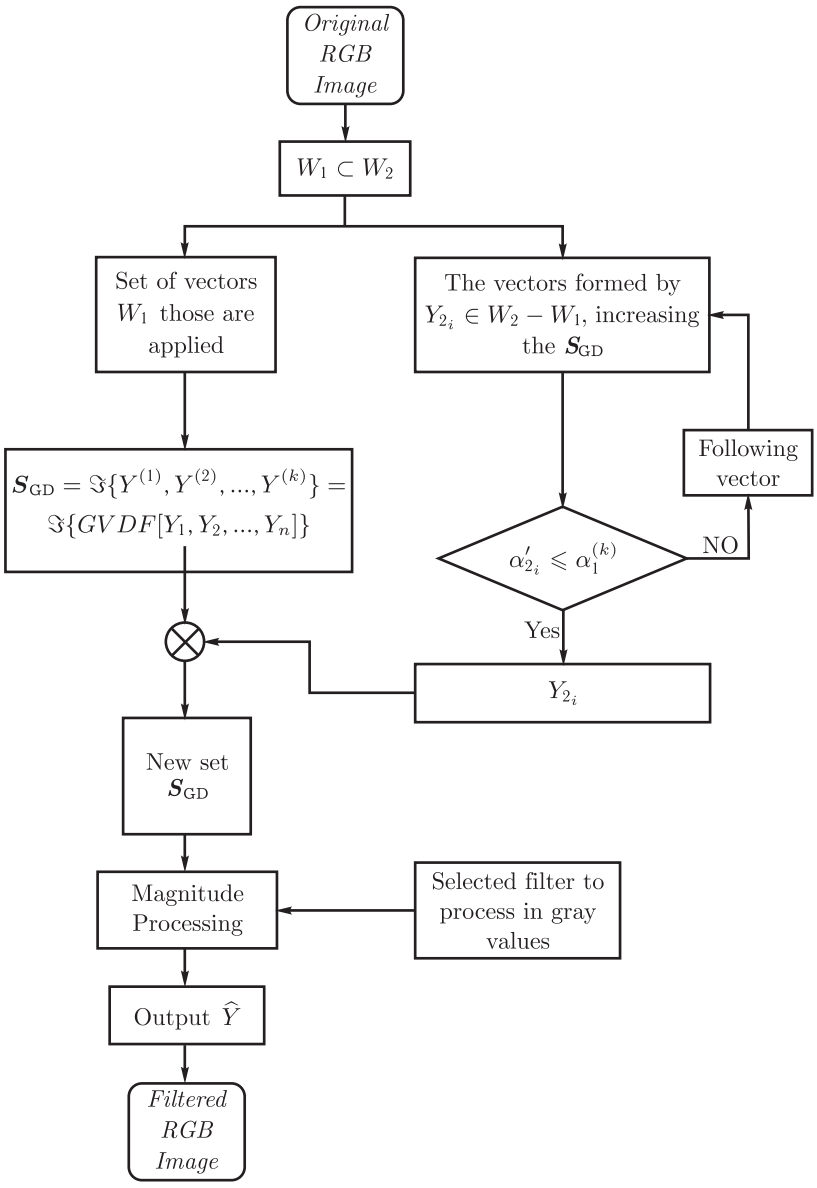


Fig. 8.4. Block diagram for VDF with double window.

where $x_i \leftrightarrow x_{(j)}$ indicates thatx_i has rank $x_{(j)}$. Thus, the full set of crisp SR relations can be represented by the crisp SR matrix as [15]:

$$R = \begin{pmatrix} R_{1,(1)} & \dots & R_{1,(N)} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ R_{N,(1)} & \dots & R_{N,(N)} \end{pmatrix}. \tag{8.16}$$

The crisp SR matrix defines the linear transformation between x_l and x_L, i.e., $x_l = x_L R^T$ and $x_L = x_l R$, where R^T is the transposition of R.

The spatial index vector, $s = [s_1, s_2, \ldots, s_N]$, where s_i is the spatial index of $x_{(j)}$, and the rank index vector $r = [r_1, r_2, \ldots, r_N]$, where r_i is the rank of x_i, are also given by linear transformations defined by the crisp R matrix, i.e., $s = [1 : N]R$, and $r = [1 : N]R^T$.

The crisp R matrix contains only the spatial and rank order information of the samples. In order to incorporate the spread information into the R relations, a real-valued membership function $\mu_F(*, *)$ is used to describe the *fuzzy* relationship between two arbitrary samples. The following intuitive constrains are imposed on the real-valued membership function [14, 15]:

$$\begin{aligned} &(1) \quad \lim_{|a-b|\to 0} \mu(a,b) = 1 \\ &(2) \quad \lim_{|a-b|\to \infty} \mu(a,b) = 0 \\ &(3) \quad |a_1 - b_1| \leqslant |a_2 - b_2| \Rightarrow \mu_F(a_1, b_1) \geqslant \mu_F(a_2, b_2). \end{aligned} \tag{8.17}$$

These constrains imply that two identical samples have relation 1, while two infinitely distant samples have relation 0. Also, the relation between samples should increase as the distance between them decreases.

In the case of color or multichannel imaging, the given two color pixels x and y resemble their physical similarity and meet the constraints of a fuzzy membership function as follows:

$$\begin{aligned} &(1) \quad \mu(x,y) \to 1, && \text{if } \|x - y\| \to 0; \\ &(2) \quad \mu(x,y) \to 0, && \text{if } \|x - y\| \to \infty; \\ &(3) \quad \mu(x_1, y_1) \geqslant \mu(x_2, y_2), && \forall \|x_1 - y_1\| \leqslant \|x_2 - y_2\|, \end{aligned} \tag{8.17a}$$

where $\|\ \|$ defines the norm used in the multichannel imaging.

Widely used membership functions that satisfy the constraints presented in (8.17) include [13, 15]:

the Gaussian membership function

$$\mu_G(a,b) = \exp[-(a-b)^2/2\sigma^2],$$

uniform membership function

$$\mu_U(a,b) = \begin{cases} 1, & |a - b \leqslant \alpha|, \\ 0, & \text{otherwise}, \end{cases}$$

and triangular membership function

$$\mu_U(a,b) = \begin{cases} 1 - |a-b|/\alpha, & |a - b \leqslant \alpha|, \\ 0, & \text{otherwise}, \end{cases}$$

where σ and α are free parameters used to control the function spread. It is important to mark that all three membership functions are symmetric, i.e., $\mu_F(a,b) = \mu_F(b,a)$.

Nevertheless, the Gaussian model usually is chosen to describe the similarity between two color pixels, mainly due to its success in the prior-art fuzzy ranked filters [16–18], and the exponential characteristics of the perceptual distance measures.

Two types the Gaussian functions are taken into account. The first is the Gaussian function based on the vector magnitude, that is,

$$\mu(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right). \tag{8.18}$$

The second is a vector extension of the Gaussian membership in the *RGB* or *YCbCr* color space, i.e., $\mu_2(x,y) = [\mu_2^{[1]}(x,y)\mu_2^{[2]}(x,y)\mu_2^{[3]}(x,y)]^T$, where

$$\mu_2^{[i]}(x,y) = \exp\left(-\frac{\left|x^{[i]}-y^{[i]}\right|^2}{2\sigma_i^2}\right) \tag{8.19}$$

denotes the membership function for the i-th color component and $[i]$ is the sample spread of the image data in that color component.

The full set of fuzzy SR relations among samples is, thus, represented by the fuzzy SR matrix

$$\widetilde{R} = \begin{pmatrix} \widetilde{R}_{1,(1)} & \cdots & \widetilde{R}_{1,(N)} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \widetilde{R}_{N,(1)} & \cdots & \widetilde{R}_{N,(N)} \end{pmatrix}, \tag{8.20}$$

where $\widetilde{R}_{i,(j)} = \mu(x_i, x_{(j)}) \in [0,1]$ and $\mu(x_i, x_{(j)})$is the utilized membership function.

The fuzzy spatial sample vector and fuzzy order statistic vector are defined as functions of the fuzzy SR matrix in a fashion analogous to the linear transformations between their crisp counterparts. In order to restrict the resulting values to the same ranges as the crisp counterparts, the fuzzy SR matrix is row normalized (denoted by $\widetilde{R}^l$) or column normalized (denoted by $\widetilde{R}^L$). Then the fuzzy spatial sample and fuzzy order statistic vectors are defined as

$$\widetilde{x}_l = x_L(\widetilde{R}^l)^T \quad \text{and} \quad \widetilde{x}_L = x_l\widetilde{R}^L.$$

Similarly, the fuzzy spatial index vector and fuzzy rank vector are given by

$$\widetilde{s} = [1:N]\widetilde{R}^L \quad \text{and} \quad \widetilde{r} = [1:N](\widetilde{R}^l)^T.$$

Carrying out the matrix expression for a single term in the above definitions yields the following expressions for the fuzzy sample $\widetilde{x}_i$, order statistic $\widetilde{x}_{(j)}$, spatial index $\widetilde{s}_j$, and rank $\widetilde{r}_i$ [14, 15]:

$$\widetilde{x}_i = \frac{\sum_{k=1}^N x_k\widetilde{R}_{i,(k)}}{\sum_{k=1}^N \widetilde{R}_{i,(k)}}, \quad \widetilde{x}_{(j)} = \frac{\sum_{k=1}^N x_k\widetilde{R}_{k,(j)}}{\sum_{k=1}^N \widetilde{R}_{k,(j)}}, \tag{8.21}$$

and

$$\widetilde{s}_i = \frac{\sum_{k=1}^N k\widetilde{R}_{k,(j)}}{\sum_{k=1}^N \widetilde{R}_{k,(j)}}, \quad \widetilde{r}_i = \frac{\sum_{k=1}^N k\widetilde{R}_{i,(k)}}{\sum_{k=1}^N \widetilde{R}_{i,(k)}}. \tag{8.22}$$

Thus, $\widetilde{x}_i$and $\widetilde{x}_{(j)}$ are weighted averages of the crisp order statistics and samples, and $\widetilde{s}_j$ and $\widetilde{s}_j$ are weighted averages of the indices $1, 2, \ldots, N$. The weights in each case are given by the fuzzy SR relations between the samples. The FZT refers to the mapping from the quadruple $\{x_l, x_L, s, r\}$ to its fuzzy counterpart $\{\widetilde{x}_l, \widetilde{x}_L, \widetilde{s}, \widetilde{r}\}$. A special class of FZT, the consistent FZT, utilizes membership functions satisfying the order invariant condition and is of great importance. Under consistent FZT, an order invariant property

is retained in the fuzzy order statistics and fuzzy ranks. In the following, we focus on investigating the order invariant and spread sensitive properties of the fuzzy ranks to see how they jointly represent the rank order and spread information.

8.3.2. Fuzzy Logics Properties.

8.3.2.1. Order Invariant Property.

The order invariant property is stated in the following theorem.

Theorem 1. *The FZT preserves the sample rank order, i.e., in a set of observation samples,* $\{x_1, x_2, \ldots, x_N\}$, *if* $x_i \leqslant x_j$ *or equivalently,* $r_i < r_j$, *where* r_i *and* r_j *are the crisp ranks of* x_i *and* x_j, *respectively, then* $\widetilde{r}_i \leqslant \widetilde{r}_j$, *where* $\widetilde{r}_i$ *and* $\widetilde{r}_j$ *are the fuzzy ranks of* x_i *and* x_j, *respectively, obtained through a consistent FZT.*

The proof of the above theorem is given in [15]. This theorem indicates that, under the consistent FZT, higher valued samples have higher fuzzy ranks. Thus, the fuzzy ranks represent the rank-order information of the samples in a fashion consistent with the crisp ranks. This implies that the fuzzy ranks can replace the crisp ranks in rank-order-based algorithms without distorting the rank-order information. The advantages of the replacement can be explained through the following spread sensitive property.

Theorem 2. *The FZT preserves the sample rank order, i.e.,* $\widetilde{x}_{(1)} \leqslant \widetilde{x}_{(2)} \leqslant \ldots \ldots \leqslant \widetilde{x}_{(N)}$, *if and only if the membership function* $\mu(*,*)$ *is such that* $C(x,t,\Delta t) = \mu(x, t+\Delta t)/\mu(x,t)$ *is a monotonically nondecreasing function of* $x \in (-\infty, t) \cup [t+\Delta t, +\infty), \forall t, \Delta t \in R$, *and* $\Delta t \geqslant 0$.

Membership functions that satisfy this condition are referred as *order invariant membership functions.* The functions presented before: the Gaussian, uniform, and triangular are order invariant membership functions. This theorem implies the following property: $r_i < r_j \Rightarrow \widetilde{r}_i < \widetilde{r}_j$ [15].

8.3.2.2. Distribution of the Fuzzy Samples.

Theorem 3. *For independent and identically distributed* (*i.i.d.*) *samples* $\{x_1, x_2, \ldots, x_N\}$, *the corresponding fuzzy samples* $\{\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_N\}$ *obtained by the aforementioned procedure based on a uniform membership function are identically distributed with the probability density function* (*pdf*) *approximately*

$$f_{\widetilde{x}_i}^{(}x) = (1-p_x)^{N-1} f_X(x) + p_x \sum_{u=1}^{N-1} (u+1) f_{S_{u+1}Ix}((u+1)x) P_u(x). \tag{8.23}$$

This approximation is correct if $f_X(x)$ is approximately constant in the interval $[x-2\alpha, x+2\alpha]$ and $u \gg 1$. The p_x is the probability of a crisp sample within the neighborhood $[x-2\alpha, x+2\alpha]$. The $f_{S_{uIt}}(*)$ is the conditional pdf of the local sum of u(crisp) samples in the interval $[x-\alpha, x+\alpha]$, and

$$P_u(t) = \begin{pmatrix} N-1 \\ u \end{pmatrix} p_t^u (1-p_t)^{N-1-u}$$

is the probability that there are u active neighboring samples around t.

The analysis of this equation shows that when p_x is small, the first term dominates and the fuzzy samples have similar distribution as the crisp samples. In the opposite case, the fuzzy samples behave like the local mean of crisp samples in a neighborhood of size 2α. We can conclude that the crisp samples having similar values are further clustered around their local mean. While the samples having disparate values tend to remain unchanged. So, fuzzy samples reflect the spread information embedded in crisp samples, and this usually is named as clustering property of the FZT.

8.3.2.3. Spread Sensitive Property.

The spread sensitive property of the fuzzy ranks can be illustrated in the following simple example. Consider two sample vectors

$$x_1 = [1.1, 1.2, 1.3, 1.4, 2.1, 2.2, 2.3, 2.4, 2.5],$$
$$x_2 = [1.1, 1.2, 1.3, 1.4, 2.1, 2.2, 2.3, 12.4, 12.5].$$

Note that there are two sample outliers in the second sample. In spite of the spread difference between the two sample vectors, the integer-valued crisp rank vectors of x_1 and x_2 are identical, i.e., $r_1 = r_2 = [1, 2, \ldots, 9]$. The real-valued fuzzy rank vectors generated using the Gaussian membership function, contrastingly, are different and jointly reveal the rank and spread information:

$$\widetilde{r}_1 = [3.44, 3.63, 3.83, 4.05, 5.69, 5.89, 6.08, 6.26, 6.42],$$
$$\widetilde{r}_2 = [3.11, 3.23, 3.37, 3.51, 4.68, 4.84, 4.99, 8.49, 8.50].$$

It is important to note that the fuzzy rank values are still confined in the range of $[1, N]$.

This enables us to define those fuzzy ranks close to 1 or N as the *extreme-valued fuzzy ranks*, as is practiced in the crisp rank case. In contrast to the crisp ranks, however, the fuzzy ranks of similarly valued samples are also similarly valued and are around the local median of their crisp counterparts; while the fuzzy ranks of disparately-valued samples, such as the two outliers in x_2, are very close to their crisp counterparts.

Hence, extreme-valued fuzzy ranks can be used to detect the presence of sample outliers more accurately than the extreme crisp ranks. Moreover, middle-valued fuzzy ranks indicate that the corresponding samples are similarly valued and no abrupt transitions, such as an edge in an image, are present in the input data.

8.3.2.4. Spread Parameter Dependence: It is worth mentioning that different spread parameter used in the membership function results in different fuzzy ranks. For the limit cases, we have [15]

$$\lim_{\gamma \to +\infty} \widetilde{r}_i = \frac{N+1}{2} \quad \text{and} \quad \lim_{\gamma \to 0} \widetilde{r}_i = r_i, \quad i = 1, 2, \ldots, N, \tag{8.24}$$

where N is the window size and γ is the spread parameter of a general membership function used in FZT.

The most appropriate spread parameter for generating the fuzzy ranks is application dependent. Optimization of the spread parameter is feasible and may be required in certain cases, while empirical values work well in others.

8.4. Fuzzy Generalization of Some Classical Filters

8.4.1. Fuzzy Identical (FI) Filter. The output of crisp identity filter is expressed as $Y_I = x_c$, where c is the special index of center sample in the filtering window. The fuzzy identity filter can be defined as [15]

$$Y_{FI} = \widetilde{x}_C = \frac{\sum_{k=1}^{N} x_k \widetilde{R}_{C,k}}{\sum_{k=1}^{N} \widetilde{R}_{C,k}}. \tag{8.25}$$

Unlike the crisp identity filter, which gives an output identical to the input, the fuzzy identity filter extracts the spread information from the input signal. That results in better edge restoration. After applying the fuzzy identical filter, each subgroup is more tightly clustered around its local means. This results in effective smoothing of undesirable

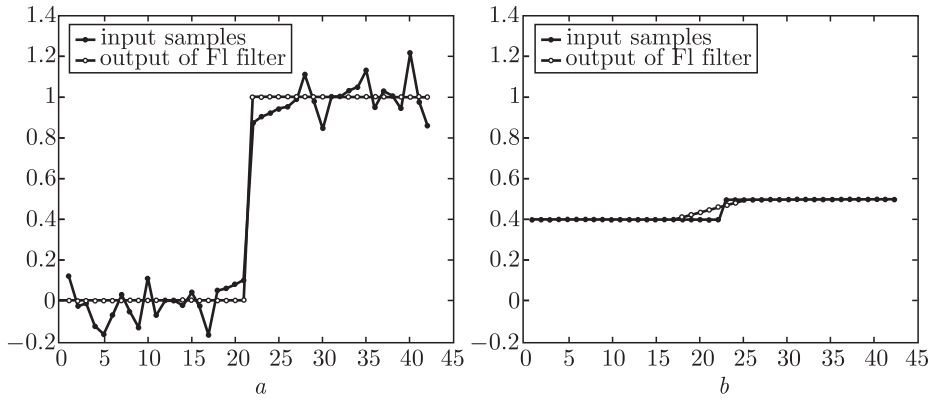


Fig. 8.5. Edge restoration of fuzzy identity filter. a) The small noisy edges are suppressed and the strong edge is restored. b) The small step edge is smoothed. The filtering window size is 9. Gaussian membership function with spread parameter $\sigma^2 = 0.1$ is used in filtering.

perturbations in large uniform regions as well as step edge restoration. In the case of a small step edge, the step transition is smoothed rather than restored. This last property if the identity fuzzy operator can be applied in image deblocking applications to remove the blocking artifacts while preserving true more large edges.

8.4.2. Fuzzy WM Filter. Classical WM filter presented above can be written as

$$Y_{WM} = MEDIAN\left[|w_1| \circ \operatorname{sing}(w_1)x_1, w_2 \circ \operatorname{sing}(w_2)x_2, \ldots,\ w_N \circ \operatorname{sing}(w_N)x_N\right],$$

where operator $\circ$ denotes replication for nonnegative integer weights w_i with a new set of samples $\{y_i = \operatorname{sign}(w_i)x_i,\ i = 1, 2, \ldots N)\}$. Defining ranks of this novel set and their fuzzy counterparts the FWM filter can be written as

$$Y_{FWM} = MEDIAN\left[|w_1| \circ \operatorname{sing}(w_1)\widetilde{x}_1, w_2 \circ \operatorname{sing}(w_2)\widetilde{x}_2, \ldots, w_N \circ \operatorname{sing}(w_N)\widetilde{x}_N\right] \quad (8.26)$$

Presented in literature, the FWM filters proved to be effective in removing heavy tailed noise while preserving image details.

One of the important properties of the FWM is it unbiasness in estimation of the inputs mean when input samples are symmetrically distributed. This property is exactly the same as in classical WM smothers

Typical FWM filter is the FWM high-pass filter, usually used in image sharpening to extract the image edges, for example, with the weight mask

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}.$$

8.4.3. Fuzzy LUM Smoother (FLUM). The classical LUM smoother with parameter $k \leqslant (N+1)/2$ is defined by the equation

$$Y_{LUM} = \begin{cases} x_{(k)}, & r_C < k, \\ x_C, & k \leqslant r_C \leqslant N-k+1, \\ x_{(N-k+1)}, & r_C > N-k+1, \end{cases}$$

where N is the filtering window size, C is the spatial index of the central sample, and r_C is its crisp rank. Replacing r_C with its fuzzy counterpart, we can write the output of the FLUM filter [15] as

$$Y_{FLUM} = \begin{cases} x_{(k)}, & \widetilde{r}_C < k, \\ x_C, & k \leqslant \widetilde{r}_C \leqslant N-k+1, \\ x_{(N-k+1)}, & \widetilde{r}_C > N-k+1. \end{cases} \tag{8.27}$$

Since the FLUM filter incorporates sample spread information, it can more effectively identify true outliers and improve filter performance. We explain this by the following example. Consider two sample vectors x_1 and x_2 as it was plotted n Figs. 8.6 a) and c), respectively. In x_2, two of nine samples are outliners. It is easy to calculate crisp rank vectors that are identical and therefore independent to the input sample spread. The real-valued fuzzy rank vectors, however, do reflect sample spread. The similarity valued samples in a vector have similar fuzzy ranks, which are close in value to local median of their crisp ranks. Application of the LUM filter to vectors x_1 and x_2 for $k \neq 1$ gives oversmoothing in both vectors, while $k < 3$ introduces undersmoothing on x_2. In contrast, the choice of $k = 3$ for the FLUM smoother avoids oversmoothing and undersmoothing on both sample vectors.

Using equation (8.27), we can define the general FLUM filter as:

$$Y_{G\_FLUM} = \begin{cases} x_{(k)}, & \widetilde{r}_C < k, \\ x_{(l)}, & t < r_C < h, \quad x_C < t_l, \\ x_{(N-l+1)}, & N-h+1 < \widetilde{r}_C \leqslant N-l+1, \quad x > t_l, \\ x_{(N-k+1)}, & \widetilde{r}_C > k, \\ x_C, & \text{otherwise}, \end{cases} \tag{8.27a}$$

where $k \leqslant l \leqslant h \leqslant (N+1)/2$, and $t_l = (x_{(l)} + x_{(N-l+1)})/2$. The parameter k controls the smoothing level, and greater k value introduces stronger smoothing; l controls the sharpening level, where the smaller l level introduces stronger sharpening; h controls fine detail preservation, where greater h value preserves larger fine details. The spread parameter of membership function is also important in filter design.

The statistic properties of FLUM filter, such as *breakdown* and *false-alarm probabilities* were investigated in [19]. The first one is the probability of outputting an impulse given a certain probability of impulse occurrence in the input, and this indicates the impulsive noise suppression capability in smoothing. The *false-alarm probability* is the probability of uncorrupted sample being modified, and indicates the detail preservation capability of the smoothing.

The analysis of these parameters have shown that *breakdown probability* for FLUM Smoother is the same as that of the LUM Smoother, but *breakdown probability* for FLUM Sharpener is lower than that of the LUM Sharpener for common parameter l and $h \neq (N+1)/2$. On other hand, the *false-alarm probability* for FLUM Sharpener and General FLUM filter is lower than that of LUM Sharpener or General LUM filter, respectively, for common parameters and $h \neq (N+1)/2$. So, such property indicates that FLUM filters have better detail preservation than the LUM filters, but this property requires that the spread parameter should be sufficiently large with respect to the input signal. These characteristics indicate the robustness to the impulsive noise.

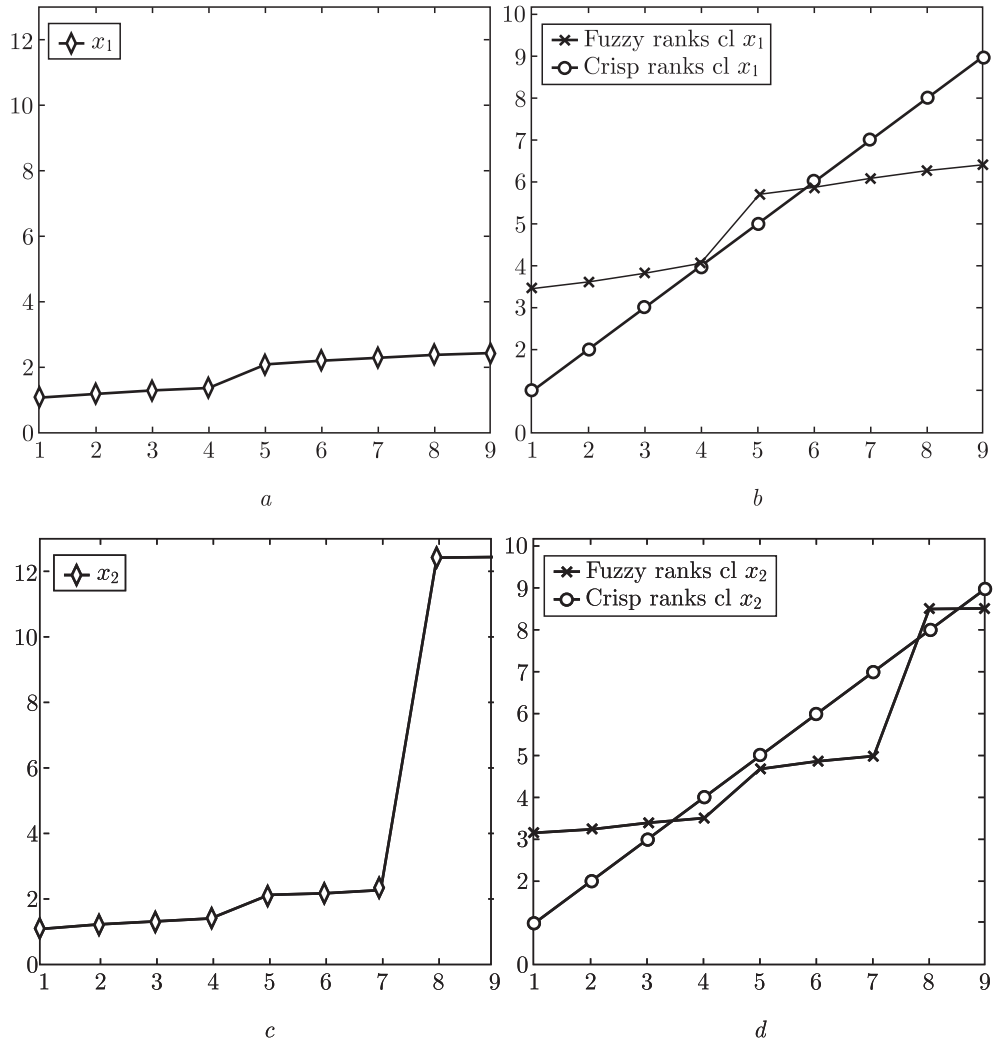


Fig. 8.6. a) sample vector x_1, b) crisp and fuzzy rank vector x_1, c) sample vector x_2, b) crisp and fuzzy rank vector x_2. The fuzzy ranks were obtained using a Gaussian membership function with $\sigma^2 = 1$ [15].

8.5. Multidimensional and/or 3D Video Processing Algorithms

8.5.1. Color Imaging Algorithms.

8.5.1.1. VRMKNN and AMN-MMKNN Filters.

We have introduced the Vector Rank M-Type K-Nearest Neighbor (VRMKNN) filter [9–11]. This filter utilizes multichannel image processing based on the vector approach [1–3], and the Rank M-Type K-Nearest Neighbor (RMKNN) algorithm [20]. The VRMKNN filter provides fine detail preservation applying the KNN algorithm [21] and the combined RM-estimators (see Sec. 6.4) by following way. The redescending M-estimators with different influence functions are combined with the R- (median,

Wilcoxon, or Ansari–Bradley–Siegel–Tukey) estimators to obtain sufficient impulsive noise suppression for each channel by using the vector approach.

The following representation of the KNN filter is often used:

$$\theta_{KNN} = \sum_{i=1}^{N} a_i x_i / \sum_{i=1}^{N} a_i \quad \text{with} \quad a_i = \begin{cases} 1, & \text{if} \quad |x_i - x| \leqslant T, \\ 0, & \text{otherwise}, \end{cases}$$

where T is the threshold, x_i is the input data sample in a sliding window, and x is the central element in a window to be estimated. Usually, T is equal to twice the standard noise deviation as in the *Sigma filter* [21]. We have proposed another scheme that employs the influence functions in the combined RM-estimators. For convenience, the VKNN filter is written below as

$$\widehat{\theta}_{KNN} = \frac{1}{K_c} \sum_{m=1}^{N} \psi(y_m)\, y_m, \tag{8.28}$$

where $\{y_m\}$ is the set of the noisy image vectors in a sliding filter window, which includes $m = 1, 2, \ldots, N$ (N is odd) vectors $y_1, y_2, \ldots, y_N$ located at spatial coordinates; and $\psi(y_m)$ is the influence function defined in the simple case as

$$\psi(y_m) = \begin{cases} 1, & \text{if } y_m \text{ are } K_c \text{ samples, which values are closest} \\ & \text{to the value of the central sample } y_{(N+1)/2}, \\ 0, & \text{otherwise.} \end{cases}$$

To improve the robustness of the VKNN filter, we use the RM-estimators (6.32)–(6.34) presented in Section 6.4. Modifying these estimators for 3D vector data, we can write the Vector Rank M-type K-Nearest Neighbour (VRMKNN) filters in the form [10, 11]

$$\widehat{\theta}^{(q)}_{VMMKNN} = MED\left\{g^{(q)}\right\}, \tag{8.29}$$

$$\widehat{\theta}^{(q)}_{VWMKNN} = MED\left\{\frac{g^{(q)} + g_1^{(q)}}{2}\right\}, \tag{8.30}$$

$$\widehat{\theta}^{(q)}_{VABSTMKNN} = MED\left\{\begin{array}{ll} R^{(q)}_{(k)}, & k \leqslant [K_c/2], \\ \dfrac{R^{(q)}_{(k)} + R^{(q)}_{(l)}}{2}, & [K_c/2] < k \leqslant K_c, \\ & k \leqslant l. \end{array}\right\}, \tag{8.31}$$

where $\widehat{\theta}^{(q)}_{VMMKNN}$, $\widehat{\theta}^{(q)}_{VWMKNN}$, and $\widehat{\theta}^{(q)}_{VABSTMKNN}$ represent the outputs of the VMMKNN, VWMKNN and VABSTMKNN filters, respectively; $g^{(q)}$ and $g_1^{(q)}$ are sets of K_c values of vectors y_m, weighted by value in accordance with the used influence function $\widetilde{\psi}(y_m)$ and closest to the estimate obtained at previous step, $\widehat{\theta}^{(q-1)}_{VRMKNN}$, in a sliding filter window; $R^{(q)}_{(k)}$ and $R^{(q)}_{(l)}$ represent the values of vectors having k and l ranks among the sliding window elements $g^{(q)}$, which are the members of the set of K_c number of vectors weighted in accordance with the used influence function $\widetilde{\psi}(y_m)$ and closest to the estimate obtained at previous step, $\widehat{\theta}^{(q-1)}_{VRMKNN}$; $\{y_m\}$ is the set of the noisy image vectors in a sliding filter window, which includes vectors $y_1, y_2, \ldots, y_N$ located at spatial coordinates (i, j); $\widehat{\theta}^{(0)}_{VRMKNN} = y_{(N+1)/2}$ is the initial estimate that is equal to central element in a sliding window; q is the index of the current iteration; and K_c is the number of the nearest neighbor vectors calculated in the form [10, 11, 22]

$$K_c = \left[K_{\min} + a \cdot D_s\left(y_{(N+1)/2}\right)\right] \leqslant K_{\max}. \tag{8.32}$$

Here, a controls the fine detail preservation; $K_{\min}$ is the minimal number of the neighbors for noise removal; $K_{\max}$ is the maximal number of the neighbors for edge restriction and fine detail smoothing; and $D_s\left(y_{(N+1)/2}\right)$ is the impulsive detector defined as follows [10, 22]:

$$D_s\left(y_{(N+1)/2}\right) = \left[\frac{MED\left\{|y_{(N+1)/2} - y_m|\right\}}{MAD}\right] + \left[\frac{1}{2} \cdot \frac{MAD}{MED\left\{y_m\right\}}\right]. \tag{8.33}$$

In equation (8.33), MED (y_m) is the median of the input data set $\{y_m\}$ in a sliding window, and MAD=MED$\{|y_m - MED(y_m)|\}$ is the median of absolute deviations from the median.

The algorithm terminates when $\widehat{theta}_{VRMKNN}^{(q)} = \widehat{theta}_{VRMKNN}^{(q-1)}$. The proposed filtering approach employs an iterative procedure. At the current q-th iteration, the procedure uses a vector data sample to form a set of elements whose values are closest to the estimate calculated at the previous step. Subsequently, the procedure calculates a median of this set or a more complex estimate according to the RM-estimators, presented in the equations (8.29) and (8.30). Then, it uses this median at the next $(q+1)$-*th* step as in the previous estimation. The number of neighbors K_c in the vector sample with closest values is calculated prior to iterations and is kept unchanged during the iterations for every central element (i, j). It is a measure of the local data activity within the sliding window and of the presence of impulsive noise at its center element, and it helps to preserve small features. Iterations have to be terminated when the current estimate becomes equal to the previous one. From simulations, we found that the iterations converge after one or two iterations, but their maximal number can attain 4–5, depending on image nature.

Another proposed AMN-MMKNN filter is based on adaptive nonparametric approach explained in 8.2.2, and determines the functional form of density probability of noise from data in a sliding filtering window [9, 22]. So, the AMN-MMKNN filter is presented by combining the Adaptive Multichannel Nonparametric (AMN) filter and the Median M-Type K-Nearest Neighbor (MMKNN) filter [2, 22]. Such a filter can be written as in (8.12):

$$\widehat{x}(y)_{AMN-MMKNN} = \sum_{l=1}^{N} x_l^{VMMKNN} \left(\frac{h_l^{-D} K\left(\frac{y - y_l}{h_l}\right)}{\sum_{l=1}^{N} h_l^{-D} K\left(\frac{y - y_l}{h_l}\right)} \right), \tag{8.34}$$

where x_l^{VMMKNN} values represent the VMMKNN filter, described in equation (8.29), to provide the reference vector; y is the current noisy observation to be estimated from given set $\{y\}_N$, y_l are the noisy vector measurements, h_l is the smooth parameter that is determined as follows:

$$h_l = N^{-q/D} \left(\sum_{j=1}^{N} \|y_j - y_i\| \right), \tag{8.35}$$

where $\|y_j - y_i\|$ is the absolute distance (L_1 metric) between the two vectors y_i and y_j for $\forall y_j, j = 1, 2, \cdots, N$; q is a parameter to be determined in the interval $0.5 > q > 0$; D is the dimension of the measurement space ($D = 3$, for color images); and the function $K(y)$ is the kernel function that has the exponential form $K(y) = \exp(-|y|)$ in the case of impulsive noise.

8.5.2.1. Discussion of the Simulation Filtering Results in Color Imaging.

Many filtering approaches exist in color imaging. As it is difficult to analyze all the existing algorithms, the objective performances and subjective visual results are compared here with some reference filters, such as VMF, GVDF, AMNF, etc., commonly used in literature. To determine the restoration properties and compare the qualitative characteristics of various color filters, the proposed 3×3 VRMKNNF (VMMKNNF, VWMKNNF, and VABSTMKNNF) with simple, Hampel's three part redescending, and Andrew's sine influence functions, the 3×3 AMN-VRMKNNF filter (AMN-VMMKNNF) with simple influence function, and also the 3×3 Vector Median (VMF), 3×3 α-Trimmed Mean (α-TMF), 3×3 Generalized Vector Directional (GVDF), 3×3 adaptive GVDF (AGVDF), 5×5 double window GVDF (GVDF_DW), 3×3 Multiple non-parametric (MAMNFE), 3×3 adaptive Multichannel nonparametric (AMNF), 3×3 adaptive Multichannel nonparametric Vector Median Filters (AMN-VMF), and two novel ones, named here as adaptive VMF (AVMF) [23] and fast adaptive similarity VMF (VMF_FAS) [10, 24] were simulated. These filters were computed and used according to references [1–3, 5–8, 22] to compare them with the proposed filtering framework. The reason of choice of these filters to compare them with the proposed ones is that their performance has been compared with various known color filters.

The widely used 320×320 RGB color (24 bits per pixel) «Lena», «Mandrill», and «Peppers» test images with different texture character were corrupted by impulsive noise according to the noise model presented in Chapter 5 with intensities varied in the wide range from 0% to 40% of spike occurrence in each a channel. Table 8.1 shows some comparative restoration results for several proposed and reference filters for *PSNR* performance in the case of the test image «Mandrill».

Table 8.1. PSNR in dB for different filters applied in case of test image «Mandrill».

Impulsive Noise Percentage	VMF	VMF_FAS	AVMF	GVDF	GVDF_DW	VMMKNNF Simple	VWMKNNF Simple	AMN-VMMKNNF Simple
2	24.111	**29.268**	24.390	21.038	21.298	24.772	29.039	23.680
4	24.053	27.736	24.316	20.972	21.260	24.644	**28.079**	23.651
8	23.873	26.044	24.090	20.861	21.172	24.380	**26.374**	23.543
10	23.7784	25.294	23.974	20.728	21.105	24.202	**25.502**	23.476
15	23.347	23.680	23.480	20.295	20.954	**23.774**	23.729	23.285
20	22.793	22.473	22.881	19.769	20.765	**23.202**	22.713	23.072
30	21.180	20.113	21.209	18.088	20.160	21.777	19.772	**22.467**
40	19.062	17.899	19.067	15.990	18.885	19.851	17.672	**21.331**
50	16.952	16.001	16.953	14.055	17.218	17.847	15.885	**19.764**

Table 8.2 exhibits the simulation results for objective criteria *PSNR* and *NCD*, employing the proposed filtering approach and some of the better reference filters according to Table 8.1.

Simulation results (see Table 8.2) clearly show that VMF_FAS and VWMKNN filter with simple cut influence function are the best algorithms in noise suppression for low noise intensity (from 2% to 10%). In high impulsive noise intensity (from 30 to 50%), the better PSNR criterion values have been obtained by algorithms AMN-VMMKNN and VMMKNN with simple cut influence function, and for 15% and 20% spike occurrence, the best algorithm is the VMMKNN (Simple Cut). The similar

Table 8.2. Comparison for NCD, MAE, and PSNR performances of proposed and reference filters.

Impulsive Noise Percentage	Algorithm	NCD			MAE			PSNR		
		Mandrill	Lena	Peppers	Mandrill	Lena	Peppers	Mandrill	Lena	Peppers
5	AVMF	0.0293	0.0096	0.008	7.36	2.39	1.97	24.27	30.95	30.81
	VMF_FASr	**0.010**	**0.0045**	**0.0045**	**2.54**	**1.194**	**1.14**	**27.21**	**31.85**	**31.19**
	ANF-VMMKNNF	0.035	0.0195	0.017	10.767	5.03	4.42	23.62	29.21	29.21
	VMF	0.034	0.016	0.012	8.71	4.29	3.14	24.02	30.07	30.30
	VWMKNNF Simple	**0.019**	**0.0096**	**0.008**	**4.96**	**2.55**	**2.114**	**27.69**	**31.45**	**30.91**
	VMMKNNF Simple	0.034	0.0169	0.014	8.74	4.44	3.54	24.58	30.22	30.34
10	AVMF	0.031	0.0117	0.0095	7.87	2.97	2.49	23.97	**30.09**	**29.79**
	VMF_FASr	**0.0159**	**0.0086**	**0.0081**	**4.06**	**2.35**	**2.07**	**25.29**	28.80	29.01
	ANF-VMMKNNF	0.0432	0.020	0.0182	11.04	5.23	4.66	23.48	28.94	28.71
	VMF	0.0349	0.0172	0.0132	8.96	4.57	3.49	23.78	29.46	29.44
	VWMKNNF Simple	**0.0226**	**0.0128**	**0.0121**	**6.13**	**3.56**	**3.212**	**25.50**	28.13	27.42
	VMMKNNF Simple	0.0359	0.0179	0.0146	9.19	4.73	3.847	24.20	**29.64**	**29.62**
15	AVMF	0.0334	0.0141	**0.0119**	8.60	**3.63**	**3.13**	23.48	**29.06**	**28.66**
	VMF_FASr	**0.0223**	**0.0135**	0.0142	**5.83**	**3.70**	**3.70**	23.68	26.28	25.63
	ANF-VMMKNNF	0.0443	0.0213	0.0193	11.38	5.46	4.92	23.285	28.59	28.32
	VMF	0.0363	0.0185	0.0151	9.43	4.92	3.95	23.35	28.64	28.44
	VWMKNNF Simple	0.0268	**0.0166**	0.0169	**7.55**	4.75	4.59	**23.73**	25.87	25.025
	VMMKNNF Simple	0.0378	0.0191	0.0162	9.76	5.07	4.26	**23.77**	**28.93**	**28.71**
20	AVMF	0.0362	**0.0166**	**0.0150**	9.49	**4.41**	**3.92**	22.88	27.83	27.30
	VMF_FASr	**0.0354**	**0.0178**	**0.0161**	**7.69**	**5.00**	**4.84**	22.47	24.80	24.45
	ANF-VMMKNNF	0.0455	0.0222	0.0209	11.77	5.74	5.26	**23.07**	**28.18**	**27.82**

Continue Table 8.2.

Impulsive Noise Percentage	Algorithm	NCD			MAE			PSNR		
		Mandrill	Lena	Peppers	Mandrill	Lena	Peppers	Mandrill	Lena	Peppers
20	VMF	0.0384	0.0200	0.0172	10.11	5.42	4.53	22.79	27.58	27.19
	VWMKNNF Simple	**0.0323**	0.0207	0.0199	**9.07**	6.120	5.34	22.71	24.06	24.42
	VMMKNNF Simple	0.0399	0.0206	0.0183	10.48	5.54	4.76	**23.20**	**27.96**	**27.68**
30	AVMF	0.0439	**0.0236**	**0.0239**	**12.00**	**6.53**	**6.27**	21.21	24.89	24.02
	VMF_FASr	**0.0427**	0.0279	0.0316	**11.84**	8.09	8.29	20.11	22.19	21.53
	ANF-VMMKNNF	0.0487	0.0253	0.0266	12.91	**6.69**	**6.48**	**22.47**	**27.04**	**26.42**
	VMF	0.0449	0.0253	**0.0250**	12.30	7.04	6.58	21.18	24.83	23.99
	VWMKNNF Simple	**0.0429**	0.0303	0.0348	13.03	9.23	9.55	19.77	21.44	20.61
	VMMKNNF Simple	0.0456	0**.0252**	0.0256	12.44	6.92	6.63	**21.78**	**25.52**	**24.63**
40	AVMF	**0.0546**	**0.0341**	**0.0393**	15.88	10.07	10.37	19.07	21.45	20.54
	VMF_FASr	0.0592	0.0415	0.0509	17.41	12.68	13.59	17.90	19.49	18.65
	ANF-VMMKNNF	**0.0546**	**0.0316**	**0.0384**	**15.04**	**8.74**	**9.06**	**21.33**	**24.86**	**24.07**
	VMF	0.0550	0.0348	0.0397	15.99	10.26	10.48	19.06	21.44	20.54
	VWMKNNF Simple	0.0569	0.0434	0.0520	17.98	13.61	14.46	17.67	19.05	18.24
	VMMKNNF Simple	**0.0543**	0.0334	**0.0393**	**15.60**	**9.60**	**10.02**	**19.85**	**22.43**	**21.40**
50	AVMF	0.0691	0.0484	0.0606	21.37	15.07	16.33	16.95	18.68	17.70
	VMF_FASr	0.0771	0.0586	0.0755	24.07	18.57	20.59	16.00	17.28	16.33
	ANF-VMMKNNF	**0.0639**	**0.0419**	**0.0571**	**18.54**	**12.19**	**13.39**	**19.76**	**22.37**	**21.34**
	VMF	0.0692	0.0485	0.0607	21.42	15.13	16.36	16.95	18.69	17.70
	VWMKNNF Simple	0.0736	0.0593	0.0745	24.09	19.40	20.88	15.89	16.99	16.20
	VMMKNNF Simple	**0.0664**	**0.0452**	**0.0596**	**20.27**	**13.72**	**15.21**	**17.85**	**19.76**	**18.63**

numerical simulation results for PSNR criterion have been obtained in filtering of other test images: «Pepper» and «Lena» [10, 22]. Analysing the data presented in Table 8.2, one can see that, in low impulsive noise, intensity of (5%) the newest AVNF filter has some advantage in comparison with the filters based on the proposed approach, as well as others reference filters, VMF and AVMF, according the criteria used. For high-noise corruption intensity, when spike occurrence is more than 15%-20% (10%, in the case of test image «Mandrill»), the proposed algorithms presents the best performance in the PSNR criterion. It is easy to see that the NCD performance presented in this table favor the VMF_FAS and AVNF for low impulsive noise corruption, less than 20%. For high-noise corruption intensity, it is difficult to select the best filter. We can only note that for «noisy» type images, such as «Mandrill», the NCD performance values of the VMF_FAS filter and proposed VWMKNNF (Simple) filter are very similar. Finally, for very high impulsive noise corruption when the percentage is more than 40%, the better NCD performance values are presented by ANF-VMMKNN filter. It is necessary to note that when the objective criteria MAE and NCD show some advantage in favor of the VMF_FAS and AVNF filters, its PSNR values are less by 0.7–1.5 dB in comparison with what the AMN-VMMKNN filter gives.

So, the presented comparison of the objective criteria shows that the restoration performance of VRMKNNF and AMN-VRMKNNF is often better than that of other analyzed filters, at least for high impulsive noise corruption, more than 10-15 %.

Figure 8.7 exhibits the processed images (and its zoomed parts) for test image «Mandrill» explaining the impulsive noise suppression and detail preservation according to Table 8.2.

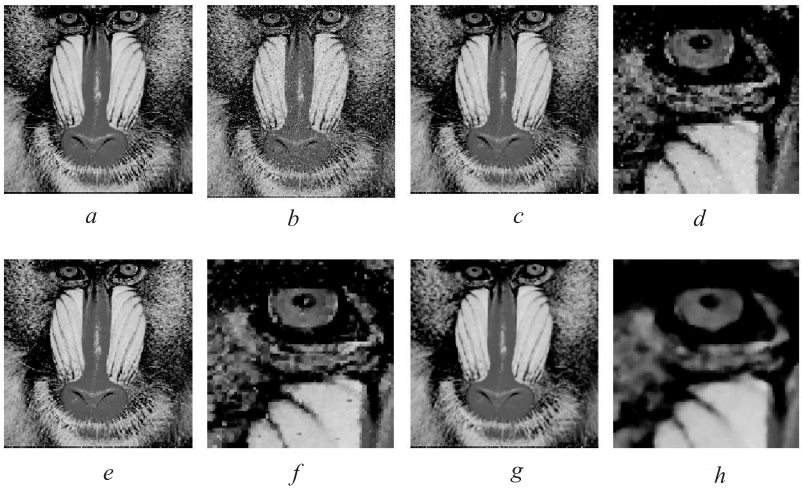


Fig. 8.7. Subjective visual quantities of restored color image «Mandrill», a) Original image «Mandrill»; b) Input noisy image corrupted by 10% impulsive noise in each channel; c) VWKNNNF (Simple) filtered image; d) VWKNNNF (Simple) filtered zoom part of (c), e) VMF_FAS filtered image; f) VMF_FAS filtered zoom part of (e), g) AMN-VMMKNNF filtered image, and h) AMN-VMMKNNF filtered zoom part of (g).

One can see, analyzing these error images, that the VNF_FAS filter present slightly better visual subjective performance in fine detail preservation, but, at the same time, it shows worse impulsive noise suppression in comparison with proposed filtering tech-

nique. The proposed VMMKNNF and AMN-VMMKNNF appear to have a good subjective quality. The VWKNNNF (Simple) output is characterized by sufficiently good fine detail preservation and noise suppression as well. Another proposed AMN-VMMKNNF presents excellent noise suppression but some image details are blurred. These subjective results confirm objective performances presented in the Tables 8.1 and 8.2.

The parameters for VRMKNNF and AMN-VRMKNNF filters and influence functions were found after numerous simulations in different test images degraded by impulsive noise. The values of parameters of the proposed filters were $0.5 < a < 15$, $K_{\min} = 5$, and $K_{\max} = 8$, and the parameters of the influence functions were: $r \leqslant 81$ for *Andrews sine*, and $\alpha = 10$, $\beta \leqslant 90$ and $r = 300$ for *Hampel three part redescending*. The idea was to find the parameter values when the values of the PSNR and MAE criteria would be optimal. The $K_{\min}$ and a values were varied from 1 to 8, and from 0 to 20, respectively. The simulation results have shown that the best performances were obtained when $K_{\min} \geqslant 5$ and $a \geqslant 2$, respectively. The parameters α, β, and r were found for different influence functions, for example, in the case of the Hampel function the optimum value α was equal to 14 for image «Mandrill», 10 for image «Lena», and 12 for video sequence «Miss America», and the value r is changed from 300 for «Mandrill», 280 for «Lena», and 290 for «Miss America». Therefore, there are some variations of about $\pm 10\%$ of the PSNR performance with the use of other parameter values, which are different from those presented here.

Finally, we have standardized these parameters as constants to realize the implementation of the proposed algorithms for real-time applications.

We also applied the proposed filters to process color video sequences presented in QCIF (Quarter Common Intermediate Format). This picture format uses 176×144 (24 bits per pixel) luminance pixels per frame. Three QCIF video color sequences «Miss America», «Flowers», and «Foreman» have been filtered to demonstrate that the proposed algorithms can potentially provide a real-time filtering solution [10]. The test video color sequences were contaminated by impulsive noises with a different percentage of spike occurrences in each channel. The restoration performance (PSNR, MAE, and NCD) in a form of its mean and root-mean-square (RMS) ones over the video sequence «Flowers» are presented in Table 8.3.

This table shows the comparison results for different reference and proposed filters applied in processing of the sequence «Flowers» contaminated by 5%, 10%, and 20% impulsive noise. One can see that for low impulsive noise contamination (5% and 10%) better performance is achieved by the VMF_FAS and AVMF filters.

At the same time, we may conclude that noise suppression performance and the PSNR criterion obtained by the proposed filtering technique are often very similar to those achieved by previously mentioned reference filters. In the case of 20% impulsive noise contamination, the AMN-VMMKNNF and VMMKNNF (Simple) are the best algorithms from the viewpoint of noise suppression quality.

Since the frames in the sequences have different image texture and changing object structure, and the noise samples vary from frame to frame, analyzing the whole video sequence in terms of mean and RMS of different criteria, we have confirmed the robustness and statistical significance of the proposed technique in noise suppression and fine detail preservation. Figure 8.8 shows the filtered frame illustrating subjective visual quality for sequence «Flowers» and conforming good quality of the processed frame by the proposed filters.

8.5.2. 3D Ultrasound Filtering. The possibility to process 3-D images presents a new application where it is necessary to improve the quality of 3-D objects inside an image, suppressing a noise of different nature.

Table 8.3. Mean values and root-mean-square (RMS) values for the *PSNR, NCD* and *MAE* criteria over the video sequence «Flowers» for 5%, 10%, and 20% of impulsive noise contamination.

Impulsive Noise Percentage	Algorithm	Mean $PSNR$	RMS $PSNR$	Mean NCD	RMS NCD	Mean MAE	RMS MAE
5	VMF	27.67	0.43	0.0113	0.0010	5.35	0.37
	GVDF	25.54	0.33	0.0144	0.0012	6.72	0.38
	AMNF	25.61	0.41	0.0156	0.0012	7.49	0.40
	AVMF	28.00	0.46	0.0091	0.0009	4.30	0.34
	VMF–FAS	**30.61**	0.54	**0.0031**	0.0003	**1.47**	0.15
	AMN–VMMKNNF Simple	25.36	0.37	0.0159	0.0013	7.51	0.39
	VMMKNNF Simple	27.87	0.40	0.0119	0.0011	5.72	0.39
	VWMKNNF Simple	**29.39**	0.28	**0.0063**	0.0005	**3.15**	0.16
10	VMF	27.08	0.38	0.0120	0.0011	5.75	0.40
	GVDF	24.36	0.35	0.0156	0.0011	7.42	0.33
	AMNF	25.40	0.29	0.0168	0.0012	8.19	0.39
	AVMF	27.32	0.40	0.0102	0.0010	4.88	0.36
	VMF–FAS	**27.71**	0.32	0.0058	0.0004	2.76	0.13
	AMN–VMMKNNF Simple	25.47	0.28	0.0164	0.0013	7.75	0.41
	VMMKNNF Simple	**27.20**	0.36	0.0126	0.0011	6.16	0.40
	VWMKNNF Simple	25.86	0.33	**0.0097**	0.0006	**4.94**	0.18
20	VMF	25.02	0.34	0.0145	0.0013	7.08	0.46
	GVDF	21.83	0.59	0.0193	0.0006	9.53	0.35
	AMNF	23.57	0.27	0.0216	0.0012	10.96	0.39
	AVMF	25.12	0.34	**0.0134**	0.0012	**6.56**	0.432
	VMF–FAS	23.66	0.26	**0.0123**	0.0008	**5.93**	0.24
	AMN–VMMKNNF Simple	**25.13**	0.31	0.0178	0.0015	8.56	0.47
	VMMKNNF Simple	**25.21**	0.28	**0.0150**	0.0012	**7.51**	0.43
	VWMKNNF Simple	21.35	0.38	0.0165	0.0004	8.66	0.31

We now consider a monochrome 3-D image $Y(i,j,k)$, where i and j represent the 2-D spatial axes, and k is either the time coordinate or the third coordinate axis of the 3D image.

Here, we use the combined RM (Rank M-type) – estimators, described in Section 6.4, applying them to 3-D filtration [25–27]:
the 3-D MM-KNN (Median M-type K-Nearest Neighbor) filter

$$\widehat{Y}^{(w)}_{MMKNN}(i,j,k) = MED\left\{h^{(w)}(i+l, j+m, k+n)\right\}, \tag{8.36}$$

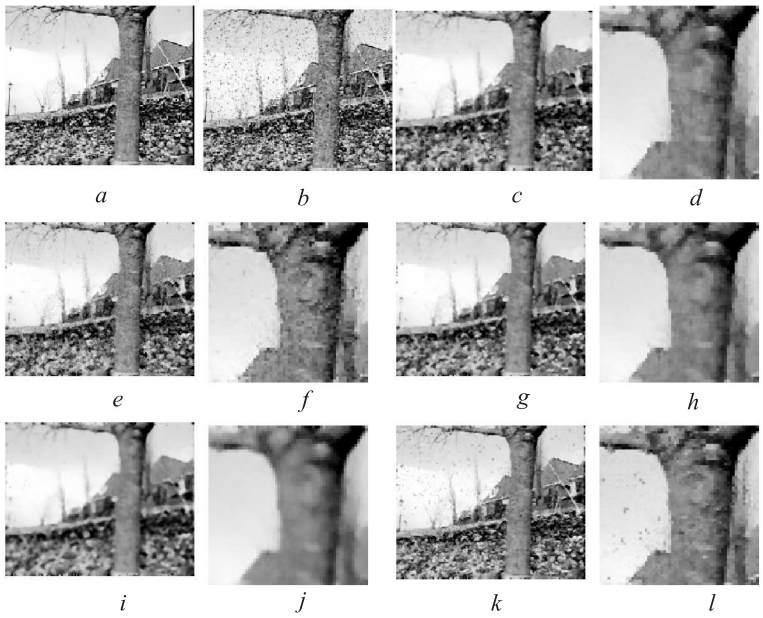


Fig. 8.8. Subjective visual qualities of restored color frame of video sequence «Flowers», a) Original test frame «Flowers», b) Input noisy frame (corrupted by 10% impulsive noise in each a channel), c) AVMF filtering frame, d) AVMF filtering zoom part of (c), e) VMF_FAS filtering frame, f) VMF_FAS filtering zoom part of (e), g) VMMKNNF filtering frame (Simple), h) VMMKNNF filtering zoom part of (g), i) AMN-VMMKNNF filtering frame, j) AMN-VMMKNNF filtering zoom part of (i), k) VWMKNNF filtering frame (Simple), l) VWMKNNF filtering zoom part of (k).

the 3-D WM-KNN (Wilcoxon M-type K-Nearest Neighbor) filter

$$\widehat{Y}^{(w)}_{WMKNN}(i,j,k) = \underset{i \leqslant j}{MED} \left\{ \frac{h^{(w)}(i+l, j+m, k+n) + h^{(w)}(i+l_1, j+m_1, k+n_1)}{2} \right\}, \quad (8.37)$$

and the 3-D ABSTM-KNN (Ansari–Bradley–Siegel–Tukey M-type K-Nearest Neighbor) filter

$$\widehat{Y}^{(w)}_{ABSTMKNN}(i,j,k) = \underset{i \leqslant j \leqslant k}{MED} \left\{ \begin{array}{ll} h^{(w)}(i+l, j+m, k+n), & i,j,k \leqslant \left[\frac{N}{2}\right], \\ \frac{h^{(w)}(i+l, j+m, k+n)}{2} + & \\ + \frac{h^{(w)}(i+l_1, j+m_1, k+n_1)}{2}, & \left[\frac{N}{2}\right] < i. \end{array} \right\}, \quad (8.38)$$

where $h^{(w)}(i+l, j+m, k+n)$ and $h^{(w)}(i+l_1, j+m_1, k+n_1)$ are the sets of K_c values of voxels weighted in accordance with the influence function $\widetilde{\psi}(X)$ in a rectangular 3-D cube slipping under filtration and containing voxels closest to the estimation obtained at the previous step, $\widehat{Y}^{(w-1)}_{RMKNN}(i,j,k)$. The initial estimation is $\widehat{Y}^{(0)}_{RMKNN}(i,j,k) = Y_{speckle}(i,j,k)$, and $\widehat{Y}^{(w)}_{RMKNN}(i,j,k)$ denotes the estimate at the iteration w. Here, $Y_{spekle}(i,j,k)$ is the 3-D image contaminated by noise in a rectangular 3-D grid of

size $M_1 \times M_2 \times M_3$. K_c is the current number of the nearest neighbor voxels defined similarly to equation (7.51), and noise detector $D_S(i,j,k)$ is presented in equation (7.52).

The algorithms finish when the condition $\widehat{Y}^{(w)}_{RMKNN}(i,j,k) = \widehat{Y}^{(w-1)}_{RMKNN}(i,j,k)$ is fulfilled (subscripts RMKNN denotes the MMKNN, or WMKNN, or ABSTMKNN). In the filters presented, the following influence functions are used: the *simple cut*, the *Hampel's three part redescending*, the *Andrew's sine*, Tukey biweight, and the*Bernoulli*.

Several parameters that characterize 3-D RM-KNN filters and influence functions were found after numerous simulations using a $3 \times 3 \times 3$ grid. The problem consists in finding the parameters values for which the *PSNR* and *MAE* criteria attain their optima. It was found that $K_{\min}$=5 and $K_{\max} = 24$ for any of 3-D RM-KNN filter are optimum. The parameters of influence functions were determined: $a = 8$ and $r = 255$ for the Simplest cut function; $a = 8$, $\alpha = 200$, $\beta = 230$, and $r = 256$, for the Hampel function; and $a = 10$ and $r = 255$, for the Andrew's sine function, the Tukey function, and the Bernoulli function.

During the investigations of 3-D filtering algorithms, US images were contaminated by impulsive noise that modeled a noise influence in communication channel. As it has been mentioned, the speckle noise is natural for US transducers. The 3-D RM-KNN filters with different influence functions have been evaluated, and their performance has been compared with known nonlinear 2-D filters, which were adapted to 3-D image processing. The following filters were applied as test ones for comparison: the *modified* α*-Trimmed Mean* [28], *Ranked-Order* (RO), *Comparison and Selection* (CS) [29], *Multistage Median* (MSM1 to MSM6) [30] *MaxMed* [31], *Selective Average* (SelAve), *Selective Median* (SelMed) [32], and *Lower-Upper-Middle* (LUM, LUM Sharp, and LUM Smooth) [33] filters. Initial US 3D sequence of an organ of the size of $640 \times 480 \times 90$ voxels (i.e., 90 frames of 2D image) was distorted by impulsive noise of random spikes with 5, 10, 15, 20, and 30% probability of appearance, and with the natural presence of speckle noise in the 3D image.

Table 8.4 shows the performance results of proposed filters and comparative results of different non linear filters applied to a frame of an original sequence presenting the *xz* plane. It is easy to see that the proposed method provides the better filtration quality in accordance with the *PSNR* and *MAE* criteria when a noise level is of 15% or high. Additional experiment was realized in the same sequence when it was degraded with 0.05 or 0.1 values of variance of speckle noise added to the natural speckle noise of the sequence. The performance results are depicted in the same table where one can see that the 3-D MM-KNN filters provide similar results in comparison to Ranked Order and Modified α-Trimmed Mean filters, and in some cases the proposed filters provide better performance.

Figure 8.9 illustrates this conclusion: the visual results in terms of restored and error images are shown, confirming that the RMKNN filters can suppress an impulsive noise and reduce the speckle noise better than other filters.

Other experimental investigations concerned different voxels cube configurations to provide better noise reduction.

Figure 8.10 shows nine voxel configurations used in the 3-D filtering algorithms. It is evident that the application of the smaller number of voxels in the filtration process would result in a significant reduction of the processing time. The implementation of such cube configurations in the proposed filters and α-Trimmed Mean filter is shown in Table 8.4 where, as is seen, the MM-KNN filter provides better results in comparison with the α-Trimmed Mean filter. In addition, it is evident that the g and i voxel configurations provide better suppression of high-intensity noise, whereas a and b configurations are efficient at a low noise intensity.

Table 8.4. Performance results of different filters in a frame of a US sequence degraded with impulsive noise (5, 10, 25, 20, and 30%), or with speckle noise (variance σ_ε^2: 0.05 and 0.1).

3-D Filters	5%		10%		15%		20%		30%		$\sigma_\varepsilon^2 = 0.05$		$\sigma_\varepsilon^2 = 0.1$	
	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE
Modified α-Trimmed Mean	24.9	7.5	24.9	7.1	24.8	7.2	24.7	7.4	24.3	7.8	20.4	15.14	19.1	**18.7**
Ranked Order	26.5	6.7	26.4	6.7	26.4	6.8	26.3	6.9	26.0	7.3	**21.6**	**14.5**	**19.8**	**18.2**
MSM1	**28.9**	**4.3**	28.5	**4.5**	27.7	**4.8**	26.7	5.3	23.9	7.1	20.6	17.6	18.1	23.7
MSM2	28.1	5.1	27.8	5.29	27.2	5.6	26.2	6.0	23.6	7.7	20.5	17.8	18.0	23.7
MSM5	**29.4**	**3.8**	**28.8**	**4.0**	27.6	**4.4**	26.0	5.2	22.6	7.6	19.6	20.2	17.0	27.4
MSM6	28.3	5.1	28.25	5.2	**28.0**	5.3	**27.7**	5.4	26.3	6.3	22.1	**14.7**	**19.7**	19.4
MaxMed	27.1	6.2	26.2	6.8	24.9	7.8	23.2	9.2	19.9	13.8	18.6	24.2	16.0	32.9
SelMed	27.4	5.6	27.0	5.9	26.7	6.2	26.3	6.4	25.4	7.1	20.8	15.8	19.0	20.1
LUM Smooth	**29.9**	**2.8**	**28.9**	**3.1**	27.3	**3.8**	25.3	**4.9**	21.0	8.8	17.95	25.1	15.4	33.8
LUM	18.6	15.5	18.2	16.2	17.9	17.1	17.6	17.7	17.6	17.7	15.5	31.4	14.4	36.7
MM-KNN CUT	28.8	4.3	**28.5**	4.6	**28.2**	4.8	**27.9**	**5.1**	**27.2**	**5.8**	**21.6**	15.2	18.9	21.0
MM-KNN HAMPELL	28.8	4.3	28.5	4.6	**28.2**	4.9	**27.9**	**5.2**	**27.2**	**5.8**	**21.6**	15.3	19.0	20.8
MM-KNN BERNOULLI	28.4	4.6	28.2	4.9	27.9	5.2	**27.7**	5.5	**27.0**	**6.1**	**22.7**	**13.3**	**20.1**	**17.8**

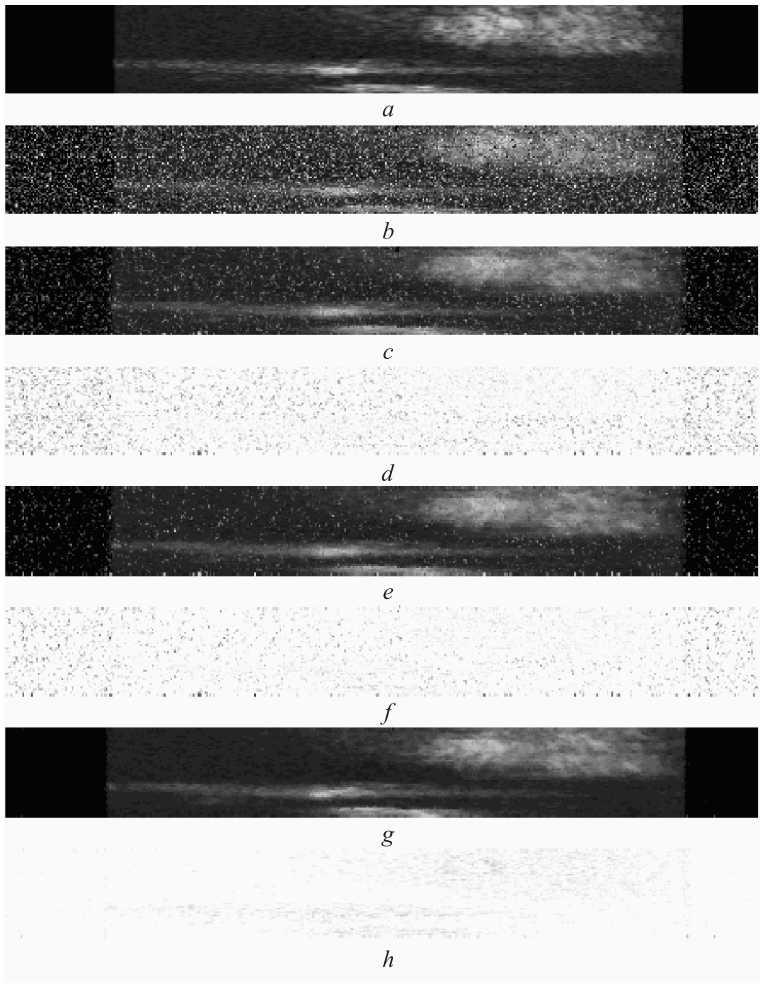


Fig. 8.9. Visual results in a frame of a US sequence. a) original image, b) image degraded by 30% of impulsive noise, c) restored image by LUM filter Smooth, d) error image produced by LUM filter Smooth, e) restored image by MSM5 filter, f) error image produced by MSM5 filter, g) restored image by filter MMKNN (Hampel), h) error image produced by MMKNN (Hampel).

8.5.3. 3D Vector Filters. We denote a current image voxel as $Y(i,j,t)$, where (i,j) and t indicate the spatial and temporal position in the video sequence, respectively, and consider a 3D sliding $3 \times 3 \times 3$ window around (i,j,t). Such a sliding window is applied to compute the filtered value $\widehat{Y}(i,j,t)$ in the video sequence.

In accordance with statistics rules, all the voxel values from the 3D window are ordered in a one dimensional array (in any particular order) [2]

$$\{x_L\} = (x_1, x_2, \ldots, x_N)^t. \tag{8.39}$$

In the case of a 3D sliding window $3 \times 3 \times 3$, N is equal to 27. Here, we use the directional processing ordering technique as ordering criteria [1].

Below we propose two methods for processing of video sequences [34–36]:

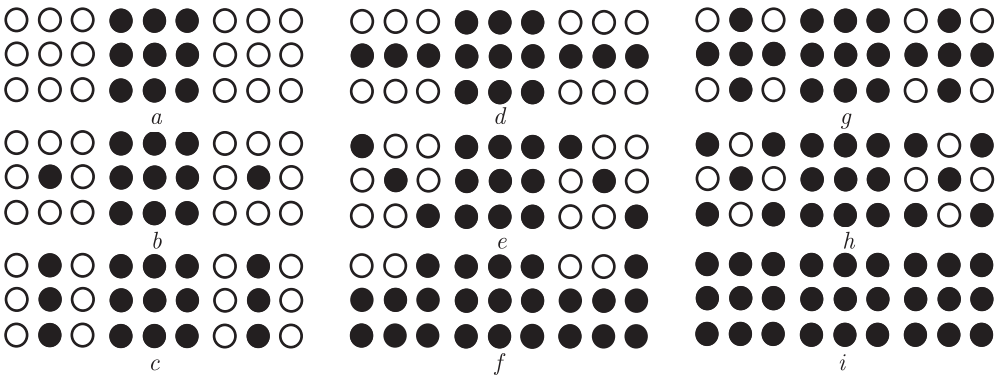


Fig. 8.10. Different configurations of processing cube.

Table 8.5. Performance results for different cube configurations in a frame of US sequence degraded with impulsive noise.

Voxel configuration	Impulsive noise					
	10%		20%		30%	
	MMKNN Cut Filter					
	PSNR	MAE	PSNR	MAE	PSNR	MAE
a	**31.18**	**3.64**	28.41	**4.54**	23.90	6.99
b	**31.31**	**3.77**	**29.41**	**4.42**	25.36	6.33
c	29.59	4.80	28.77	5.28	26.50	6.50
d	29.62	4.70	**28.86**	5.16	26.60	6.35
e	28.51	4.81	28.71	5.29	26.47	6.50
f	29.50	4.81	28.68	5.30	26.45	6.51
g	28.97	4.75	28.43	5.23	**27.10**	**6.13**
h	28.76	4.89	28.19	5.38	26.89	6.29
i	28.52	4.56	27.92	5.14	**27.22**	**5.82**
	Modified α-Trimmed Mean Filter					
a	**30.08**	**4.66**	26.32	6.98	22.28	11.66
b	**30.85**	**4.36**	28.24	5.69	24.16	9.17
c	29.63	4.96	**28.75**	**5.49**	26.14	7.44
d	29.73	4.84	**28.88**	**5.35**	26.24	7.29
e	29.54	4.97	28.68	5.49	26.10	7.45
f	29.52	4.98	28.66	5.50	26.08	7.46
g	28.65	5.41	28.29	5.68	**26.86**	**6.86**
h	28.41	5.57	28.04	5.85	**26.65**	**7.03**
i	26.04	7.41	25.75	7.76	25.22	8.59

a) first filtering method: The voxel values are sorted in x_L according to their difference with a current one $Y(i,j,t)$ to get a new 1D array $x_l = (x_{(1)}, x_{(2)}, \ldots, x_{(N)})^t$, where $x_{(1)} = Y(i,j,t)$ is the centrally located in the set of voxels for sliding 3D window, and $x_{(i)}$ $i = 2, \ldots, N$, are voxels that satisfy to the condition $A(x_{(1)}, x_{(i)}) \leqslant A(x_{(1)}, x_{(j)})$, $j = 2, \ldots, N$, where A is the angle between the multichannel voxels. So, the novel 1D array can be written as $x_l = (x_{(2)}, \ldots, x_{(N)})^t$. The set of the K-Nearest Neighbor vectors with respect to central voxel is obtained as

$$\{x^{(1)}, x^{(2)}, \ldots, x^{(K)}\}^t = VDKNNF[x_{(2)}, \ldots, x_{(N)}]^t, \quad 1 \leqslant K \leqslant N. \tag{8.40}$$

The first K terms of the ordered sequence $\{x^{(i)}\}^t$ constitute the output of the VDKNN (Vector Directional K-Nearest Neighbor) filter, which produces a set of vectors with approximately same direction. Finally, the magnitude processing step is applied to obtain a single output vector for each voxel. It is done by VMF (Vector Median Filter) [4] forming the VVDKNNVMF to process video sequence:

$$VMF\{x^{(1)}, x^{(2)}, \ldots, x^{(K)}\}^t = x_{VMF}; \tag{8.41}$$

b) second filtering method: Vectors are ordered as follows:

$$\sum_{i=1}^{n} A(x_{BD}, x_i) \leqslant \sum_{i=1}^{n} A(x_j, x_i), \quad \forall j = 1, 2, \ldots, n, \tag{8.42}$$

where A is the angle among vectors x_i and x_j. The vector x_{BD} is found according to criterion of minimum deviation among vectors (minimum error estimated from angle location) that gives the Basic Vector Directional Process (BVDP). Finally, the video generalized vector directional filter (VGVDF) selects a set of the vectors, which present the minimum deviation with respect to other vectors [1]:

$$x_D = \Im\{x^{(1)}, x^{(2)}, \ldots, x^{(K)}\} = \Im\{VGVDF[x_1, x_2, \ldots, x_N]\}. \tag{8.43}$$

So, the VGVDF produces a set of vectors with similar directions and this set should be processed after by a magnitude algorithm $\Im$ to form an output vector (output color voxel). To enhance the characteristics, we propose to use an impulsive detector in the VGVDF. The detector noise value Val is determined in the form [22]

$$Val = \left\| x_{(N+1)/2} - \frac{1}{K} \sum_{i=1}^{K} x^{(i)} \right\| \tag{8.44}$$

as the absolute distance between the central sample $x_{(N+1)/2}$ and mean of the first K directionally ordered vectors $x^{(1)}, x^{(2)}, \ldots, x^{(K)}$ obtained in the VGVDF and associated with the smallest angles $\alpha_1 \leqslant \alpha_2 \leqslant \ldots \leqslant \alpha_K$, where α_j is found as $\sum_{i=1}^{n} A(x_j, x_i)$, $\forall j = 1, 2, \ldots, K$, for $K \leqslant n$.

The impulsive noise detector compares the value *Val* with a threshold *Tol* to determine the central voxel status and realizes the following processing procedure:

$$\text{IF} \quad Val \geqslant Tol \quad \text{THEN} \quad x_{(N+1)/2} \quad is\ impulse \quad \text{ELSE} \quad is\ noise - free \tag{8.45}$$

If this voxel is impulse, the magnitude processing algorithm (median filter) works; in the other case, the output voxel is not changed.

During the simulations, the color video sequences «Miss America» and «Flowers» were tested. They have been corrupted by impulsive noise of different intensity in each channel of a color frame.

The *PSNR, MAE,* and *NCD* criteria were applied to characterize the noise suppression, fine detail and edge preserving, and color chromaticity preservation in each algorithm.

Filters used for comparison were: Median Filter adapted to 3D processing, denoted as «*Video-MF*»; Video $a-$trimmed mean (*VATM*) filter; K-Nearest neighbor filter (*KNNF_1*), which was implemented using 3D window; *KNNF_2* [37], using the Euclidian distance to order the voxel values and Cross×Cross×Cross window to process the video sequence; video adaptive vector median filter (*VAVMF*) adapted to video processing. The adaptation in Vector median filter [4] was proposed to process the video sequences realizing Video-vector median filter (*VVMF*). The proposed Video vector directional K-nearest neighbor vector median filter (*VVDKNNVMF*) uses the first proposed filtering method. The parameter value $K = 7$ was chosen according to optimal *PSNR* values. The video adaptive vector directional median filters (*VAVDMF_1, VAVDMF_2* and *VAVDMFATM*) use the second proposed filtering method. Filter *VAVDMF_1* is applied to each frame in the video color sequence. The parameters $K = 5$ and $Tol = 20$ were chosen according to maximum of *PSNR*. Filter *VAVDMF_2* applies 3D window, and parameters $K = 5$ and $Tol = 18$ were found according to optimum of criterion *PSNR*. In both algorithms, if central voxel is noisy, the median filter processing was applied. Finally, the *VAVDMFATM* filter uses a 3D window to process the sequence, and parameters and were found according to maximum values of *PSNR* (if central voxel is noisy, the *VATM* filter is used). Parameters values K and Tol found to process the sequence «Flowers» were optimized according to criterion of maximum for *PSNR* value, the same parameters were also applied in processing of the sequence «Miss America» obtaining the best results too. Table 8.6 exposes the PSNR values for VATM and VAVDATM filters in case of «Miss America» frames corrupted by 15% impulsive noise.

Table 8.6. PSNR values for VVMF, VATM, and VAVDATM algorithms for «Miss America» frames with 15% of impulsive noise.

	Frame2	Frame10	Frame20	Frame30	Frame40
VVMF	35.11	33.61	34.14	34.49	33.87
VATM	35.28	33.90	34.44	34.74	33.97
VAVDATM	**36.22**	**34.94**	**35.37**	**35.69**	**34.93**
	Frame50	Frame60	Frame70	Frame80	Frame90
VVMF	34.70	34.75	33.30	33.73	34.50
VATM	34.95	35.05	33.51	34.17	34.76
VAVDATM	**35.92**	**36.06**	**34.45**	**35.14**	**35.73**
	Frame100	Frame110	Frame120	Frame130	Frame140
VVMF	32.71	34.55	34.90	33.98	34.84
VATM	33.08	34.80	35.12	34.22	35.04
VAVDATM	**33.82**	**35.78**	**36.30**	**35.22**	**36.08**

Analyzing Table 8.6 one can see that the best numerical results in 15% corruption is given by the novel VAVDATM filter, showing better noise suppression in medium percentages of corruption «Miss America» sequence during whole video sequence.

Similar results are presented in Table 8.7 for the MAE criterion.

Table 8.7. MAE values for VVMF, VATM, and VAVDATM algorithms for «Miss America» frames with 15% impulsive noise.

	Frame2	Frame10	Frame20	Frame30	Frame40
VVMF	2.27	2.96	2.61	2.50	2.86
VATM	2.38	2.99	2.71	2.63	2.92
VAVDATM	**1.46**	**1.83**	**1.66**	**1.61**	**1.81**
	Frame50	Frame60	Frame70	Frame80	Frame90
VVMF	2.42	2.50	3.11	2.82	2.60
VATM	2.53	2.57	3.13	2.86	2.68
VAVDATM	**1.54**	**1.54**	**1.97**	**1.76**	**1.65**
	Frame100	Frame110	Frame120	Frame130	Frame140
VVMF	3.21	2.47	2.53	2.93	2.48
VATM	3.20	2.55	2.60	2.95	2.57
VAVDATM	**2.14**	**1.57**	**1.51**	**1.81**	**1.53**

This table exposes MAE values and confirms that the best numerical results in the case of 15% corruption is given by the VAVDATM, which provided better detail and edge preservation in «Miss America» sequence during the whole video sequence.

Figure 8.11 demonstrates the NCD criterion measure, which characterizes color preservation property in a video color sequence «Miss America». The best algorithm according to this criterion is realized by novel VAVDATM filter practically for all levels of impulsive noise percentages.

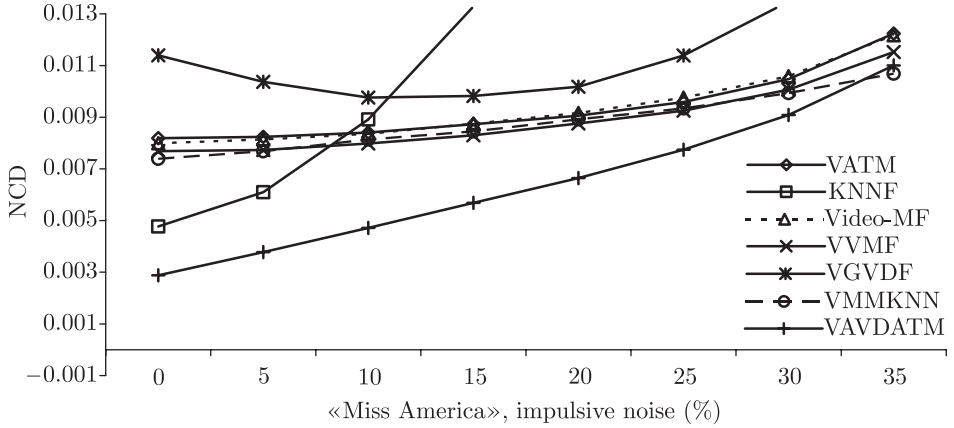
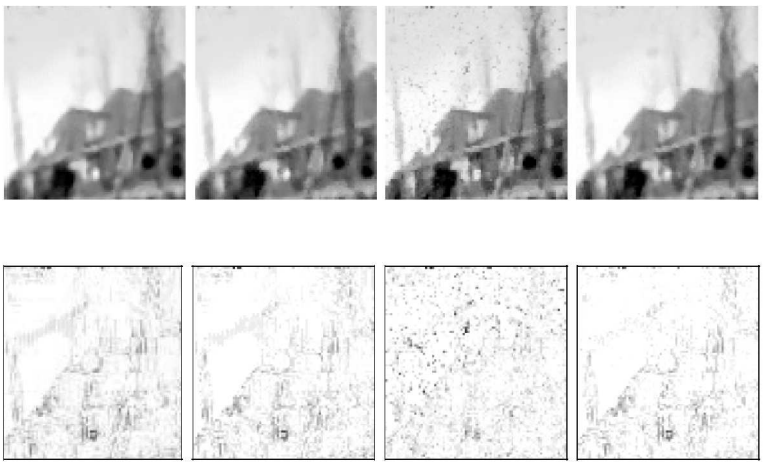


Fig. 8.11. NCD values for different percentages of impulsive noise in the case of «Miss America» frame.

Other implementations using optimized parameters to present better results were: *VVMF, Video-MF*, and *VATM* also achieved good performances with similar characteristics for all these algorithms. Analysis of the filtered images presented in Fig. 8.12



a *b* *c* *d*

Fig. 8.12. Filtered images in the case of 15% contamination by impulsive noise in the «Flowers» color video sequence.*a*) VATM filtered and error images; *b*) Proposed VVMF filtered and error images; *c*) KNNF_2 filtered and error images; *d*) Proposed VAVDATM filtered and error images.

demonstrates good visual subjective performance in preservation of the video color sequences using proposed filtering techniques.

8.5.4. Fuzzy Vectorial Techniques in 3D Filtering of Gaussian Noise. As it was discussed in the fifth chapter of the book, the additive noise can be modelled as Gaussian one. So, the problem consists of development of the framework that should be applied to suppress the influence of such noise. Another additional difficulty is the existence of small fine details and edges in the image or motions of the objects if near fames are being involved together in processing stage. The preprocessing stage in this case is to detect such motion or distinguish it on the background of plane areas in another part of image or frame. There exist numerous applications of the motion detection. It can be used for surveillance purposes, e.g., to monitor a room, in which there no motion is supposed, or the detection results can be useful as the input data for more advanced, higher level video processing techniques, such as the tracking of objects through time.

In our proposal, the motion detector combines the membership degree appropriately using defined fuzzy rules. The membership degree of motion for each pixel in a 3D sliding window is determined by the proposed membership function. Both fuzzy membership function and fuzzy rules are defined in such a way that the performance of the motion detector is optimized in terms of its robustness to a noise.

As it was explained, fuzzy image processing has three main stages: 1) image fuzzification, 2) modification of membership values, and 3) image defuzzification. Once the image data were transformed from pixel values plane to the membership values plane known as fuzzification, the fuzzy techniques should be employed that can be a fuzzy clustering, a fuzzy rule based approach, a fuzzy integration approach, etc.

There exist various techniques to detect pixel-by-pixel changes. The simplest is to subtract the color levels of successive frames, and to conclude that the pixel has changed comparing with some threshold.

Video denoising is usually realized by temporal processing only [37–39], or spatial temporal filtering [40–43]. Another approach consists in the use of the robust techniques designed in the 3D processing of video sequences. The main drawback of this approach

is that some of these algorithms have expensive requirements to hardware and software because of filtration up to three images at same time [44]. The approach proposed here applies adequate mathematical operations to consume less computing time, and are realized dividing different operations in accordance to proposed membership functions. Additional advantage of the filtering framework is that only two frames (past and present) to reduce the requirements in a complete processing system are used. Also, to confirm the effectiveness of the approach the comparison with other methods is made.

8.5.4.1. Architecture of Spatial-Temporal Filter.

Figure 8.13 exposes the proposed denoising scheme in spatial-temporal filtering. According to the figure, at the initial step, the histogram of the first frame is formed employing the angle deviations, and after that, the standard deviation (SD) is calculated. This value is used in temporal algorithm as an initial one, adapted during processing.

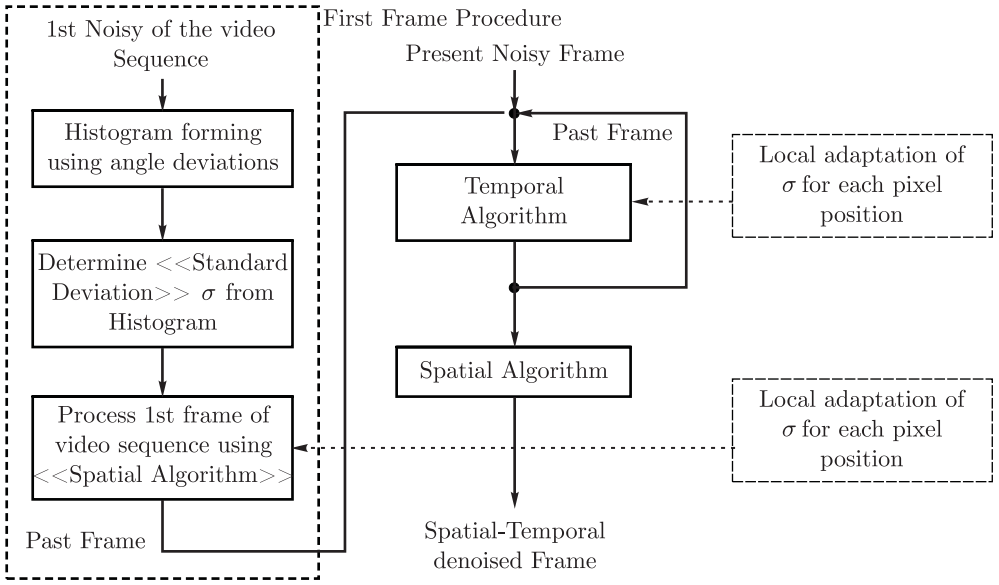


Fig. 8.13. Proposed denoising scheme based in spatial-temporal techniques

The filtered frame in such video sequence will be the *past frame* in the temporal algorithm. Here, only the *past* and *present frames* are employed to produce an output filtered image, which will be the *past frame* $(t-1)$ in a recursive manner of the *actual* one at time t, and, at the same time, it is processed in a final step by the spatial algorithm to result in the *Spatial-Temporal Denoised Frame*.

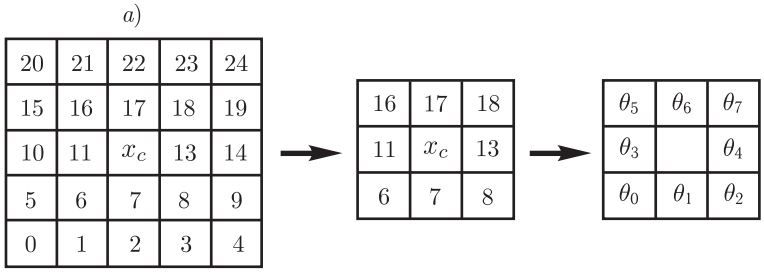
8.5.4.2. First Frame Procedure.

The procedure is divided into three steps: Histogram Calculation, Noise Estimation, and Spatial Algorithm Operations. The Past Frame, as the first frame, should be processed first.

The 3×3 window processing is used to calculate the mean value $\overline{x}_\beta$ ($\beta = Red$, $Green$, $Blue$ are the RGB planes of a color image). After that, an angle deviation (distance) among two vectors $\theta_c = A(\overline{x}_\beta, x_c)$ is calculated, where first vector is the formed mean value $\overline{x}_\beta$, and the other one is the central pixel of the sample. Due to color image representation, the angle can be in the interval $\pi/2$. The procedure of noise estimator is as follows: $\theta_c \leqslant \pi/2$, *then the histogram is increased by «1», otherwise is «0»*. Once obtaining the Histogram, the probabilities of occurrence of each of the values;

the median value of the probabilities $\mu = \sum_j jp_j$, the variance $\sigma_\beta^2 = \sum_j (j-\mu)^2 p_j$, where, $j = 0, \ldots, 255$, and SD $\sigma_\beta = \sqrt{\sigma_\beta^2}$, are to be calculated. The latter parameter is used in the noise estimator and denoising procedure at the first step.

Having the value of noise intensity, the processing of the first frame of the video sequence is run. Two kinds of windows processing that are shown in Figure 8.14 are employed in the proposed procedure.



a) 5×5 Window Processing

b) 3×3 Window Preprocessing

Fig. 8.14. Windows processing used by Spatial Filter.

Let denote by $\theta_i = A(x_i, x_c)$ the angle deviation of x_i with respect to x_c, where $i = 0, 1, 2, \ldots, N-1$, $i \neq c$, $N = 8$ (3×3 window), and C = *central pixel*. Let the angles be numbered as shown in Fig. 8.14 b, which can be helpful in ordering when uniform areas should be detected, so the following considerations can be applied:

IF (θ_1 AND θ_3 AND θ_4 AND $\theta_6 \geqslant \tau_1$)
OR (θ_0 AND θ_2 AND θ_5 AND $\theta_7 \geqslant \tau_1$)
THEN «*Mean Weighted Filtering Algorithm*»,
ELSE «*Spatial Filtering Algorithm*»,

where τ_1 is a threshold defined as 0.1. The «AND» operation is defined as «Logical AND», the «OR» operation is «Logical OR» [13, 45].

8.5.4.3. Mean Weighted Filter.

This filter is defined by next expression:

$$y_{\beta out} = \frac{\left[\sum\limits_{i=0, i\neq c}^{N-1} x_{\beta i}\left(\frac{2}{1+\exp(\theta_i)}\right) + x_{\beta c}\right]}{\left[\sum\limits_{i=0, i\neq c}^{N-1}\left(\frac{2}{1+\exp(\theta_i)}\right) + 1\right]}, \quad N = 8 \tag{8.46}$$

and realizes a fast processing in the case of Gaussian noise, weighting the central pixel with the highest value, so it smoothes sufficiently the uniform regions. If it is not possible to filter the samples applying Mean Weighted Algorithm, the Spatial Filter should be used, providing good reference values before the temporal filtering.

8.5.4.4. Spatial Filter.

Here, the procedure is separated in each a color plane to process them independently, obtaining values σ_β. The values σ_β are used later in the special procedure, where the

Standard Deviations are adapted locally. Operations to realize this point are developed as follows:

1. Calculate the probability of each a sample inside the 5×5 window processing (Fig. 8.14a), and calculate a mean value for this window $\overline{x}_{\beta 5\times 5}$.
2. Use a mean value $\overline{x}_{\beta 5\times 5}$ for obtaining the variance for 5×5 samples, and finally, to calculate the local SD in each a plane σ_β ($\beta = R, G, B$).
3. Use the SD obtained in entire frame and for each sample of this frame and compare them. *If* $\sigma_\beta < \sigma'_\beta$, *then* $\sigma_\beta = \sigma'_\beta$, *otherwise* $\sigma'_\beta = \sigma_\beta$.

A threshold value according to better experimental values of *PSNR* and *MAE* criteria was chosen as $T_\beta = 2\sigma_\beta$.

The increase in the performance of Spatial Filtering Algorithm can be achieved using the fuzzy values that represent relationship among the pixels to clarify the noise presence in a given sample.

For each pixel $x(i, j)$ of a sample taken, we find eight neighbours, which corresponds to any of the directions: N = North, E = East, S = South, W = West, NW = North–West, NE = North–East, SE = South–East, SW = South–West [17].

Let $A_\beta(i, j)$ be a given plane of the input noisy image for any channel $\beta = (R, G, B)$. Then the gradient in this plane can be defined as:

$$\nabla_{(k,l)} = |A_\beta(i + k, j + l) - A_\beta(i, j)|, \quad k, l \in \{-1, 0, 1\},$$

where the pair (k, l) corresponds to one of the eight directions and these gradients are called «*main gradient values*» and the point (i, j) is called «*the centre of the gradient values*». To avoid blur in presence of an edge, it has been proposed to use additionally two «*derived gradient values*». These three gradient values for a certain direction are finally connected together into one single general value called «*fuzzy gradient value*». The two derived gradient values in the same direction as the main gradient, are determined by its centres making a right angle with the direction of the corresponding main gradient. This procedure is illustrated in Fig. 8.15 [17].

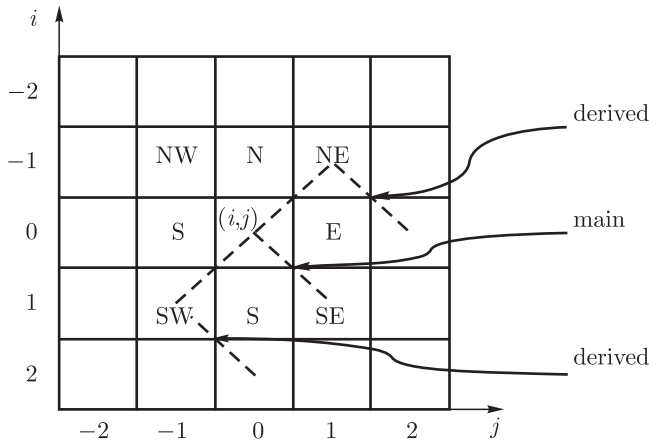


Fig. 8.15. Main and derived directions to Vectorial and Gradient values.

Using Fig. 8.15 and Table 8.8 one should define the following variable $\gamma = NW, N, NE, E, SE, S, SW, W$. Now, the pixels should be treated as the vectors to employ the directional processing. Applying the gradient and vectorial values, we can obtain «*fuzzy vectorial gradient values*» that should be defined by Fuzzy Rules.

Table 8.8. Involved Vectorial-Gradient Directions to Calculate the Fuzzy Vectorial.

Direction	Main Vectorial-Gradient Involved	Derived Vectorial-Gradient Involved
NW	(i,j), $(i-1,j-1)$	$(i+1,j-1)$, $(i-1,j+1)$
N	(i,j), $(i-1,1)$	$(i,j-1)$, $(i,j+1)$
NE	(i,j), $(i-1,j+1)$	$(i-1,j-1)$, $(i+1,j+1)$
W	(i,j), $(i,j-1)$	$(i-1,j)$, $(i+1,j)$
E	(i,j), $(i,j+1)$	$(i-1,j)$, $(i+1,j)$
SW	(i,j), $(i+1,j-1)$	$(i-1,j-1)$, $(i+1,j+1)$

To obtain *«Main Vectorial Gradients»* the gradient values found before to perform a vectorial process are taken into account. The following scheme to fulfil this consideration is proposed:

if $\nabla_{\gamma\beta} < T_\beta$ for each one of the three gradient values, then the angle deviation in γ's directions for three vectorial values involved is calculated.

Finding an angle deviation for each image channel, it is easy to calculate the weight values in the direction related. To compute *«Main and Derived Vectorial Gradients»*, the next equation is used [2]:

$$\alpha_{\gamma\beta} = \frac{2}{1+\exp\left[\cos^{-1}\left\{\dfrac{2\cdot(255)^2 + x_{\gamma\beta}x'_{\gamma\beta}}{\sqrt{2\cdot(255)^2 + x_{\gamma\beta}^2}\cdot\sqrt{2\cdot(255)^2 + {x'_{\gamma\beta}}^2}}\right\}\right]}, \tag{8.47}$$

where the angle deviations for each color image channel by each pixel involved in each sample are taken. Now, taking into account these values and gradients in the related direction, the membership function is used to find *«Main and Derived Vectorial-Gradient Value»*.

Membership function to obtain *«Main and Derived Vectorial-Gradient Value»* is defined as in [17]:

$$\mu_{BIG} = \begin{cases} \max\{x,y\}, & \text{if} \quad \nabla_{\gamma\beta} < T_\beta, \\ 0, & \text{otherwise}, \end{cases} \tag{8.48}$$

where

$$x = \alpha_{\gamma(M,D1,D2)\beta}, \quad y = 1 - \left[\nabla_{\gamma(M,D1,D2)\beta}/T_\beta\right],$$

and M = Main value, $D1$ = Derived 1 value, and $D2$ = Derived 2 value. Figure 4.14 and Table 8.8 expose the pixels involved by each one of the directions. Finally, the process to obtain *«Fuzzy Vectorial Gradient Values»* is defined by a Fuzzy Rule connecting Gradients with Vectorial values.

Fuzzy Rule 1: *Fuzzy Vectorial Gradient value* is defined as $\nabla_{\gamma\beta}\alpha_{\gamma\beta}$, as follows:

IF ($\nabla_{\gamma\beta M}$ is BIG AND $\nabla_{\gamma\beta D1}$ is BIG)
OR ($\nabla_{\gamma\beta M}$ is BIG AND $\nabla_{\gamma\beta D2}$ is BIG)
THEN $\nabla_{\gamma\beta}\alpha_{\gamma\beta}$ is true.

Here, AND is defined as a fuzzy intersection and is expressed by an algebraic product $A \cdot B$ [45]; OR is defined as a fuzzy union and is expressed as an algebraic sum $A + B - A \cdot B$ [45].

According to this fuzzy rule, one can see that if Main and Derived Vectorial-Gradients are close enough to each other in absolute difference and in angle deviations, the pixels have similarity with respect to the central pixel and it means that they

can be taken into a current processing sample to suppress a noise. The found Fuzzy Vectorial-Gradient values permit to manipulate with weights given to the pixels in direction related minimizing computational charge in a weighted mean filter. On the other hand, this permits to avoid pixels that have not sufficient similarity to central one, giving them a weight with a minimum value. Final step in spatial filtering of noise is realized employing a *Weighted Mean procedure:*

$$y_{\beta out} = \frac{\sum_{\gamma} \omega_\gamma x_{\gamma\beta}}{\sum_{\gamma} \omega_\gamma}, \tag{8.49}$$

where $\omega_\gamma = \nabla_{\gamma\beta}\alpha_{\gamma\beta}$, and $x_{\gamma\beta}$ represents each pixel used inside the pre-processing window.

Spatial Algorithm discussed before suppresses Gaussian noise efficiently, but can smooth fine details and edges. To avoid this drawback, a «Temporal Algorithm» is proposed providing good preservation of the mentioned image characteristics. This algorithm permits to realize motion detection in past and present frames of video sequence. Connection between these algorithms is discussed in the next section.

8.5.4.5. Temporal-Spatial Filtering.

Here, only the past and present frames are processed together to avoid dramatic charge in memory requirement and time processing. The proposed fuzzy logic rules are used in each a color plane of two frames in an independent manner. Also, a 3×3 pre-processing window is employed to calculate parameters needed in the algorithm.

The angle deviations and gradient values related to the central pixel in the present B frame with respect to its neighbours from past frame A are found as

$$\theta_i^1 = A(x_i^A, x_c^B), \quad \nabla_i^1 = \left|x_i^A - x_c^B\right|; \quad i = 0, 1, \ldots, N-1; \; N = 8, \tag{8.50}$$

$$\begin{aligned} \theta_i^2 &= A(x_i^A, x_i^B), \quad \nabla_i^2 = \left|x_i^A - x_i^B\right|, \\ \theta_i^3 &= A(x_i^B, x_c^B), \quad \nabla_i^3 = \left|x_i^B - x_c^B\right|, \end{aligned} \tag{8.51}$$

where x_c^B is the central pixel in present frame. The angle and gradient values for the corresponding pixel positions in both frames are calculated. Also, only the same parameters for the present frame are computed, finally, eliminating operations in the past frame, as it is illustrated in Figs. 8.16 a, b, c.

Let us define the membership functions to obtain a value that indicates the degree in which a certain gradient value or vectorial value matches the predicate. If a gradient or a vectorial value have membership degree one for the fuzzy set SMALL, it means that it is SMALL for sure in this fuzzy set, and no movement is achieved by the pixel related in the sample taken.

Selection of this kind of membership functions is done due to nature of the pixels, where a movement is not a linear response, and a pixel has different meanings in each scene of the video sequence. Examples of the used membership functions are illustrated in Fig. 8.17.

Membership functions BIG and SMALL for angles and gradients are defined by the following expressions [45]:

$$\mu_{SMALL}(\chi) = \begin{cases} 1 & \text{if} \quad \chi < \mu_1, \\ \exp\left\{-((\chi - \mu_1)^2/(2\sigma^2))\right\} & \text{otherwise}, \end{cases} \tag{8.52}$$

$$\mu_{BIG}(\chi) = \begin{cases} 1 & \text{if} \quad \chi > \mu_2, \\ \exp\left\{-((\chi - \mu_2)^2(2\sigma^2))\right\} & \text{otherwise}, \end{cases} \tag{8.53}$$

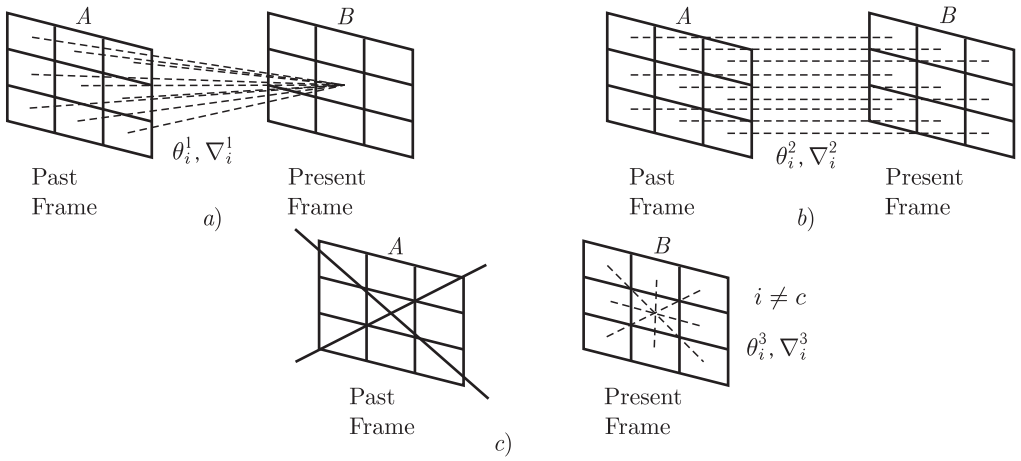


Fig. 8.16. Procedures to find angles and gradient values.

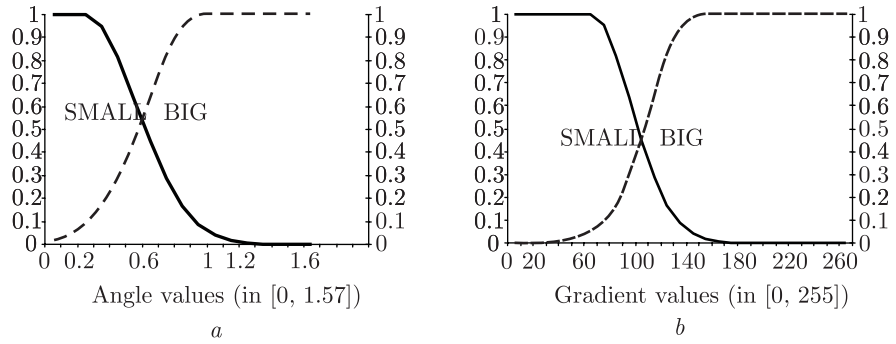


Fig. 8.17. Membership functions SMALL and BIG for angle and gradients deviations.

where $\chi = \theta, \nabla$ with parameters $\mu_1 = \varphi_1$, φ_2, $\mu_2 = \varphi_3$, φ_4, $\varphi_1 = 0.2$, $\varphi_2 = 60$, $\varphi_3 = 0.9$, and $\varphi_4 = 140$, using $\sigma^2 = 0.1$ for $\varphi 1$ and $\varphi 3$, and using $\sigma^2 = 1000$ for $\varphi 2$ and $\varphi 4$. It implies that BIG means *the movement probability* and SMALL means *no movement probability*. So, if a value of «0» is at the fuzzy set BIG is under its membership function, it means *no movement* for sure and vice-versa, and if a value of «0» is at the fuzzy set SMALL under its membership function, it means *movement* for sure.

We have designed the fuzzy rules to detect the presence of movement pixel by pixel. Firstly, let detect movement with respect to central pixel in the present frame with the pixels in the past frame; secondly, the movement detection in respect to pixel by pixel in both positions of the frames is realized, and finally, this procedure is only applied in the present frame using central pixel and its neighbours. These three movement values contribute in a parameter that characterizes the movement confidence. Fuzzy rules are illustrated in Fig. 8.18.

Fuzzy Rule 2: Definition of the Fuzzy Vectorial-Gradient value *SBB(x,y,t)*:

IF $\theta^1(x,y,t)$ is SMALL AND $\theta^2(x,y,t)$ is BIG
AND $\theta^3(x,y,t)$ is BIG AND $\nabla^1(x,y,t)$ is SMALL
AND $\nabla^2(x,y,t)$ is BIG AND $\nabla^3(x,y,t)$ is BIG

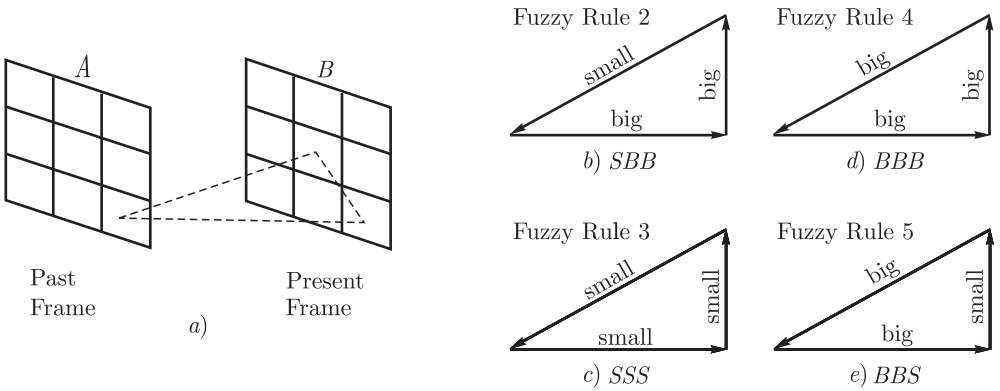


Fig. 8.18. Fuzzy Rules 2, 3, 4, and 5 used to determine movement confidence employing the past and present frames.

THEN $SBB(x,y,t)$ is true (Fig 8.18 b).

Fuzzy Rule 3: Definition of the fuzzy Vectorial-Gradient value *SSS(x,y,t)*:

IF $\theta^1(x,y,t)$ is SMALL AND $\theta^2(x,y,t)$ is SMALL
AND $\theta^3(x,y,t)$ is SMALL AND $\nabla^1(x,y,t)$ is SMALL
AND $\nabla^2(x,y,t)$ is SMALL AND $\nabla^3(x,y,t)$ is SMALL
THEN $SSS(x,y,t)$ is true (Fig 8.18 c).

Fuzzy Rule 4: Definition of the fuzzy Vectorial-Gradient value *BBB(x,y,t)*:

IF $\theta^1(x,y,t)$ is BIG AND $\theta^2(x,y,t)$ is BIG
AND $\theta^3(x,y,t)$ is BIG AND $\nabla^1(x,y,t)$ is BIG
AND $\nabla^2(x,y,t)$ is BIG AND $\nabla^3(x,y,t)$ is BIG
THEN $BBB(x,y,t)$ is true (Fig 8.18 d).

Fuzzy Rule 5: Definition of the fuzzy Vectorial-Gradient value *BBS(x,y,t):*

IF $\theta^1(x,y,t)$ is BIG AND $\theta^2(x,y,t)$ is BIG
AND $\theta^3(x,y,t)$ is SMALL AND $\nabla^1(x,y,t)$ is BIG
AND $\nabla^2(x,y,t)$ is BIG AND $\nabla^3(x,y,t)$ is SMALL
THEN $BBS(x,y,t)$ is true (Fig 8.18 e).

To reduce the execution time of the algorithm, the processing effort should be reduced. The idea is to distinguish the different areas, especially to find some regions that could be processed by magnitude filter without affecting fine image details. Here, the SD of the sample that includes the $3 \times 3 \times 2$ window for each color channel in the past and present frames is calculated, obtaining the parameter σ''_β. This procedure is similar to the procedure used before. After that, we compare it with standard deviation σ'_β obtained for Spatial Filter in the following way:

IF $\{(\sigma''_{red} \geqslant 0.4 * \sigma'_{red})$ *AND* $(\sigma''_{green} \geqslant 0.4 * \sigma'_{green})$ *AND* $(\sigma''_{blue} \geqslant 0.4*)\}$, *THEN* Fuzzy Rules 2, 3, 4, and 5 should be employed, *OTHERWISE* a Mean Filter is performed.

Here, the AND operation is the «Logical AND».

If a Mean Filter Algorithm is applied,

$$\overline{y}_{\beta out} = \sum_{i=0}^{F} x_{\beta i}/F, \quad F = 17, \tag{8.54}$$

it means the presence of a uniform region. In equation (8.54), $x_{\beta i}$ marks each of the pixels in a $3 \times 3 \times 2$ window pre-processing. Parameter $\alpha = 0.125$ is used to adapt standard deviation and control the filtering charge.

So, if drastic changes happen in some samples, they are reflected in their fuzzy vectorial-gradient values.

The following expressions are used to update the SD for next frames and distinguish the fine details on the background of uniform regions for each Fuzzy Rule:

$$\sigma'_{\beta} = (\alpha\,(\sigma_{total}/5)) + (1-\alpha)(\sigma'_{\beta}), \tag{8.55}$$

where σ_{total} is defined as $\sigma_{total} = (\sigma''_{red} + \sigma''_{green} + \sigma''_{blue})/3$.

If the number of pixels with $SBB(x,y,t)$ value is the biggest one among others calculated, the algorithm can be expressed as the procedure

IF $\{(SBB(x,y,t) > SSS(x,y,t))$ AND $(SBB(x,y,t) > BBB(x,y,t))$ AND $(SBB(x,y,t) > BBS(x,y,t))\}$, THEN Weighted Mean Algorithm is performed using the found $SBB(x,y,t)$ values as weights:

$$y_{\beta out} = \sum_{i=1}^{\#pixels} p^{A}_{\beta i} \cdot SBB_i \Big/ \sum_{i=1}^{\#pixels} SBB_i. \tag{8.56}$$

For this case, the SD adaptation of the sample for the $SBB(x,y,t)$ fuzzy value is equal to 0.875. The fuzzy $SBB(x,y,t)$ value shows that a central pixel is possibly in movement because of big differences in corresponding values.

If the number of pixels with $SSS(x,y,t)$ value is the biggest, the algorithm can be expressed as the procedure

IF $\{(SSS(x,y,t) > SBB(x,y,t))$ AND $(SSS(x,y,t) > BBB(x,y,t))$ AND $(SSS(x,y,t) > BBS(x,y,t))\}$, THEN Weighted Mean Algorithm is performed using the $SSS(x,y,t)$ values as weights:

$$y_{\beta out} = \sum_{i=1}^{\#pixels} 0.5(p^{A}_{\beta i} + p^{B}_{\beta i}) \cdot SSS_i \Big/ \sum_{i=1}^{\#pixels} SSS_i, \tag{8.57}$$

and α for the $SSS(x,y,t)$ fuzzy value is equal to 0.125. The value $SSS(x,y,t)$ shows that a central pixel is not in movement because of small differences in all directions. This permits to use in equation (4.57) all the pixels in both frames.

If the number of pixels with $BBS(x,y,t)$ value is the biggest, the following algorithm is applied:

IF $\{(BBS(x,y,t) > SBB(x,y,t))$ AND $(BBS(x,y,t) > SSS(x,y,t))$ AND $(BBS(x,y,t) > BBB(x,y,t))\}$, THEN Weighted Mean Algorithm is performed using the $BBS(x,y,t)$ values as weights:

$$y_{\beta out} = \sum_{i=1}^{pixels} p^{B}_{\beta i} \cdot (1 - BBS_i \Big/ \sum_{i=1}^{pixels} (1 - BBS_i), \tag{8.58}$$

where α for the $BBS(x,y,t)$ fuzzy value is equal to 0.875. The $(1 - BBS(x,y,t))$ value is taken because the interest is in obtaining a significant value in how SMALL membership degree has the pixel in present frame, omitting membership degree value

due to past frame pixel. The $BBS(x,y,t)$ value exposes that the central pixel is nearly related with its neighbours in the present frame only, but not in the past frame. The probable reason can be due to big movement or residual noise presence in the past frame.

In equations given above, $p^A(x,y,t)$ and $p^B(x,y,t)$ represent each pixel in the past and present frames that satisfy the IF condition, and $y_{\beta out}$ is the output in spatial and temporal filtering.

The number of pixels with $BBB(x,y,t)$ value is the biggest when the majority of pixels are not related in any way with other neighbourhood pixels. So, it has been decided to realize the following procedure:

IF $\{(BBB(x,y,t) > SBB(x,y,t))$ AND $(BBB(x,y,t) > SSS(x,y,t))$ AND $(BBB(x,y,t) > BBS(x,y,t))\}$, THEN there exist motion or noise for sure.

To investigate this condition, we consider the nine Fuzzy Vectorial-Gradient values obtained from $BBB(x,y,t)$. The central value is selected and at least three more fuzzy neighbours values to detect movement present in the sample. We use the Fuzzy Rule «R» to obtain «*motion_noise*» confidence. The activation degree of «R» is just the conjunction of the four subfacts, which are combined by a chosen triangular norm defined as A AND $B = A * B$ [45]. Computations are specifically the intersection of all possible combinations of $BBB(x,y,t)$ and three different neighboring BIG membership degrees $BBB(x+i, y+j, t)$, $(i,j = -1,0,1)$. This gives 56 values obtained using triangular norm. The values are added using algebraic sum of all instances to obtain the *motion_noise* confidence. Algebraic sum is given by $A + B - A * B$.

If $(motion\_noise) = 1$, then $\alpha = 0.875$, else if $(motion\_noise) = 0$, then $\alpha = 0.125$, otherwise $\alpha = 0.5$.

Using these values we obtain output pixel as follows:

$$y_{\beta out} = (1-\alpha)p^B_{\beta c} + \alpha p^A_{\beta c}, \tag{8.59}$$

where $p^A_{\beta i}$ and $p^B_{\beta i}$ determine each pixel in the past and present frame of a sequence. The $BBB(x,y,t)$ value shows that a central pixel and its neighbours do not have relation among them, and it is highly probably that this pixel is either in motion or is a noisy pixel.

If there is no majority in pixels calculated by any Fuzzy Rule, it can be concluded that sample values in the past and present frames have similar nature. So, using only the central pixels from the present and past frames, we can obtain an output pixel

$$y_{\beta out} = 0.5p^B_{\beta c} + 0.5p^A_{\beta c}, \tag{8.60}$$

where $\alpha = 0.5$ is used to update standard deviation.

At the final step, the algorithm employs the Spatial Filter for smoothing the non-stationary noise left after the preceding temporal filter. This can be done by a local spatial algorithm, which adapts to image structures and noise levels in the corresponding spatial neighbourhood. This algorithm needs the only modification in its threshold value $T_\beta = 0.25\sigma'_\beta$ because of the adaptive way used for the SD.

8.5.4.6. Criteria in Filtering of the Images

We evaluate the proposed processing employing motion detection, using different objective and subjective criteria. The definition of these criteria: *PSNR*, *MSE*, *MAE*, and *NCD* has been done in the Chapter 5 (see equations. (5.5)–(5.8)). Here, additionally we use the *Mean Chromaticity Error* (*MCRE*) that is a measure characterizing the color

chromaticity by the error of chromaticity between two color images [2, 22]:

$$MCRE = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} C\left[f(i,j), \widehat{f}(i,j)\right] \Big/ MN, \tag{8.61}$$

where $f(i,j)$, and $\widehat{f}(i,j)$ are original and filtered image vectors estimated in (i,j) pixel position, $C\left[f(i,j), \widehat{f}(i,j)\right]$ is the chromaticity error between two vectors, which is defined as the $P\widehat{P}$ distance among two points P and $\widehat{P}$ that are the intersection points of $f(i,j)$ and $\widehat{f}(i,j)$ with the Maxwell triangle, respectively [2].

We also use a subjective visual criterion presenting the filtered images and/or their error images for implemented better filters to compare the capabilities of noise suppression and detail preservation for the algorithms. So, subjective visual comparison of the images provides information about the spatial distortion and artifacts introduced by different filters, as well as the noise suppression quality of the algorithm and present performance of the filter, when filtering images are observed by a human visual system.

8.5.4.7. Simulation Results.

Two different video sequences were used to qualify effectiveness of the proposed approach and compare it with known techniques. Both «Miss America» and «Flowers» sequences present different texture characteristics to provide a better understanding in the robustness of the proposed algorithm. Video sequences were contaminated with different Gaussian noise of levels from 0.0 to 0.05 in its variance. Frames in a QCIF format (176×144 pixels) are treated in *RGB* color space with 24 bits (true color), 8 bits for each channel, and 100 frames for each video sequence. The filtered frames were evaluated according to the *PSNR*, *MAE*, *NCD*, *NMSE*, and *MCRE* criteria to support performance of the proposed framework.

The proposed Fuzzy Directional Adaptive Recursive Temporal Filtering for Gaussian noise named as *FDARTF_G* was compared with another similar one, the *FMRSTF* (Fuzzy Motion Recursive Spatial-Temporal Filtering) [40], which only employs the gradients. Another reference procedure chosen is the *FVMRSTF* (Fuzzy Vectorial Motion Recursive Spatial-Temporal Filtering), which presents some modification of FMRSTF, combining the gradients and angles in processing. Other two filters chosen as reference in simulation to evaluate the quality of the proposed approach were «Generalized Vector Directional Processing» [41–44] adapted to process three frames at a time and the «Median M-type K-Nearest Neighbour» (*MM-KNN*) filter to remove impulsive noise from corrupted images. The latter was designed by us (see sec. 8.5.1) to remove impulsive noise from grey and color images in 2D and 3D [9–11, 22] and proved its good efficiency in comparison with other filtering procedures in suppression of additive noise in grey images [9, 22]. Here, it was adapted to process three color frames corrupted with additive noise at a time.

One can see in Table 8.9 that the proposed algorithm *VGVDF_G* in filtering the «Flowers» sequence can effectively suppress Gaussian noise of low intensity and is the best in the PSNR measure until the variance is less than 0.03. According to the MAE criterion that characterizes the preservation of fine details, the proposed approach exposes the best results for low intensity noise, 0.005 in variance, but presents acceptable results for other levels of corruption. According to this criterion, another proposed algorithm (VMMKNN1_2_G) is the best in preservation of fine image details. Analyzing objective criteria, the NMSE and MCRE, we observe that the proposed VGVDF_G presents a better performance until the variance is less than 0.03. The NCD criterion, which presents the chromaticity properties, shows that the proposed approach exposes the best results for low intensity noise, 0.005 in variance, and

presents acceptable results for other levels of corruption, where the best performance is demonstrated by the *VMMKNN1_2_G*.

Analyzing «Miss America» sequence (portrait type sequence) in comparison with «Flowers» sequence presenting «noisy» type sequence with large areas occupied by small fine details objects, one can see that the best results in all subjective criteria are demonstrated by the proposed VGVDF_G filter.

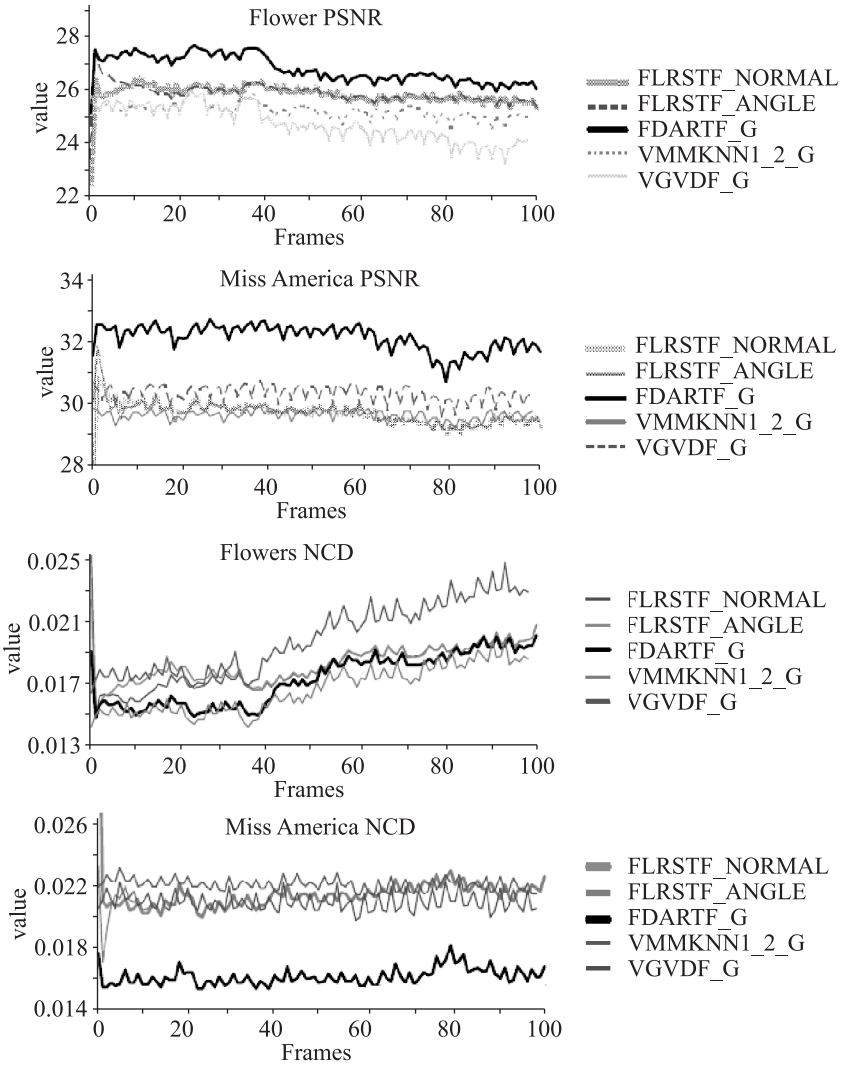


Fig. 8.19. PSNR and NCD criteria applied to process first 100 frames of Flowers and Miss America color sequences corrupted by Gaussian Noise of variance 0.005.

Other simulation results presented in Fig. 8.19 have numerically justified the robustness of the proposed algorithm exposing the filtering results over the first 100 frames of two sequences. It is easy to see that Flowers sequence corrupted by low-intensity Gaussian noise being processed by novel technique has presented the best

Table 8.9. Numerical results for different objective criteria characterizing the proposed algorithm and reference filters.

Criteria	Flowers Frame 20, Gaussian noise with variance = 0.005				
	FLRSTF_ NORMAL	FLRSTF_ ANGLE	FDARTF_G	VMMKNN 1_2_G	VGVDF_G
PSNR	26.19	26.01	**27.31**	25.35	25.46
MAE	9.63	9.83	**8.50**	8.78	8.96
MCRE	0.019	0.018	**0.014**	0.024	0.021
NCD	0.016	0.017	**0.015**	**0.015**	0.017
	variance = 0.01				
PSNR	24.36	24.34	**25.72**	24.63	24.72
MAE	11.93	11.97	10.44	**9.92**	10.15
MCRE	0.023	0.023	**0.017**	0.026	0.024
NCD	0.021	0.021	0.019	**0.017**	0.019
	variance = 0.02				
PSNR	22.60	22.57	**23.75**	23.42	23.627
MAE	14.645	14.694	13.182	**11.861**	11.912
MCRE	0.0284	0.0285	**0.0228**	0.0322	0.0276
NCD	0.0251	0.0252	0.0234	**0.0198**	0.022
	variance = 0.03				
PSNR	21.468	21.465	22.702	22.523	**22.794**
MAE	16.684	16.698	14.853	13.351	**13.316**
MCRE	0.0326	0.0324	**0.0258**	0.0362	0.0308
NCD	0.0285	0.0287	0.026	**0.0217**	0.0241
	Miss America Frame 20, Gaussian noise with variance = 0.005				
PSNR	29.93	29.91	**32.51**	29.80	30.66
MAE	5.82	5.83	**4.46**	6.18	5.55
MCER	0.035	0.035	**0.023**	0.031	0.025
NCD	0.02	0.02	**0.016**	0.021	0.02
	variance = 0.01				
PSNR	27.67	27.68	**30.06**	27.61	28.66
MAE	7.477	7.5	**6.069**	8.143	7.213
MCRE	0.045	0.045	**0.032**	0.043	0.032
NCD	0.026	0.026	**0.021**	0.028	0.026
	variance = 0.02				
PSNR	25.492	25.507	**27.251**	24.950	25.874
MAE	9.634	9.645	**8.376**	11.278	10.19
MCRE	0.057	0.057	**0.044**	0.064	0.045
NCD	0.033	0.033	**0.03**	0.039	0.037
	variance = 0.03				
PSNR	24.183	24.174	**26.024**	23.238	24.236
MAE	11.145	11.167	**9.603**	13.929	12.404
MCRE	0.064	0.064	**0.048**	0.083	0.054
NCD	0.0384	0.03843	**0.035**	0.049	0.045

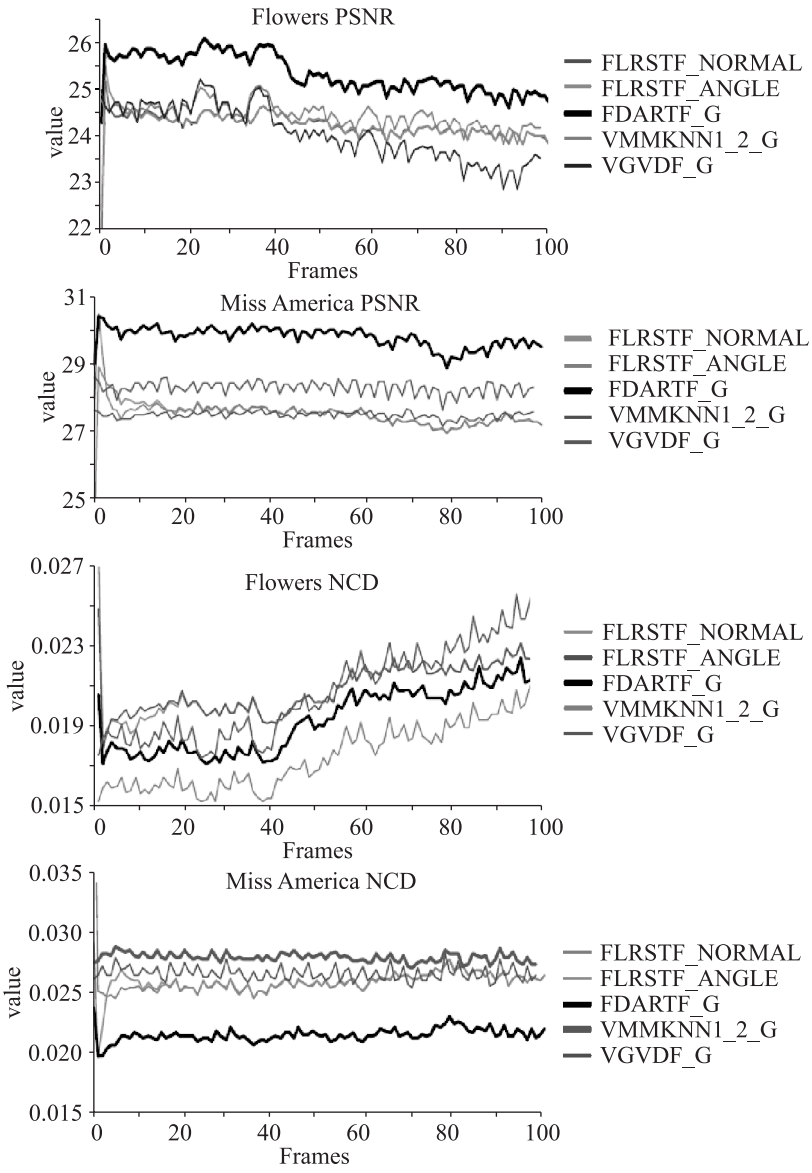


Fig. 8.20. PSNR and NCD criteria applied to process first 100 frames of Flowers and Miss America color sequences corrupted by Gaussian Noise of variance 0.01.

PSNR values in the majority of the frames. In the NCD criterion, the best results in the majority of the frames are exposed by VMMKNN1_2_G. (Fig. 8.19) has confirmed that, for Miss America sequence, the best results in all criteria are presented by the proposed filtering technique.

Similarly to Fig. 8.19, other simulation results are presented in Fig. 8.20, where the sequences were corrupted by a Gaussian noise of variance 0.01. Fig. 8.20 shows that the Flowers sequence processed by the designed procedure has a good performance in

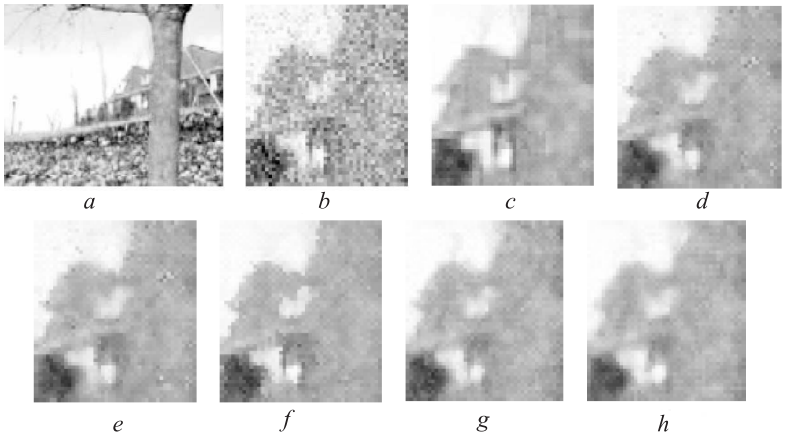


Fig. 8.21. Sequence Flowers, frame 50; a) Original image, b) Corrupted by Gaussian noise with 0.01 variance. c) Zoomed section. d) FLRSTF_NORMAL e) FLRSTF_ANGLE, f) FDARTF_G, g) VMMKNN1_2_G, h) VGVDF_G.

the PSNR criterion; on the contrary, the NCD and MAE criteria demonstrate a better performance obtained by another designed VMMKNN1_2_G filter. In the case of Miss America sequence filtering, the best results are demonstrated by the proposed algorithm, as it is clearly shown in the same figure.

Visual results exposing filtered images are shown at Fig. 8.21, where the images are corrupted by Gaussian noise with 0.01 in variance. It is easy to see the better preservation of image details when the proposed technique is applied. For example, one can clearly see good noise suppression observing the tree in Flowers sequence, and better fine details preservation in other parts of the image.

Figure 8.22 presents filtered visual results for Miss America sequence in the case of Gaussian noise corruption. Here, we can a see a better preservation of the image details when the proposed algorithm is applied. Also, one can see a better noise suppression,

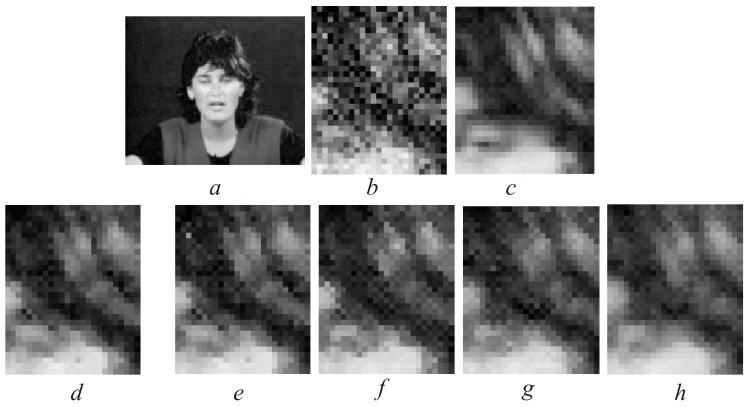


Fig. 8.22. Miss America, frame 50; a) Original image, b) Corrupted by Gaussian noise with 0.01 variance, c) Zoomed section, d) FLRSTF_NORMAL, e) FLRSTF_ANGLE, f) FDARTF_G, g) VMMKNN1_2_G, h) VGVDF_G.

observing uniform regions of the frame, and a better fine details preservation, analyzing the eye in this image.

So, the developed novel robust algorithm applies the fuzzy and directional techniques to suppress Gaussian noise in color video sequences and special motion detection scheme to characterize motion presented in two frames (past and present), as well as to determine filtering effort, adapting it according to noise presence and movement. The pixels in present and past frames or only pixels in present frame based on a decision in calculated values are used. The simulation results have demonstrated that, using gradient and vector values jointly, according to proposed algorithm, it is possible to improve its performance in comparison with the case when only one of these characteristics is used. This consideration has been proved in simulation results obtained by VGVDF_G, FLRSTF_NORMAL, and FLRSTF_ANGULO. The PSNR, MAE and NCD objective criteria clearly quantify the performance and robustness of the proposed framework in such characteristics as noise suppression, fine details preservation, chromaticity error, and perceptual error. The detailed analysis of filtering results over the hundred frames of the sequences has demonstrated the robustness of the proposed approach [46, 47, 48].

References

1. *Trahanias, P.E. and Venetsanopoulos A.N.* Directional Processing of Color Images: Theory and Experimental Results. IEEE Transactions on Image Processing, 1996, vol. 5, no. 6, pp. 868–880.
2. *Plataniotis, K.N. and Venetsanopoulos, A.N.* Color Image Processing and Applications, Berlin, Springer Verlag. 2000.
3. *Lukac, R., Smolka, B., Martin, K., Plataniotis, K.N, and Venetsanopoulos, A.N.* Vector filtering for color imaging. IEEE Signal Processing Magazine, Special Issue on Color Image Processing 2005. vol. 22, no. 1, pp. 74–86.
4. *Astola, J., Haavisto, P., and Neuvo, Y.* Vector median filters. Proc. of the IEEE. 1990; vol. 78, no. 4, pp. 678–689.
5. *Lukac, R., Smolka, B., Plataniotis, K.N., and Venetsanopoulos, A.N.* Selection weighted vector directional filters. Computer Vision and Image Understanding. 2004. vol. 94, pp. 140–167.
6. *Szczepanski, M., Smolka, B., Plataniotis, K.N., and Venetsanopoulos, A.N.* On the distance function approach to color image enhancement. Discrete Applied Mathematics. 2004, vol. 139, pp. 283–305.
7. *Lukac, R., Plataniotis, K.N., Smolka, B., and Venetsanopoulos, A.N.* Generalized selection weighted vector filters. EURASIP Journal on Applied Signal Processing: Special Issue on Nonlinear Signal and Image Processing. 2004, vol. 1, pp. 1870–1885.
8. *Lukac, R.* Adaptive color image filtering based on center weighted vector directional filters. Multidimensional Systems and Signal Processing. 2004, vol. 15, pp. 169–196.
9. *Ponomaryov, V., Gallegos-Funes, F., and Rosales-Silva, A.* Adaptive multichannel non parametric median M-type K-nearest neighbour (AMN-MMKNN) filter to remove impulsive noise from color images. Electronic Letters. 2004, vol. 40, no. 1, pp. 796–798.
10. *Ponomaryov, V.I., Gallegos-Funes, F.J., and Rosales-Silva, A.* Real-time color imaging based on RM-filters for impulsive noise reduction. Journal of Imaging Science and Technology, 2005, vol. 49, no. 3, pp. 205–219.
11. *Ponomaryov, V.I., Gallegos-Funes, F.J., and Rosales-Silva A.* Real-Time Color Image Processing Using Order Statistics Filters. Journal of Mathematical Imaging and Vision, Springer, 2005, vol. 23, no 3, pp. 315–319.
12. *Barner, K.E. and Arce G.R.* Eds., Nonlinear Signal and Image Processing: Theory, Methods, and Applications. Boca Raton, FL: CRC, 2004.

13. *Kerre, E.E.* Fuzzy Sets and Approximate Reasoning. Xian, China: Xian Jiaotong Univ.Press, 1998.
14. *Barner, K.E, Flaig, A., and Arce, G.R.* Fuzzy ranking: theory and applications. Signal Process.—Special Issue Fuzzy Process., 2000, vol. 80, pp. 1017–1036.
15. *Nie, Y. and. Barner, K.E.* Fuzzy transformation and its application in image processing. IEEE Trans. Image Process. 2006, vol. 15, no. 4, pp. 910–927.
16. *Russo, F. and Ramponi, G.* A fuzzy filter for image corrupted by impulse noise. IEEE Signal Processing Letters. 1996, vol. 3, no. 6, pp. 168–171.
17. *Schulte, S., De Witte, V., Nachtegael, M., Van der Weken, D., and Kerre, E.E.* Fuzzy random impulse noise reduction method. Fuzzy Sets Systems. 2006, vol. 158, no. 3, pp. 270–283.
18. *Morillas, S, Gregori, V., and Sapena, A.* Fuzzy bilateral filtering for color images. Lecture Notes Computer Science. 2006, vol. 4141, pp. 138–145.
19. *Nie, Y. and Barner, K.E.*, Fuzzy Rank LUM Filters. IEEE Trans. Image Process., 2006, vol. 15, no. 12, pp. 3636–3654.
20. *Gallegos-Funes, F.J. and Ponomaryov, V.I.* Real-time image filtering scheme based on robust estimators in presence of impulsive noise. Real Time Imaging, Elsevier. 2004, vol. 8, no. 2, pp. 69–80.
21. *Astola, J. and Kuosmanen, P.* Fundamentals of Nonlinear Digital Filtering, CRC Press, Boca Raton-New York, 1997.
22. *Ponomaryov, V.I.* Real-time 2D-3D filtering using order statistics based algorithms. Journal of Real-Time Image Processing. Springer Edit. 2007, vol. 1, no. 3, pp. 173–194. Survey paper.
23. *Lukac R.* Adaptive vector median filtering. Pattern Recognition Letters 2003, vol. 24, no. 12, pp. 1889–1899.
24. *Smolka, B., Chydzinski, A., Plataniotis, K.N., and Venetsanopoulos, A.N.* New filtering technique for the impulsive noise removal in color images. Mathematical Problems in Engineering. 2004, vol. 1, pp. 79–91.
25. *Ponomaryov, V., Gallegos-Funes, F., Sansores-Pech, R., and Sadovnychiy, S.* Real-time noise suppression in 3d ultrasound imaging based on order statistics. Electronics Letters, 2006, vol. 42, is. 2, pp. 80–81.
26. *Kravchenko, V.F., Ponomaryov, V.I., Pustovoit, V.I., and Sansores-Pech, R.* Nelineinaia filtratsia v realnom vremeni dlia trehmernyh ultrazvukovyh izobrazenii iskazennyh shumami. Dokladi Akademii Nauk RF, 2006, T. 408, No. 4, pp. 469–475.
27. *Kravchenko, V.F., Ponomaryov, V.I., Pustovoit, V.I., and Sansores-Pech, R.* Real-Time Nonlinear Filtration for Three-Dimensional Ultrasound Images Distorted by Noise. Doklady Physics. Distributed by Springer Inc. 2006, vol. 51, no. 6, pp. 304–310.
28. *Lee, Y.H. and Kassan, S.A.* Generalized median filtering and related nonlinear filtering techniques. IEEE Trans. Acoustic Speech and Signal Processing. 1985, vol. ASSP-35, pp. 672–683.
29. *Hardie R.C. and Barner, K.E.* Rank conditioned rank selection filter for signal restoration. IEEE Trans. On Image Processing. 1994, vol. 3, pp. 192–206.
30. *Arce, G.R.* Multistage order statistic filters for image sequence processing. IEEE Trans. Signal Processing. 1991, vol. 39, no. 5, pp. 1146–1163.
31. *Nieminen A. and Neuvo Y.* Comments of theoretical analysis of the max/median filter. IEEE Trans. Acoust., Speech, and Signal Process. 1988, vol. ASSP-36, pp. 826–827.
32. *Ko, S.J., Lee, Y.H., and Fam, A.T.* Selective median filters. Proc. IEEE Symp. On Circuits and Systems, Helsinki, 1988. pp. 1495–1498.
33. *Hardie, R.C. and Boncelet, C.G.* LUM filters: a class of rank order based filters for smoothing and sharpening. IEEE Trans. Signal Processing. 1993, vol. 41, pp. 1061–1076.
34. *Ponomaryov, V., Rosales-Silva, A., and Golikov, V.* Adaptive and vector directional processing applied to video colour images. Electronics Letters. 2006, vol. 42, is. 11, pp. 623–624.
35. *Ponomaryov, V., Gallegos-Funes, F., Rosales-Silva, A., and Loboda, I.* 3D vector directional filters to process video sequences. Journal «Lecture Notes in Computer Science». Edit. Springer. 2006, vol. 4225, pp. 474–480.

36. *Ponomaryov, V., Rosales, A., Gallegos, F., and Loboda, I.* Adaptive Vector Directional Filters to process Multichannel Images. IEICE Trans. On Fundamentals of Electronics, Communications and Computer Sciences. 2007, vol. E90-B, no. 2, pp. 429–430.
37. *Zlokolica, V., Philips, D., and Van de Ville, D.* A New Non-Linear Filter for Video Processing. Proc. of IEEE Benelux Signal Processing Symposium (SPS-2002). Belgium. 2002.
38. *Amer and Schroder, H.* A new video noise reduction algorithm using spatial subbands. ICECS'96. 1996, vol. 1, pp. 45–48.
39. *Rajagopalan, R.* Synthesizing processed video by filtering temporal relationships. IEEE Trans. on Image Processing. 2002, vol. 11, pp. 26–36.
40. *Zlokolica, V., Schulte, S., Pizurica, A., Philips, W., and Kerre, E.* Fuzzy Logic Recursive Motion Detection and denoising of Video Sequences. Journal of Electronic Imaging, 2006, vol. 15, is. 2. pp.
41. *Mehmet, K., Ibrahim, S., and Murat, T.* Adaptive motion-compensated filtering of noisy image sequences. IEEE Trans. On Circuits and Systems for Video Technology. 1993, vol. 3, pp. 277–290.
42. *Dugad, R. and Ahuja, N.* Video denoising by combining Kalman and Wiener estimates. IEEE International Conf. on Image Processing. 1999, pp. 152–156.
43. *Cocchia F., Carrato, S., and Ramponi, G.* Design and real-time implementation of a 3-D rational filter for edge preserving smoothing. Trans. on Consumer Electronics. 1997, vol. 43, pp. 1291–1300.
44. *Van De Ville, D., Nachtegael M., Van der Weken, D., Philips Wilfried, R., Lemahieu, I.L., and Kerre E.* New fuzzy filter for Gaussian noise reduction. Proc. SPIE. Visual Communications and Image Processing conference. 2001, vol. 4310, pp. 1–9, 2001.
45. *Kulkarni, A.D.* Computer Vision and Fuzzy-Neural Systems. Prentice-Hall, New York, Fuzzy Logic Fundamentals, Chapter 3, pp. 61–103, 2001, www.informit.com/content/images /0135705991/samplechapter/0135705991.pdf. 504p.
46. *Rosales-Silva, A., Ponomaryov, V., and Gallegos-Funes, F.* Fuzzy Directional Adaptive Recursive Temporal Filter for Denoising of Video Sequences. Journal «Lecture Notes in Computer Science», Edit. Springer. 2007, vol. LNCS4827, pp. 660–670.
47. *Rosales-Silva, A., Ponomaryov, V., and Gallegos-Funes, F.* Fuzzy Vector Directional Filters for Multichannel Image Denoising. Journal «Lecture Notes in Computer Science». Springer Edit. 2007, vol. LNCS4756, pp. 124–133.
48. *Kravchenko V.F., Ponomaryov V.I., and Pustovoit, V.I.* Algorithms of the 3D filtration with using fuzzy logics theory for color image sequences degraded by noise. Doklady Physics. Distributed by Springer Inc., 2008, vol. 53, no. 7, pp. 363–367.

Chapter 9

PROCESSING OF MULTIDIMENSIONAL SIGNALS USING DSP AND FPGA PLATFORMS

9.1. Platforms for Real Time Processing: DSP vs FPGA

Real-time imaging systems form a special class that has important applications in robotics, industrial inspection, high-definition television, advanced simulators, computer-aided systems, etc. The main characteristic of a real-time imaging system is its capability to realize processing in a finite time, but this definition is very broad [1]. Depending on the problem, this time can be very short, say, 30 ms or, as in medical diagnostic applications, longer, up to several seconds.

Usually, to satisfy conditions of real time imaging, different hardware platforms are used, especially the FPGA and DSP devices.

The hardware *FPLDs (Field Programmable Logic Devices)* that are fabricated by Xilink [2] and Altera [3] consists of arrays or matrixes of basic logic elements with a possibility to interconnect them via programming for implementation. Common for this type of FPLD is the *FPGA (Field Programmable Gate Arrays)*. It has the following principal characteristics.

Operation velocity attains to 200 MHz.

Density. These devices have programming circuits useful for programming ports. They can integrate up to 3 millions elements.

Development tools. The programming language is VHDL (Very High Speed Integrated Circuit Hardware Description Language) according to the IEEE standard. The compiler used by Altera is Max+PLUS II or codes in VHDL. There is a compiler for the C language named Handle C.

Programmable digital processors (DSPs), usually fabricated by Texas Instruments, Motorola, and Analog Devices, are realized as semiconductor devices capable of analog-to-digital convertion.

The main characteristics of this hardware are following.

Clock Velocity attains 1GHz in some types of devices produced by Texas Instruments, which permits more than 5000 million instructions per second (MIPS) and dozen of billion floating-point operations per second (GFLOPS).

Parallel architecture permits one to realize different clock instructions.

Development tools. There are compilers of C language for Windows and interface from Matlab to different DSP and also LabVIEW hardware and software;

Platform. The Started Kit is connected via parallel port to PC and EVM (Evaluation Module).

FPGA vs DSP.

Different aspects should be taken into account when a method of image/video processing is to be chosen. This depends on advantages and drawbacks of each platform. One should also take into account what platform is used for the given real-time application and what kind of platform is best suited for it.

Implementation that is based on FPGA should be careful when such operations as division are to be realized [4, 5]. The principal problem in DSPs is memory management

due to possible difficulties of the DMA (Direct Memory Access) and/or internal memory cache.

Below, we apply the DSP and FPGA to different problems, explaining every time the selection of the hardware.

9.2. Compression-Windowing Procedures Applicable in Radar Systems

Remote sensing systems, such as SAR usually apply FM linear signals to resolve closely placed targets and improve the signal-to-noise ratio (SNR) [6–8]. The principal parameter in radar applications is the range resolution, which is determined by the output signal bandwidth. In the case of FM signals, the output of a matched filter presents a compressed pulse accompanied by responses in other ranges, called lateral lobes in time or range. Weighting functions (windows) are usually applied in processing of output signals to reduce these lateral lobes. This can lead to a degradation of SNR, but, at the same time, the use of windows allows one to reduce side-lobes. Another drawback is that the width of the main lobe can increase, and, therefore the precise resolution is degraded. The selection of a window is determined by a compromise between the changes in the level of noise and in the attenuation of the side lobes. Several criteria should be taken into account when these window functions are applied, such as width of the main lobe, reduction of the side lobes amplitudes, rate of decrease of side lobes, etc. [9–11].

For implementation of signal processing procedures, such as pulse compression and windowing in the radar, one can choose different hardware: DSP and FPGA (Field Programmable Gate Array). Here, we use FPGA architecture, which has advantages in performance but also offers much of the flexibility of programmable DSP processors. The higher performance of the FPGA makes the applications developed with them closer to specific solutions and makes them appropriate in special purpose integrated circuits (ASICs) or commercial off-the-shelf (COTS) platforms [12].

9.2.1. Linear FM Signal. The linear FM signal usually used in SAR can be written as [6, 13, 14]

$$S(t) = \begin{cases} S_0(t)\cos(\omega_0 t + \mu t^2/2), & |t| \leqslant T/2, \\ 0, & \text{other } t, \end{cases} \tag{9.1}$$

where ω_0 is the central frequency, μ is the compression coefficient, and $S_0(t)$ is the signal amplitude. The use of such signal can significantly improve the resolution quality and detection capability of the radar. For the time $|t| \leqslant \tau/2$, we can write the following equation for a linear FM signal:

$$S(t) = S_0(t)\cos(2\pi f_0 t + \frac{\pi \Delta f}{\tau} t^2) = S_0(t)\Re\left(e^{i\pi(2f_0 t + \frac{\Delta f}{\tau} t^2)}\right)$$

with the complex amplitude $\overline{S}_0(t) = S_0(t)\exp(i\pi\frac{\Delta f}{\tau_u}t^2)$. It is known that resolution properties of a radar are defined by the ambiguity function which has the form [6, 13]

$$|\chi(\tau,\Omega)| = \frac{1}{2E}\left|\int_{-\infty}^{\infty} S_0(t)\exp(i\pi\frac{\Delta f}{\tau}t^2)\times \right.$$

$$\left. \times S_0(t-\tau)\exp(-i\pi\frac{\Delta f}{\tau}(t-\tau)^2)\exp(i2\pi\Omega t)dt\right|. \tag{9.2}$$

The resolution analysis can be realized by using the areas for each of the axes τ and ν as

$$\Lambda_\tau(\nu) = \int_{-\infty}^{\infty} |\chi(\tau_i, \nu)|^2 \, d\tau_i \quad \text{and} \quad \Lambda_\nu(\tau) = \int_{-\infty}^{\infty} |\chi(\tau, \nu_i)|^2 \, d\nu_i$$

which, in the case of matched filter, can be written as

$$\Lambda_\tau(\nu) = \frac{1}{T_{ef}^2} \int_{-\infty}^{\infty} |S(f)|^2 |S(f-\nu)|^2 \, df,$$

$$\Lambda_\nu(\tau) = \frac{1}{T_{\text{эф}}^2} \int_{-\infty}^{\infty} |S(t)|^2 |S(t-\tau)|^2 \, dt,$$

where T_{ef} is the effective signal duration. The areas of resolution in time and frequency are defined as it follows:

$$\Lambda_\tau(0) = \frac{1}{T_{ef}^2} \int_{-\infty}^{\infty} |S(f)|^4 \, df = \frac{1}{T_{ef}^2} \int_{-\infty}^{\infty} G^2(f) df,$$

$$\Lambda_\nu(0) = \frac{1}{T_{ef}^2} \int_{-\infty}^{\infty} |S(t)|^4 \, dt = \frac{1}{T_{ef}^2} \int_{-\infty}^{\infty} A_0^4(t) dt,$$

where $G(f) = \int \chi(\tau, 0) \exp(-i2\pi f\tau) \, d\tau$ is the power spectrum of the modulation and $A_0(t)$ is the signal amplitude. The precision of the estimations of the range R and radial velocity V_r as the maximum point for the ambiguity function in the axes τ and Ω can be characterized by the dispersion of the parameters τ and f [13]:

$$\sigma_\tau^2 = \frac{1}{(E/N_0)(2\pi f_{\text{quad}})^2} \quad \text{and} \quad \sigma_f^2 = \frac{1}{(E/N_0)(2\pi t_{\text{quad}})^2},$$

where

$$f_{\text{quad}} = \left(\frac{\int f^2 |S(f)|^2 \, df}{\int |S(f)|^2 \, df} \right)^{1/2}$$

is the quadratic mean band of the signal,

$$t_{\text{quad}} = \left(\frac{\int t^2 |S(t)|^2 \, df}{\int |S(t)|^2 \, df} \right)^{1/2}$$

is quadratic mean square signal duration, and E/N_0 is the SNR in the system input.

Thus, the use of linear FM permits one to attain high resolution both in range and velocity, since the frequency deviation and signal duration are independent parameters, and their product, which characterizes the estimation precision, also can be a large value.

Pulse compression is a standard signal processing technique used to minimize the peak transmission power, to maximize SNR, and to get better resolution. It is known

that the impulse response $h(t)$ of such filter is the complex conjugate of the time-reversed chirp:

$$h(t) = ks^*(-t + \tau_p). \tag{9.3}$$

Here, $s(t)$ is the transmitted signal and $r(t)$ is the received signal. So the output of the matched filter can be written as

$$g(t) = \frac{1}{T} \int_{-T/2}^{T/2} r(\tau)\, s^*(\tau - t)\, d\tau, \tag{9.4}$$

or, in the case of discrete time, as:

$$g(n) = \frac{1}{N} \sum_{k=0}^{N-1} r(n) S^*(n-k).$$

Figure 9.1 presents some explanation of the radar compression procedure in the case of linear frequency modulation.

Figure 9.2 presents the model used for the pulse compression. We used four FIR filters with real and imaginary parts at the input. The signal reference is also presented by the real and imaginary part. The real part of pulse compression is calculated by the summation of FIR 1 and FIR 2, and the imaginary part is calculated using FIR 3 and FIR 4 outputs. Finally, the absolute value (ABS) is formed using complex signal to realize the pulse compression procedure.

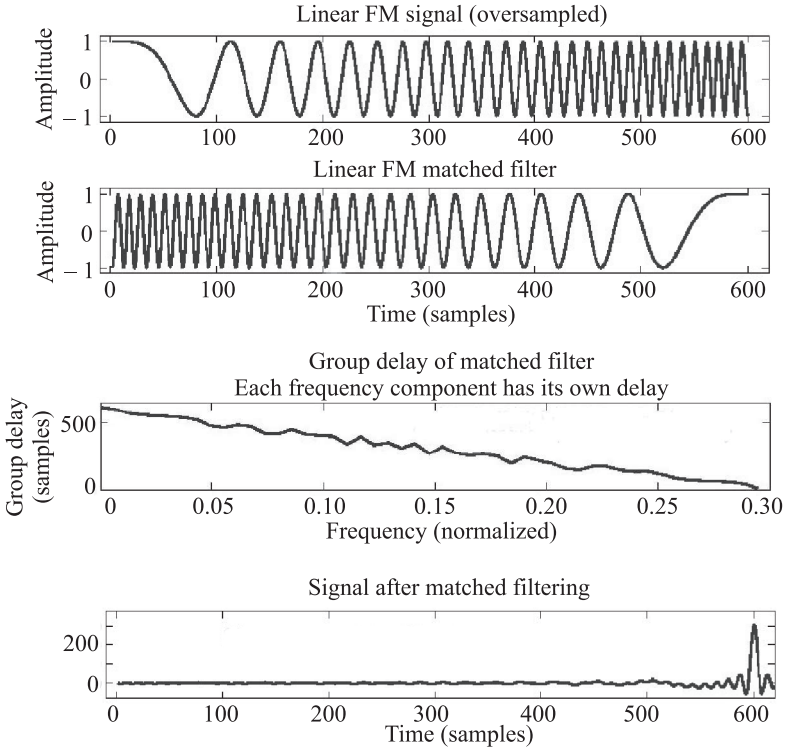


Fig. 9.1. Chirp signal and matched filter.

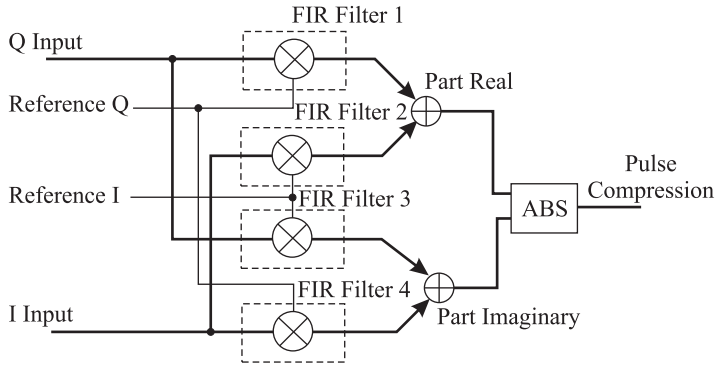


Fig. 9.2. Model of pulse compression.

9.2.2. Windowing. In applications of remote sensing, it is required to reconstruct the target parameters, which can be darkened by side lobes of a very large (or powerful) target [6, 7]. So, the principal difficulty is to distinguish between a small target and side lobes of a large or powerful target. Note that the first side lobe of the uniform phantom has an attenuation of 13 dB below the main lobe.

The usage of the weighting functions (windows) in the time domain essentially influences the effect of spectral loss. Weighting function in time domain can be implemented by multiplying the FIR filter coefficients and radar signal correcting it. The absence of this function for a finite analyzed signal part is equivalent to the use of a rectangular window. The use of the windows different from rectangular and smoothing discontinuities of the signal at the ends of the segment allows one to reduce the side-lobe level, but, at the same time, the main lobe can extend and resolution can be degraded. The choice of the window is determined by a compromise between the noise and side-lobe levels. If we want to obtain a high resolution between near signal components, and distant components are absent, windows with a very narrow main lobe and minimal amplitude of neighboring side-lobes are required [9, 11].

Below, we present a novel method for design of windows based on combination of atomic functions (AF) with classical windows, such as Gauss, Bernstein, and Dolph–Chebyshev functions. Characteristics of the new weighting functions, as well as those of classical Hamming, Blackman–Harris, and Kaiser–Bessel windows are presented too. Some of them possess extraordinary properties making these functions useful in digital signal processing, including that connected with SAR.

9.2.3. Simulation Experiments. In our experiments we used a radar with the following parameters: the signal in the form of a linear FM (Chirp), the frequency deviation (Δf) of 9.375 MHz, the pulse width (T) of 3.2 μs, and number of the taps in the matched filter equal to 800, the sampling frequency of 249.68 MHz, and the antenna gain of 40 dB [15, 16]. We also used weighting function in time domain, and simply multiplied the coefficients the FIR filter window and the signal.

The performance of different windows has been widely studied. Special attention was given to improvement their side-lobe behavior.

We are going to evaluate all functions presented in Table 9.1 and compare each of these windows with the best classical one, the Hamming window. The parameters used for this evaluation are the *window gain, side lobe level, main lobe width*, and *the coefficient of noise performance*. All these parameters have been defined above in Chapters 2 and 3 of this book [11, 17, 18] as follows:

window gain, $\mathrm{K}_{\text{gain}} = 0.5 \int\limits_{-0.5}^{0.5} W(t)dt$, where $W(t)$ represents the window;

side lobe level, $10 \log \left(\max\limits_{k} |S_{com}(t=0)/S_{com}(t)|^2 \right)$;

main lobe width at 6dB level is defined from the following equation with respect for t:

$$10 \log \left(|S_{com}(t=0)/S_{com}(t)|^2 \right) = 6 \; dB,$$

where $S_{com}(t)$ is the compressed signal after window processing;

and *coefficient of noise performance*, $SNR_{window}/SNR_{rectangular}$.

The analysis of this table reveals that, of all the windows, the best window performance for radar pulse compression is demonstrated by the Hamming window, which has the gain of 0.54, the side-lobe level of -32 dB, and the main lobe width of 239.2 ηsec. One can see that the main lobe width is nearly double the 129 ηsec main lobe width of the rectangular window. However, comparing all the windows with the Hamming one, we come to the following conclusion. The function fup_4 offers smaller attenuation in the main lobe amplitude, as well as a lower main lobe width in comparison with the Hamming window. Since the attenuation of the side lobes is one of the most important parameters, the use function $\mathrm{up}(x)$ is preferable due to its better performance. From analysis of Table 9.1, one may conclude that the best results can be obtained with the novel windows such as the $\mathrm{fup}_4(x) \cdot D_3(x)$.

Table 9.1. Values of parameters for different windows in radar pulse compression.

Windows	Gain	Side lobe level (dB)	Main lobe width at -6 dB (ηsec)	Coefficient of performance in noise presence
Rectangular	1	-13.7	129.5	1
Hamming	0.5398	-32	239.2	0.7527
Blackman	0.4199	-31.3	350	0.6096
Blackman–Harris	0.3588	-30.1	477.1	0.5395
Kaiser–Bessel $\beta = 3.5$	0.6403	-26.7	220.2	0.7534
$\mathrm{fup}_3(x)$	0.3871	-30.7	368.7	0.4823
$\mathrm{fup}_4(x)$	0.5797	-25.7	208	0.4542
$\mathrm{up}(x)$	0.5	-31.4	249.1	0.6359
$\mathrm{fup}_4^2(x) \cdot B^2(x)$	0.4002	-30.9	256.8	0.5858
$\mathrm{fup}_4(x) \cdot D_3(x)$	0.4246	-31.3	240.3	0.6230
$\mathrm{fup}_4(x) \cdot D_{3.5}(x)$	0.3988	-30.9	255.5	0.5907
$\mathrm{fup}_6^2(x) \cdot G_2^2(x)$	0.3769	-30.5	266.3	0.5720
$\mathrm{fup}_6(x) \cdot G_3(x)$	0.386	-30.7	259.9	0.5833
$\mathrm{fup}_6^2(x) \cdot G_3(x)$	0.3611	-30.2	255.9	0.5522

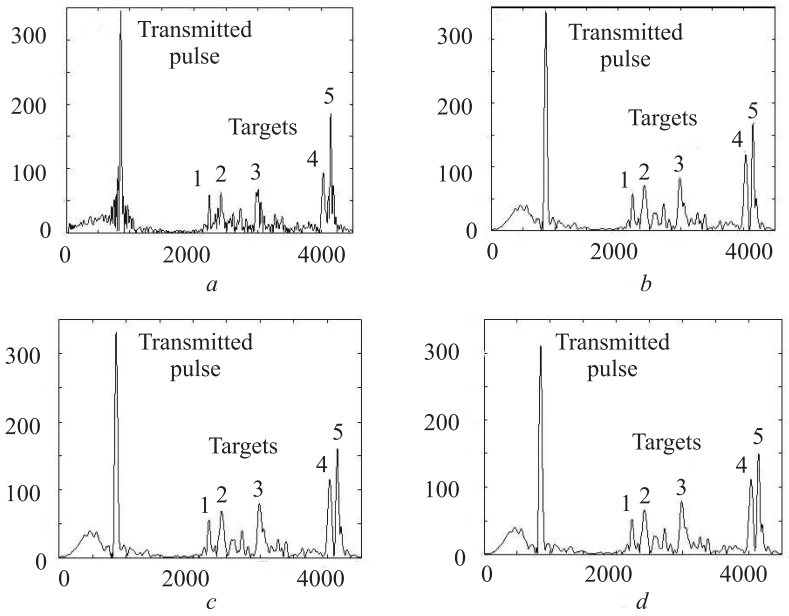


Fig. 9.3. a) Recangular window; b) Hamming window; c) fup_6 window; d) fup_9 window.

Figure 9.3 a, b, c, d show the results of processing for several windows, demonstrating their visual performance. Here, the radar data volume was reduced to 4500 samples for representing the radar scattering from the targets.

9.2.4. Hardware Implementation. In addition to consideration of the model, we proposed its implementation on a hardware level, realizing the window processing in real time [19].

We employed a Field Programmable Gate Array (FPGA), which has a low price and is good for fabricating different systems. In order to test the windowing, the Kit FPGA Xilinx model VIRTEX II XC2V3000 was used to realize the radar pulse operation. This Kit has 2 DAC's and 4 ADC's, each one of 10 bits. Figures 9.4 and 9.5 present the model and result of pulse creation. We used in this case the frequency sampling of 40 MHz.

Implementation of the matched filter in the FPGA makes it possible to eliminate special chips previously needed. We have tested the performance of such a model in the FPGA Xilinx model VIRTEX II XC2V3000. The use of the hardware Xtreme DSP along with the software, namely, Matlab 6.5, Simulink, System Generator, and FUSE made it possible to implement the pulse compression procedure in the FPGA in real time. From analysis of different models, we established the final structure of FPGA shown in Figure 9.5. One of the advantages of the abovementioned software and hardware is the ease of changing parameters of each a block in the system. If a better precision is required, the number of bits can be increased in each block.

The maximum number of taps was 65, and the same FPGA model shown in Figure 9.5 was used for implementation of classical and novel windows based on atomic functions. The delay of the system was 2.7 μseg, and, if it is necessary to increase the number of taps, the delay will increase too. The system can run at 162 MHz.

Figure 9.6 illustrates the pulse compression in the radar on the abovementioned hardware for some classical and novel windows.

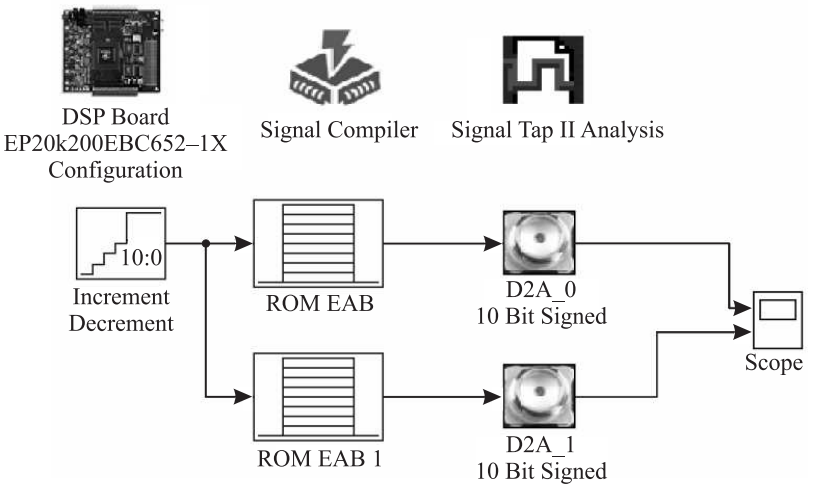


Fig. 9.4. Model for radar transmitting pulse.

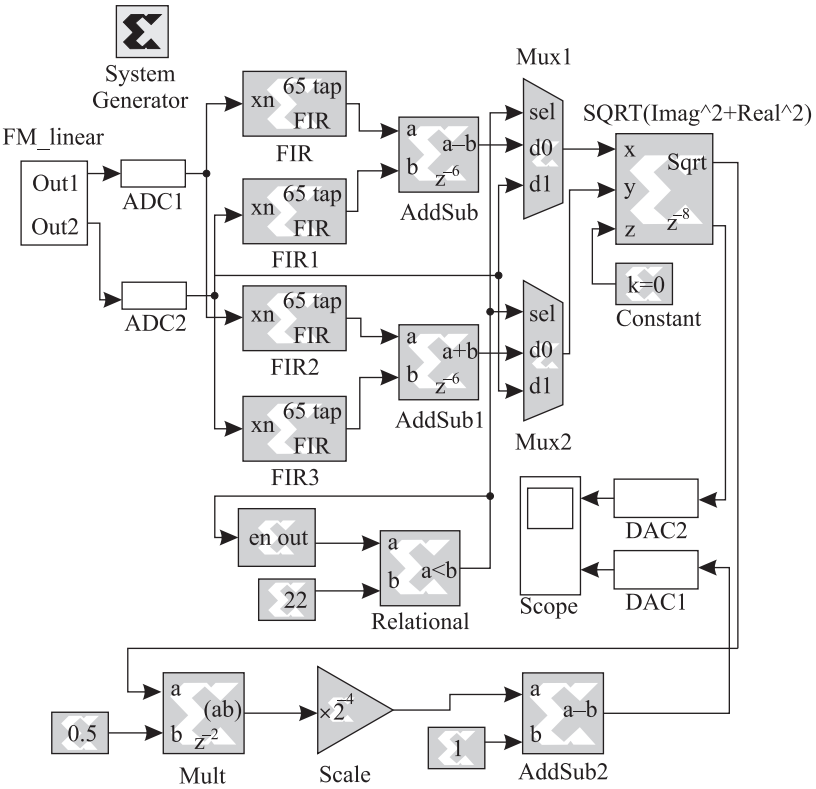


Fig. 9.5. Hardware model.

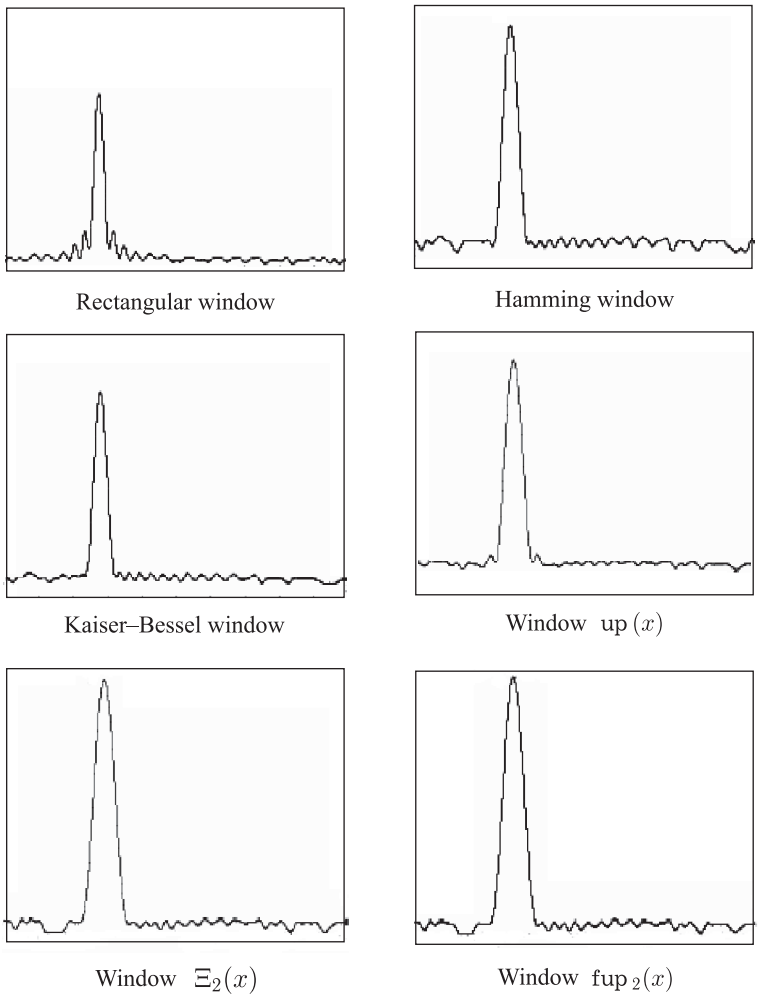


Fig. 9.6. Hardware results of windowing.

Experimental results presented in Fig.9.6 demonstrate the pulse compression for the rectangular, Hamming, Kaiser–Bessel, and the novel windows based on the atomic functions. One can see that the novel windows offer a better decrease of the sidelobes with a distance from the main lobe. Figure 9.7 demonstrates the experimental hardware results of windowing in pulse compression radar using real data to resolve several targets for the rectangular, Hamming, Kaiser–Bessel, and some novel windows: up (x), fup $_4(x) \cdot D_3(x)$, and fup $_6(x) \cdot G_3(x)$.

We can see that windows based on the AF exhibit an essentially more rapid decrease of the side lobes as compared to the classical ones. Thus, obviously, the resolution of nearly placed targets 2 and 3 is better for windows generated by functions up (x), fup $_4(x) \cdot D_3(x)$, and fup $_6(x) \cdot G_3(x)$ than for the rectangular, Hamming, and Kaiser-Bessel windows.

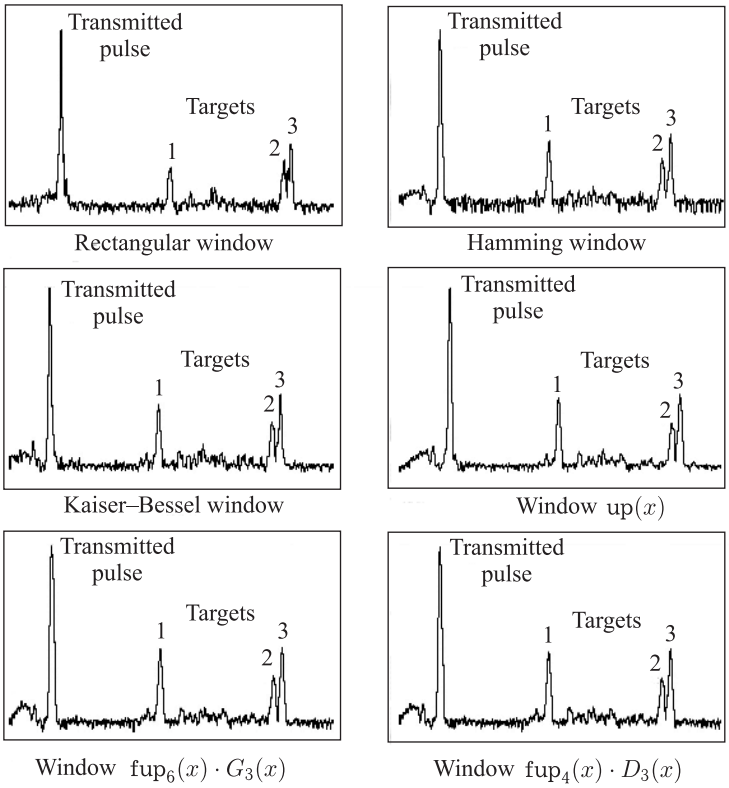


Fig. 9.7. Hardware results of windowing in pulse compression radar resolving several targets.

Table 9.2. Numerical hardware results for window processing.

Window	Sidelobe level, (dB)	Main-lobe width at -6 dB (ηs)
Rectangular	-14.1	124
Blackman, 4 term	-31.8	274
Chebyshev	-31.7	272
Hamming	-33.3	196
Hanning	-31.8	214
Kaiser–Bessel	-31.1	172
$\text{fup}_2(x)$	-29.2	259
up (x)	-29.8	258
$\text{fup}_4^2(x) \cdot B^2(x)$	-31.8	256
$\text{fup}_4(x) \cdot D_3(x)$	-32.2	237
$\text{fup}_4(x) \cdot D_{3.5}(x)$	-31.6	248
$\text{fup}_6^2(x) \cdot G_2^2(x)$	-31.9	263
$\text{fup}_6(x) \cdot G_3(x)$	-31.3	297
$\text{fup}_6^2(x) \cdot G_3(x)$	-32.7	302

The maximum side-lobe amplitudes and the main-lobe width of -6 dB was obtained in the experiments presented in Table 9.2. As can be seen from Tables 9.1 and 9.2, most of values of the side-lobe amplitudes and the main-lobe width coincide.

The pulse compression results in the presence of noise (input SNR= 20 dB) are shown in Figure 9.8. As can be seen there, the best results are attained by the AF $\mathrm{up}_{(}(x)$ since this function provides small sidelobes and a good resolution. Another window demonstrating good characteristics in the presence of noise is $\mathrm{fup}_4(x) \cdot D_3(x)$, but the gain in this case equals 0.36, which is smaller than for the $\mathrm{up}_{(}(x)$ function.

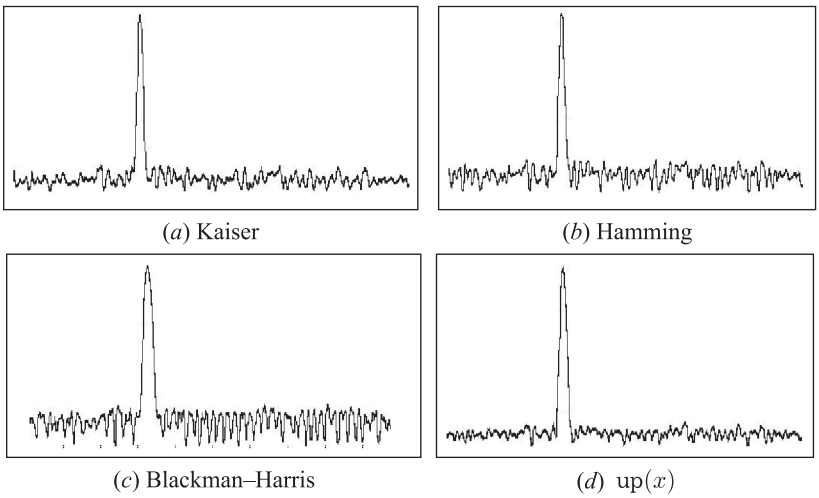


Fig. 9.8. Main lobe in real-time pulse compression for different windows (SNR= 20 dB).

So, novel windowing functions have demonstrated a better quality of pulse compression in the FM radar applications. The windows $\mathrm{fup}_4(x) \cdot D_3(x)$, $\mathrm{fup}_4(x) \cdot D_{3.5}(x)$, and $\mathrm{fup}_6(x) \cdot G_3(x)$ gave better results in comparison with the classical ones.

The good performances of these functions makes possible the application of the novel windows in processing of FM radar data. Also, the FPGA implementation of the model using the novel windows has demonstrated an effective elimination of the sidelobes.

9.3. Runtime Analysis of 2D-3D Filtering Algorithms

9.3.1. 3D Ultrasound Filtering. The runtime analysis of the 3-D RM-KNN and other filters were implemented on the basis the Texas Instruments DSP TMS320C6711 [4, 5]. This DSP has a performance as high as 1 GFLOPS at a clock rate of 176 MHz [5]. The filtering algorithms were implemented in C language using the BORLAND C 3.1 for all routines, data structure processing, and low-level I/O operations. Then, the programs were compiled and executed by the DSP TMS320C6711 applying the Code Composer Studio 2.0. Then, the programs were compiled in the C6711 C compiler in order to create the assembler file (.asm file), the object file (.obj file), and the executable COFF file (.out file). The.out file was simulated in the DSP using a stand-alone simulator. Finally, the.out file was loaded and executed in the C6711 target using COFF loader utility.

The experiment was realized using an ultrasound sequence of $525 \times 382 \times 12$ image voxels. The sequence was degraded with 5, 10, 15, 20, 25, and 30% impulsive noise.

Table 9.3 shows the processing time in seconds for the proposed filters and other filters used for comparison. The processing time includes the time for acquisition, processing, and storing data. One can see from this table that the processing time for the selection median and average filters have sufficiently small time values. These filters use the technique that permits dividing the cube into two groups for fast calculation of the mean and the median, but for the LUM Smooth, LUM Sharp, and LUM filters, the time is increased at the stage of ordering of 27 voxels. The processing time for the RM-KNN filters is large in comparison with the other filters. It is easy to see that processing time is greater but the PSNR and MAE performance criteria are sufficiently better for the RM-KNN filters in comparison with other known filters, as it has been demonstrated in Chapter 7.

Other experiments were also made, in which the ultrasound sequence was processed using different cube configurations. Table 9.4 presents the processing time in seconds for different cube configurations in the MM-KNN filter with the Hampel and Cut influence functions. From the analysis of this table, a conclusion can made that the use of the cube voxel configurations from a to f makes possible a significant reduction of the processing time without a significant loss in the quality of filtering.

Table 9.3. Processing time in seconds for 3-D filtering in the case of impulsive noise.

3-D Filters	Impulsive noise percentage					
	5%	10%	15%	20%	25%	30%
Modified Trimmed Mean	2.1716	2.1716	2.1716	2.1716	2.1716	2.1716
Ranked Order	1.6836	1.6836	1.6836	1.6836	1.6836	1.6836
MSM1	0.5846	0.5846	0.5846	0.5846	0.5846	0.5846
MSM2	0.5773	0.5773	0.5773	0.5773	0.5773	0.5773
MSM5	1.2198	1.2198	1.2198	1.2198	1.2198	1.2198
MSM6	1.1667	1.1667	1.1667	1.1667	1.1667	1.1667
MaxMed	1.1981	1.1981	1.1981	1.1981	1.1981	1.1981
SelMed	2.3240	2.3240	2.3240	2.3240	2.3240	2.3240
LUM Smooth	4.122	4.705	5.355	5.754	6.047	7.123
LUM	4.317	4.915	5.582	5.984	6.285	7.402
MM-KNN CUT	20.49	20.59	20.61	20.63	20.66	20.87
MM-KNN HAMPEL	2051	20.53	21.02	21.26	21.26	21.75
MM-KNN BERNOULLI	21.94	22.01	22.25	22.49	22.73	22.98

9.3.2. Runtime Analysis in Color Imaging. The runtime analysis of various filters (see Sec. 8.5.1) in color image processing was realized on DSP TMS320C6711 Texas Instruments. The processing time in seconds of the different filters is presented in Table 9.5 and includes the time of acquisition, processing, and storing of data.

From the analysis of this table, it is easy to see that the processing times of the proposed VRMKNNF with different influence functions lies in the range from 0.2 to 0.5 s. The time for the proposed VMMMKNNNF and VWMKNNNF is less than for

Table 9.4. Processing time for the MM-KNN filters for different cube configurations.

Cube configurations	MMKNN Hampel			MMKNN Cut		
	Impulsive Noise Percent					
	10	20	30	10	20	30
a	1.763	1.787	1.812	1,594	1.643	1.659
b	2.053	2.077	2.101	1.866	1.908	1.916
c	4.618	4.635	4.692	4.807	4.823	4.872
d	4.696	5.201	5.264	4.754	5.199	5.252
e	4.66	4.628	4.690	4.804	4.816	4.869
f	4.616	4.642	4.693	4.800	4.830	4.871
g	9.939	10.04	10.06	10.06	10.06	10.06
h	9.969	10.02	10.04	10.05	10.07	10.08
i	21.02	21.26	21.75	20.61	20.66	20.87

Table 9.5. Processing time for different filters on the «Mandrill», «Lena», and «Peppers» color images degraded by 10, 20, and 30% impulsive noise.

Algorithm	Processing Time		
	Mandrill	Lena	Peppers
VMF	0.039	0.039	0.039
GVDF	0.533	0.564	0.565
GVDF_DW	0.720	0.721	0.723
AMN-VMF	0.648	0.648	0.648
AVMF	0.137	0.137	0.137
VMF_FAS	0.22	0.22	0.22
AMN-VMMKNNF Simple	3.666	3.687	3.726
VMMKNNF Simple	0.311	0.296	0.316
VMMKNNF Hampel	0.181	0.199	0.196
VWMKNNF Hampel	0.413	0.398	0.409
VABSTMKNNF Hampel	0.322	0.264	0.355

classical reference filters, except the VMF, α-TMF, and AMNF filters, and slightly more than for the AVMF and VMF_FAS filters. So, the proposed the VRMKNNF filter can process up to 5 images of 320×320 pixels per second, depending on the influence function used. The processing time for the AMN-VMMKNNF filter is greater than for any another filter, but, as shown above, this filter has a better performance for high noise corruption.

Table 9.6 presents the processing time required for processing of «Miss America», «Flowers», and «Foreman» video sequences of 150, 120, and 400 frames by different filters. The VMMKNNF (Hampel) filter is capable of processing any of these sequences with a speed from 10 to 14 frames per second.

It is obvious that, being applied to images with the number of pixels four or five times less than 320×320, the VMMKNNF and VWMKNNF filters can preserve edges

and small-size details and remove impulsive noise sufficiently well compared to other filters with standard film velocity for computer vision applications.

Table 9.6. Processing time for different filters applied to video color frame sequences.

Algorithm	Processing Time					
	Flowers		Foreman		Miss America	
	Min	Max	Min	Max	Min	Max
VMF	0.0153	0.0153	0.0153	0.0153	0.0153	0.0153
AGVDF	0.2263	0.2426	0.2143	0.2424	0.2106	0.2424
AMNF	0.0371	0.0371	0.0371	0.0371	0.0371	0.0371
AMN-VMF	0.2506	0.2506	0.2506	0.2506	0.2506	0.2506
AVMF	0.0533	0.0533	0.0533	0.0533	0.0533	0.0533
VMF_FAS	0.0944	0.0944	0.0944	0.0944	0.0944	0.0944
AMN-VMMKNNF Simple	1.4426	1.4503	1.3919	1.4510	1.3417	1.4454
VMMKNNF Hampel	0.0595	0.0702	0.0811	0.0865	0.0917	0.0983
VWMKNNF Hampel	0.2661	0.3110	0.2686	0.2896	0.2912	0.3040
VABSTMKNNF Simple	0.0924	0.1005	0.0908	0.0986	0.0929	0.0998
VABSTMKNNF Hampel	0.1200	0.1236	0.1182	0.1228	0.1194	0.1266

9.3.3. 3D Vector Filters in Multichannel Processing. Here, we discuss some results of runtime analysis for 3D color imaging.

The Imaging Developer's Kit (IDK) has been made as a platform for development and demonstration of image/video processing applications on TMS320C6000 platform. The IDK is based on the TMS320C6711 floating point DSK board, useful for development of algorithms in imaging and video processing. The use of Daughter-Card, which supports several bits by pixels, for video capture, display, and data conversion and drivers developed by Texas Instruments provides imaging of 16 bits/pixel in 565 RGB format.

The IDK hardware consists of a C6711 DSK with 16MB SDRAM and a daughter-card with the following capabilities: Video Capture of NTSC/PAL signals (composite video); Display of RGB signals, 640×480 or 800×600 resolution, 16-bits per pixel (565 format); Video data formatting by an on-board FPGA to convert captured interleaved 4:2:2 data to separate *Y, Cr, Cb* components that may be sent to the DSP for processing; Video capture and display drivers software written using DSP/BIOS and CSL. This board has capabilities required for being used in speed processors to work with 3D algorithms.

Only two algorithms are presented here in order to demonstrate the effectiveness of this kind of board. These algorithms are *«Median Filter»* and *«Vector Median Filter»* used for processing 1, 2 and 3 frames. Thus we prove the general idea that 3D algorithms are more powerful that 2D algorithms but require more memory and processor speed. One should distinguish two processing times: the first, taken only by the algorithm (*Time spent by the algorithm*) and the second, the *«complete time»* (to capture, store, process, and show) as the time spent in a complete processing system.

As we can notice, the processing time for one frame is satisfactory, so that the median filter is capable of processing 24 frames/s although the vector median filter can process only 7 frames/sec. But processing of «two frames» requires more than double processing time. In 3D algorithms, the processor is not sufficiently fast, and a more

Table 9.7. Processing time required by the Vectorial filters in the Imaging Developer's Kit.

ALGORITHM	One Frame		Two Frames		Three Frames	
	Time spent by the algo-rithm, ms	Complete Time, ms	Time spent by the algo-rithm, ms	Complete Time, ms	Time spent by the algo-rithm, ms	Complete Time, ms
MF	25.4	40.8	91.41	108.15	290.97	308.67
VMF	122.6	137.5	336.86	352.9	641.55	659.73
GVDF		$4.5\,10^3$				
AMNF		$3.2\,10^3$				
AMN-KNN		$7.5\,10^3$				
AVMF		78.0				
FDARTF_G (Sec. 8.5.4)		$6.0\,10^3$				

powerfulness processor is required. There is hardware capable of realizing the processing that will be used in future developments.

9.4. Compression-Recognition Techniques Using Wavelet Transform

9.4.1. Compression Algorithms Based on Wavelet Transform. Different fields, such as astronomy, medical imaging, and computer vision manage the data of large volume. So, this data should be compressed to optimize the storage devices. There is a lot of approaches to 2D–3D signal compression. Here, we present wavelet-based techniques for compression procedures focusing on different threshold rules. The basic idea behind these techniques is to use Wavelets to transform data set into a different basis, where the unimportant information can be eliminated. Also, we have tested both classical and novel wavelet algorithms based on the atomic functions, which demonstrate excellent compression results. Below, decimated wavelet transforms (WT) and the MAE fidelity criterion are used to evaluate the different compression methods for US and MG images [20–22].

9.4.1.1. Properties of Wavelet Transform.

The wavelet (WL) transform introduces an intriguing twist to the basic concept defined by the Fourier transform. In WL analysis, a variety of different probing functions may be used, but the family always consists of enlarged or compressed versions of the basic function, as well as translations (see also Chapter 4). This concept leads to the defining equation for the continuous wavelet transform (CWT) presented in equation (7.69).

The Discrete Wavelet Transform (DWT) achieves this parsimony by restricting the variation in translation and scale, usually to powers of 2. When the scale is changed in powers of 2, the discrete wavelet transform is sometimes termed as the Dyadic Wavelet Transform.

The Discrete Wavelet Transform (DWT) is easy to realize using filter banks. DWT can be implemented applying some equations, but it is usually made using filter bank

techniques. The most popular scheme of the DWT for 2-D signal uses only two filters for rows and columns, as in the symmetric filter bank (SFB).

In SFB demonstrated in Fig. 7.10, *Lx* and *Hx* denote the low-pass and high-pass filters to rows of an image of $M \times N$ pixels, *Ly* and *Hy* denote the low-pass and high-pass filters to columns of an image of $M/2 \times N$ pixels. This filter produces four subband images of $M/2 \times N$ pixels LL_1 (Low-Low), LH_1 (Low-High), HL_1 (High-Low) and HH_1 (High-High). Applying this procedure once again to sub-image LL_1, it is possible to obtain four sub-images of $M/4 \times N/4$ pixels LL_2, LH_2, HL_2 and HH_2.

The DWT is a bilateral transform; all of the information in the original waveform should be contained in the subband signals. These subband signals, or some aspect of the subband signals, such as their energy over a given time period, could provide a succinct description of some important aspect of the original signal.

However, other decomposition structures are valid, including the complete or balanced tree structure. In this decomposition scheme, both high-pass and low-pass subbands are further decomposed into high-pass and low-pass subbands up till the terminal signals. This structure is known as wavelet packets (WP).

9.4.1.2. Compression by Wavelet Threshold.

The three main steps of compression using the wavelet coefficient and threshold technique are as follows [23]:

1. *Calculate the wavelet coefficient matrix applying a WT to the original image.*
2. *Modify (threshold or shrink) the detail coefficients to obtain the reduced number of coefficients.*
3. *Encode the modified coefficients to obtain the compressed image.*

Thresholding Functions. These functions determine how the thresholds are applied to the data. The most popular are four thresholds, a single threshold ($\pm t$) is required for the hard $[\delta_t^H(w)]$, soft $[\delta_t^S(w)]$, and garrote $[\delta_t^G(w)]$ functions, but for semisoft function $[\delta_{t_1,t_2}^{SS}(w)]$ two thresholds ($\pm t_1$ and $\pm t_2$) are required (Fig. 9.9).

Hard function does not modify the original data of the wavelet coefficients, so, for this reason we use only hard function in the experiments. The hard function is given by the following equation:

$$S = \begin{cases} x & \text{if} \quad |x| > t \\ 0 & \text{if} \quad |x| \leqslant t \end{cases}, \tag{9.5}$$

where x is the original signal, S is the threshold signal and t is threshold.

Thresholding Rules. The thresholding rules determine how the thresholds should be calculated. Certain rules calculate the threshold independently of the image type, while others obtain different thresholds for different values of every image. We indicate the assumptions of each method as they are described here [20, 23, 24].

Average Threshold. This is the simplest method to calculate threshold. It consists of obtaining the average of the minimum and maximum data values, and it is given by the following equation:

$$t = \left(\frac{V_{\max} - V_{\min}}{2}\right). \tag{9.6}$$

Universal Threshold. The universal rule was proposed by Donoho [23]. It uses the statistics of the wavelet coefficients, and it is based on the standard deviation of the image. Universal threshold is given by

$$t = \sigma\sqrt{\frac{2\,\log_2(N)}{N}}, \tag{9.7}$$

where N is number of pixels of the image and σ is the standard deviation.

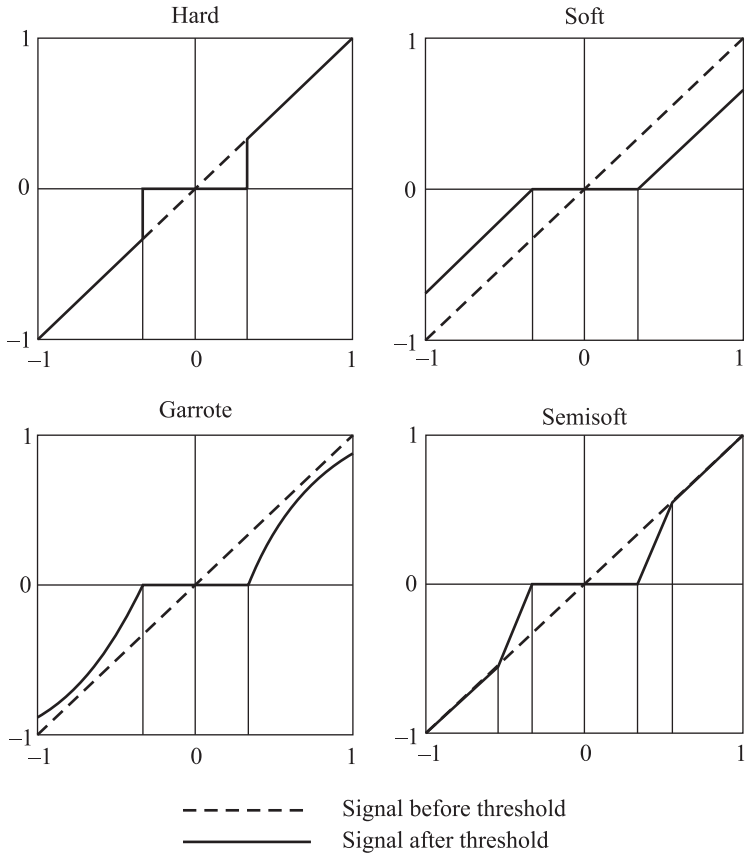


Fig. 9.9. Different thresholding functions.

Top Threshold. The top rule is a global method that is independent of the thresholding function. Given p as the fraction of the largest coefficients to keep, the threshold t is the (1-p)'th quantile of the empirical distribution of the absolute values of the wavelet coefficients. It is applied according to the hard function.

Bayes Shrink Threshold. The Bayes shrink rule uses a Bayesian mathematical framework for images to derive subband-dependent thresholds that are nearly optimal. The formula for the threshold for a given subband is

$$t_s = \frac{\widehat{\sigma}^2}{\widehat{\sigma}_X^2}, \tag{9.8}$$

where $\widehat{\sigma}^2$ is the estimated noise variance, and $\widehat{\sigma}_X^2$ is the estimated signal variance on the subband (X) considered. The noise variance is estimated as the scaled median absolute deviation of the diagonal detail coefficients on level 1 in the subband HH_1.

9.4.1.3. Proposed Compression Scheme.

The proposed compression scheme is shown in Fig. 9.10. The original image is decomposed using WL or WP, applying threshold to obtained decomposition coefficients, applying quantization and calculating the encoding reduced coefficients, finally, the compressed image should be formed.

For an objective evaluation of fidelity of the compressed images, we employ several criteria that usually are used to evaluate the difference between two images [25–28], the compression rate (CR) was also used to evaluate the compression level of the system.

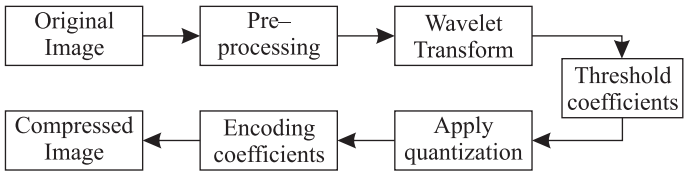


Fig. 9.10. Proposed compression scheme.

Mean Absolute Error (MAE). This criterion is often used as a numeric measure of distortion for fine details and contours of the image and characterizes the average difference between the original and compressed images:

$$MAE = \frac{1}{M \cdot N} \sum_{m=1}^{M} \sum_{n=1}^{N} |y[m,n] - \widehat{y}[m,n]|, \tag{9.9}$$

where M, N are sizes of the image, $y[m,n]$ is original image, and $\widehat{y}[m,n]$ is compressed image.

Compression Rate (CR). This criterion characterizes the compression quality an is given as follows:

$$CR = \frac{Original\ image\ size}{Compressed\ image\ size}. \tag{9.10}$$

We have carried out analysis of different wavelets used especially for Ultrasound (US) and Mammography (MG) images compression, also including new wavelets families based on atomic functions. This way we have been able to determine what type of wavelet filters works better for a specialized compression scheme for this type of images. Here, we determine and compare their key properties: Frequency response, approximation order, projection cosine, and Riesz bounds — these key properties were obtained for the classic wavelets W9/7, Daubechies8, and Symlet8, as well as for the complex Kravchenko–Rvachev wavelets $\psi(t)$ (see Chapter 4) based on the Atomic Functions $\mathrm{up}\,(t)$, $\mathrm{fup}_2(t)$, and *eup(t)*, introduced in Chapters 1 and 2 [29].

Let us investigate wavelet-based techniques for compression and focusing on different wavelets based in Atomic Functions. The basic idea behind these techniques is to use wavelets to transform data set into a different basis, where unimportant information can be eliminated.

Using the key properties of the different wavelets, we can justify the obtained experimental results for compression of US and MG images [20].

Wavelet Transform and Filter Banks: The Discrete Wavelet Transform (DWT) is easy to realize using filter banks as it has been mentioned before, DWT can be implemented applying some equations, but it is usually made using filter bank techniques.

The most popular scheme of the DWT for 2-D signal uses only two filters for rows and columns, as in the symmetric filter bank.

Wavelets Used in Compression: In tests carried out previously, it was found that better results are obtained when compressing the Ultrasound images with the Symlet wavelet and with the Daubechies wavelet for mammography images [22–24].

Therefore, we realize an evaluation to compare the acting of 3 wavelet families based on atomic functions with the classical wavelets that presented better acting.

We use the complex Kravchenko–Rvachev wavelets [29] ψ(t) based on the atomic functions $\mathrm{up}(t)$ $\mathrm{fup}_2(t)$ and $\mathrm{eup}(t)$.

Wavelet Key properties: The wavelet decomposition algorithm employs two analysis filters $\widetilde{H}(z)$ (low-pass) and $\widetilde{G}(z)$ (high-pass). The reconstruction algorithm applies the complementary synthesis filters $H(z)$ (low-pass) and $G(z)$ (high-pass). These four filters constitute a perfect reconstruction filter bank. In the present case, the system is entirely specified by the low-pass filters $H(z)$ and $\widetilde{H}(z)$, which form a biorthogonal pair. The high-pass operators are obtained by simple shift and modulation presented in following equation:

$$\widetilde{G}(z) = zH(-z) \quad \text{and} \quad G(z) = z^{-1}\widetilde{H}(-z). \tag{9.11}$$

The wavelet transform has a continuous-time domain interpretation that involves the scaling functions $\widetilde{\varphi}(x)$ and $\varphi(x)$, which are solutions of two-scale relations with filters $\widetilde{H}(z)$ and $H(z)$, respectively.

The scaling function $\varphi(x)$ associated with the filter $H(z)$ is the L^2- solution (if it exists) of the two-scale relation given by following equation:

$$\varphi(x) = \frac{2}{H(1)} \sum_{k \in Z} h_k \varphi(2x - k). \tag{9.12}$$

While it is usually difficult to obtain an explicit characterization of φ(x) in the time domain, one can express its Fourier transform as a convergent infinite product:

$$\widehat{\varphi}(\omega) = \prod_{k=1}^{\infty} \frac{H(e^{j\frac{\omega}{2^k}})}{H(1)}. \tag{9.13}$$

A simple way to generate a scaling function is to run the synthesis part of the wavelet transform algorithm starting with an impulse. This is often referred as the cascade algorithm [24].

Much of the early works in wavelet theory have been devoted to carrying out the mathematical properties (convergence, regularity, order, etc.) of these scaling functions. Usually, the wavelets themselves do not pose a problem because they are linear combination of the scaling functions:

$$\begin{aligned} \psi(x) &= \frac{2}{H(1)} \sum_k g_k \varphi(2x - k), \\ \widetilde{\psi}(x) &= \frac{2}{\widetilde{H}(1)} \sum_k \widetilde{g}_k \widetilde{\varphi}(2x - k). \end{aligned} \tag{9.14}$$

The corresponding analysis and synthesis wavelet basis functions are $\widetilde{\psi}_{i,k} = 2^{-i/2}\widetilde{\psi}(x/2^i - k)$ and $\psi_{i,k} = 2^{-i/2}\psi(x/2^i - k)$, respectively, where i and k are the translation and scale indices.

A necessary condition for the convergence of (4) to an L^2-stable function $\varphi(x)$ is that the filter $H(z)$ has a zero at $z = -1$. More generally, the refinement filters will have a specified number of «regularity factors», which determine their order of approximation.

For all the wavelet families used in the compression scheme, the following properties were obtained to justify the experimental results.

Frequency response: This property allows us to appreciate the behavior of the analysis and synthesis filters in a graphic way to appreciate the differences among the different wavelet families used.

Approximation Order: The parameter L is the number of factors $(1+z^{-1})$ that divide the transfer function $H(z)$. The approximation order plays a crucial role in wavelet

theory [30]. They imply that the scaling function $\varphi(x)$ reproduces all polynomials of degree less or equal to $n = L - 1$; in particular, it satisfies the *partition of unity* ($\sum_k \varphi(x-k) = 1$) (see Chapter 4). They are also directly responsible for the vanishing moments of the analysis wavelet: $\int x^n \widetilde{\psi}(x)dx = 0$ for $n = 0, 1, 2\ldots, L-1$. Finally, the order L also corresponds to the rate of decay of the projection error as a scale that it goes to zero and indicates the number of coefficients in the filters [31].

The next point concerns the stability of the wavelet representation and its underlying multi-resolution bases. The crucial mathematical property is that it translates the scaling functions and wavelets from Riesz bases [32]. Thus, one needs to characterize their Riesz bounds and other related quantities.

The cross-correlation function is a 2π periodic function $a_{\varphi 1,\varphi 2}(\omega)$ given by

$$a_{\varphi 1,\varphi 2}(\omega) = \sum_{k\in \mathrm{Z}} \widehat{\varphi}_2(\omega + 2k\pi) * \widehat{\varphi}_1(\omega + 2k\pi), \tag{9.15}$$

$$a_{\varphi 1,\varphi 2}(\omega) = \sum_{k\in \mathrm{Z}} e^{-kj\omega} \varphi_{12}(k), \tag{9.16}$$

and associated with the pair $\{\varphi_1, \varphi_2\}$. The corresponding cross-correlation function is given by the following equation: $\varphi_{12}(x) = \int \varphi_2(\xi)\varphi_1(\xi + x)d\xi$.

Thus, one can define a biorthogonal pair $\{\varphi, \widetilde{\varphi}\}$ as a set of scaling functions for which the cross-correlation filter is identity ($a_{\varphi,\widetilde{\varphi}}(\omega) = 1$). Here, we mostly consider the autocorrelation filter, such as $a_{\varphi,\varphi}(\omega)$ also denoted by $a_\varphi(\omega)$.

Riesz Bounds: The tightest upper and lower bounds, $B < \infty$ and $A > 0$, of the autocorrelation filter of $\varphi(x)$ are the Riesz bounds of $\varphi(x)$ and given by $A^2 = \inf\limits_{\omega\in[0,2\pi]} a_\varphi(\omega)$ and $B^2 = \sup\limits_{\omega\in[0,2\pi]} a_\varphi(\omega)$. Equivalently, they satisfy to equations

$$A = \inf_{c\in\ell^2} \frac{\left\|\sum_{k\in \mathrm{Z}} c_k \varphi(x-k)\right\|_{L^2}}{\|c\|_{\ell^2}}, \quad B = \sup_{c\in\ell^2} \frac{\left\|\sum_{k\in \mathrm{Z}} c_k \varphi(x-k)\right\|_{L^2}}{\|c\|_{\ell^2}}. \tag{9.17}$$

The existence of the Riesz bounds ensures that the underlying basis functions are in L^2 and that they are linearly independent (in the ℓ^2 space). The Riesz basis property expresses the equivalence between the L^2-norm of the expanded functions and the ℓ^2-norm of their coefficients in the wavelet or scaling function basis. There is a perfect norm equivalence (Parseval's relation) if and only if $A = B = 1$, so, in this case the basis is orthonormal.

Projection Cosine: The (generalized) projection angle θ between the synthesis and analysis subspaces V_a and $\widetilde{V}_a$ is defined as [33]

$$\cos\theta = \inf_{f\in\widetilde{V}_a} \frac{\|\mathrm{P}_a f\|_{L^2}}{\|f\|_{L^2}} = \frac{1}{\sup\limits_{\omega\in[0,2\pi]} \sqrt{a_\varphi(\omega)\cdot a_{\widetilde{\varphi}}(\omega)}}. \tag{9.18}$$

This fundamental quantity is scale-independent, and it allows us to compare the performance of the biorthogonal projection $\widetilde{\mathrm{P}}_a$ with that of the optimal least squares solution P_a for a given approximation space V_a. Specifically, we have the following sharp error bound [34]:

$$\forall f \in L^2, \|f - \mathrm{P}_a f\|_{L^2} \leqslant \left\|f - \widetilde{\mathrm{P}}_a f\right\|_{L^2} \leqslant \frac{1}{\cos\theta} \|f - \mathrm{P}_a f\|_{L^2}. \tag{9.19}$$

The projection angle θ between the synthesis and analysis subspaces should be 90 degrees in orthogonal spaces. In other words, the biorthogonal projector will be

essentially as good as the optimal one (orthogonal projector onto the same space) provides that $\cos\theta$ is close to one.

Simulation results: We carried out numerous simulated experiments to compare the performance of the compression algorithm using different wavelet functions (classical and based the AFs). Firstly, let us present the experimental results applying the symlet family compared with three different families of wavelets based on the AFs that gives the best performance for ultrasound images compression. Secondly, we present the experimental results applying Daubechies family compared with three different families of wavelets based on the AFs that gives the best performance for mammography images compression.

In the compression procedure, we use five decomposition levels.

Figures 9.11 and 9.12 present visual results of compressed US and MG images, respectively. The original and error image, which were amplified by 40 times, expose the compression procedure quality.

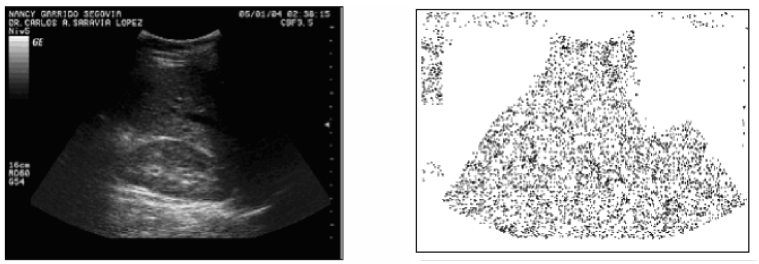


Fig. 9.11. Compressed US image and error image amplified by 40 times.

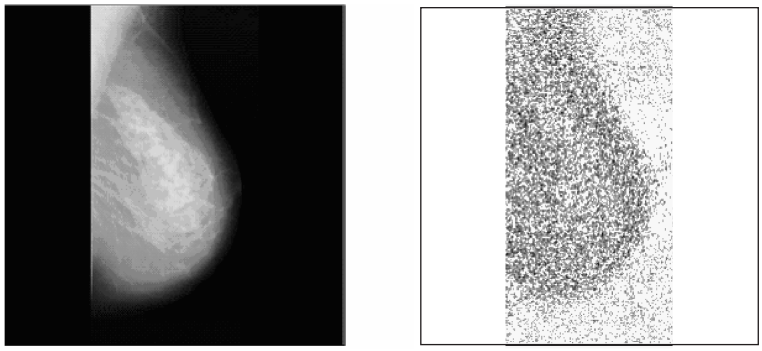


Fig. 9.12. Compressed MG image and error image amplified by 40 times.

Figures 9.13 and 9.14 present the results obtained for MAE criteria with symlets, based on AF $\mathrm{up}(t)$, on WA $\mathrm{fup}_2(t)$, and on WA $eup(t)$ wavelets, respectively for ultrasound images.

Figures 9.15 and 9.16 present the results obtained for CR criteria, with symlets, WA $\mathrm{up}(t)$, WA $\mathrm{fup}_2(t)$, and WA $eup(t)$ wavelets, respectively for ultrasound images.

Figures 9.17 and 9.18 present the results obtained for MAE criteria, with Daubechies, WA $\mathrm{up}(t)$, WA $\mathrm{fup}_2(t)$, and WA $eup(t)$ wavelets, respectively for mammography images.

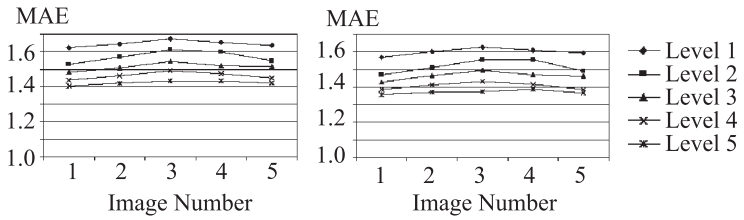


Fig. 9.13. MAE criterion for compressed ultrasound images with symlets and WA $\mathrm{up}\,(t)$ wavelets, respectively (left to right).

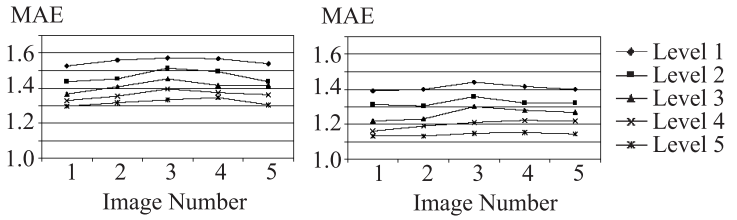


Fig. 9.14. MAE criterion for compressed ultrasound images with WA $\mathrm{fup}_2(t)$ and $\mathrm{eup}(t)$ wavelets, respectively (left to right).

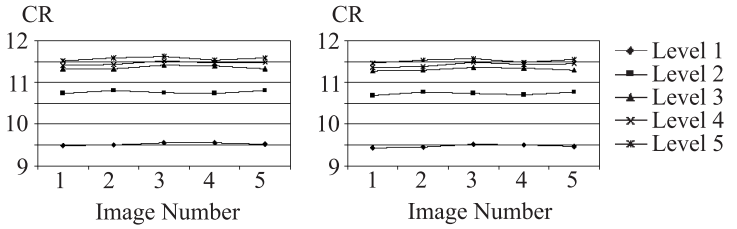


Fig. 9.15. CR criterion for compressed ultrasound images with symlets and WA $\mathrm{up}\,(t)$ wavelets, respectively (left to right).

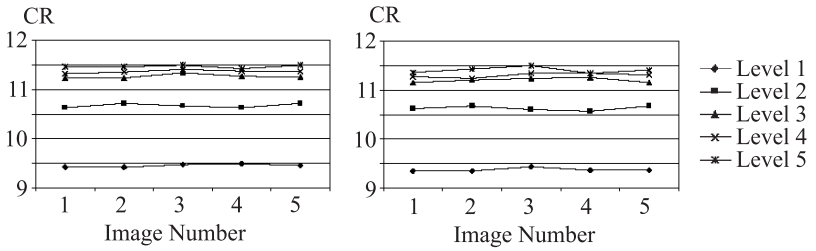


Fig. 9.16. CR criterion for compressed ultrasound images with WA $\mathrm{fup}_2(t)$ and $eup(t)$ wavelets, respectively (left to right).

Figures 9.19 and 9.20 present the results obtained for CR criteria, with Daubechies, based on the AF $\mathrm{up}\,(t)$, based on the AF $\mathrm{fup}_2(t)$, and based on the AF $eup(t)$ wavelets, respectively for mammography images.

Figure 9.21 presents the frequency response for wavelet 9/7 (solid line), Daubechies 8 (dotted line), and symlet 8 (dashed line).

Figure 9.22 presents the frequency response for Kravchenko–Rvachev wavelets based on the atomic functions $\mathrm{up}\,(t)$ (solid line), $\mathrm{fup}_2(t)$ (dotted line), and $eup(t)$ (dashed line).

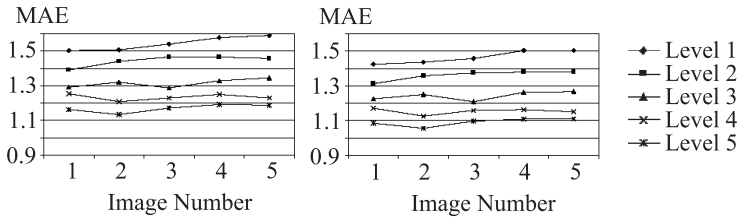


Fig. 9.17. MAE criterion for compressed mammography images with symlets and WA $\mathrm{up}(t)$ wavelets, respectively (left to right).

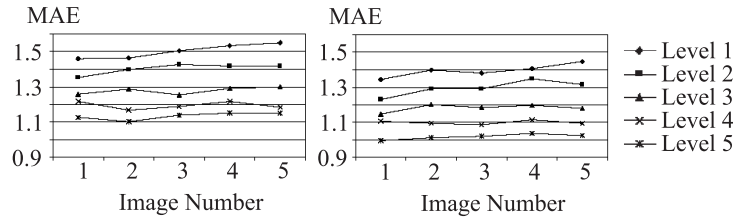


Fig. 9.18. MAE criterion for compressed mammography images with based on AF $\mathrm{fup}_2(t)$ and $eup(t)$ wavelets, respectively (left to right).

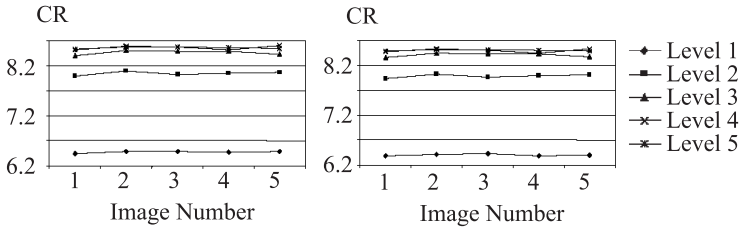


Fig. 9.19. CR criterion for compressed mammography images with Daubechies and WA $\mathrm{up}(t)$ wavelets, respectively (left to right).

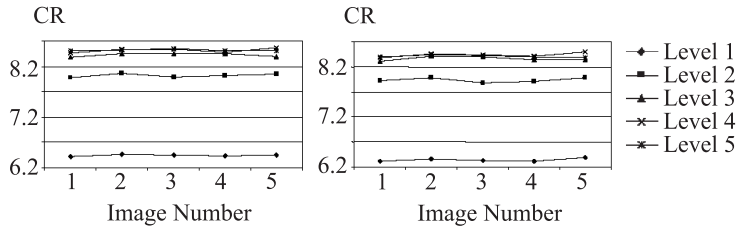


Fig. 9.20. CR criterion for compressed mammography images with WA $\mathrm{fup}_2(t)$ and $eup(t)$ wavelets, respectively (left to right).

Finally, Table 9.8 presents the key properties of the different wavelets used in compression of the US and MG images.

It is known from statistical theory that the approximation property of estimation of random variable is characterized by relative error $\delta = 2(1 - r)$, where r is correlation coefficient that is equal to projection cosine in this case. So, calculations of this error show that wavelet based on $eup(x)$ can potentially produce relative variance error of 0.00464 (6.8% in RMS value), and at same time wavelet *Daubechies 8* gives value of

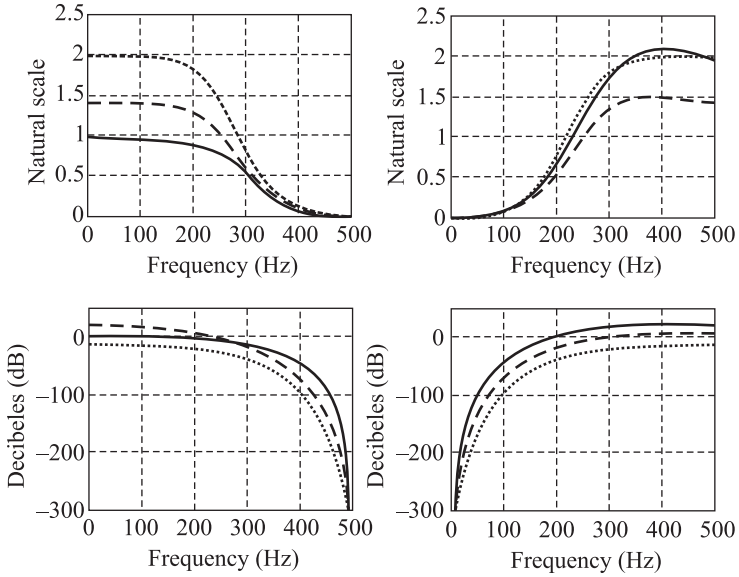


Fig. 9.21. Frequency response: wavelet 9/7 (solid line), Daubechies 8 (dotted line), and symlet 8 (dashed line).

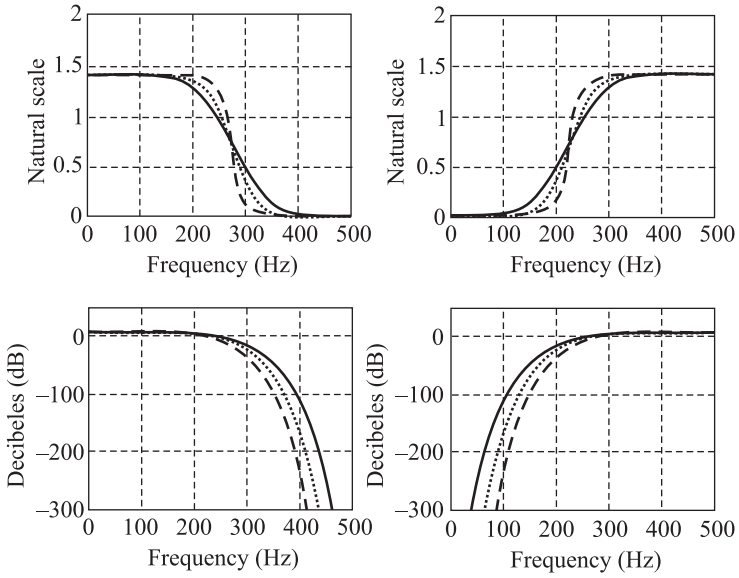


Fig. 9.22. Frequency response: based on the atomic functions $\mathrm{up}\,(t)$ (solid line), $\mathrm{fup}_2(t)$ (dotted line), and $eup(t)$ (dashed line).

0.02242 (more than 15% in RMS value), and *wavelet 9/7* value of 0.03234 (more than 18% in RMS value). So, wavelet based on $eup(x)$ gives about three less relative error in RMS values than *wavelet 9/7* that is the basic wavelet used in JPEG2000 standard.

Numerous tests were carried out in comparing of different wavelets functions (classical and WA) to choose the best one for compression of medical US and MG

Table 9.8. Summary of key properties of different wavelet families [44].

Key properties for different Wavelet filters						
Type	Wavelet 9/7		Daubechies 8		Symlet 8	
	Dec.	Rec.	Dec.	Rec.	Dec.	Rec.
Approximation Order	4		4		4	
Projection cosine	0.98387		0.98879		0.98781	
Riesz Bounds	0.926	0.943	0.833	0.849	0.880	0.896
	1.065	1.084	1.267	1.290	1.273	1.295
Type	A.F.W. $\mathrm{up}(t)$		A.F.W. $\mathrm{fup}_2(t)$		A.F.W. $eup(t)$	
	Dec.	Rec.	Dec.	Rec.	Dec.	Rec.
Approximation Order	4		4		4	
Projection cosine	0.99176		0.99472		0.99769	
Riesz Bounds	0.792	0.806	0.713	0.726	0.641	0.653
	1.514	1.542	1.802	1.834	2.145	2.183

images. The compression algorithm based on average threshold has shown better detail preservation, presenting the best MAE, but the compression rate decreases from 4 to 5 times approximately in comparison with the universal and Bayes shrink thresholds that can maintain a good quality image (low MAE). The top threshold presents a low MAE values but also low compression rate values. Finally, it is advisable to use the universal threshold that exposed the best *CR* and can maintain the quality of the image in comparison with other threshold methods.

Also, it is observed that for two modalities of images used in the tests the wavelet families based on the atomic functions presented smaller levels of mean absolute error, with relationship to their equivalent of the Daubechies and symlets families.

Frequency response analysis has shown that Daubechies and symlet filters are more selective than the Wavelet 9/7 filters used in compression standard JPEG2000. It has been observed that the WA filters have an answer of respond function in more selective frequency than the filters of the traditional families, and WA system $eup(x)$ presented the best frequency response among all investigated wavelets.

Since the approximation order has been taken the same for all used filters, this derives mainly in two things, the applying filters have the same number of coefficients and, therefore they imply the same computational complexity when being implemented in the compression algorithm. Also, the convergence of the error should be of the same order for all the filters.

The existence of the limits Riesz bounds demonstrates that the coefficients of the analysis and synthesis filters are lineally independent. The found projection cosine shows that the WA systems are close to the ideal value. This implies that they are better than semi-orthogonal and the «most independent». Also, the fact that they are lineally independent ensures that errors does not accumulate in the decomposition/reconstruction procedure.

These properties appreciate that the families of wavelets based on atomic functions have sufficiently better acting than the traditional families. Also, it is appreciated that the WA system $eup(x)$ has a better acting.

Finally, it is advisable to use the WA systems because they present the best image quality and maintain the compression rate almost equal. Likewise the different properties

calculated for the wavelet filters demonstrate that the WA filters should have a better acting that the traditional wavelets. The WA system $eup(x)$ presents better levels of MAE criterion for two image modalities.

9.4.2. Wavelet Transform and Neural Network in Classification of Mammography. Several research groups have focused on the development of computerized systems that can have different types of medical images and extract useful information for the medical professional [20–22, 25].

Mammography (MG) is currently the only proven and cost-effective method to detect early breast cancer. Because of the small sizes of Microcalcifications (MCs) and the relatively noisy MG background, subtle MCs can be missed by radiologists. Computerized methods for detection of MCs have been developed in a number of works [35–40]. MG interpretation can be considered a two-step process. A radiologist first screens the MGs for abnormalities. If a suspicious abnormality is detected, further diagnostic workup is then performed to estimate the likelihood that the abnormality is malignant.

Computer-aided detection (CAD) is considered to be one of the promising approaches that may improve the efficacy of MG. CAD lesion detection can be used during screening to reduce oversight of suspicious lesions that warrant further diagnostic workup. CAD represents one of the most successful paradigms of medical-image analysis by incorporating most of the significant developments that have occurred in enhancement and segmentations of candidate features, in feature extraction and classification, and reduction or characterization of false positives.

Mammography is one of the radiological fields where CAD systems have been widely applied because the demand for accurate and efficient diagnosis is so high. The presence of abnormalities of specific appearance could indicate cancerous circumstances, and their early detection improves the prognosis of the disease. The principal stages of a typical CAD scheme are: *preprocessing, segmentation, feature analysis* (*extraction, selection, and validation*), and *classification* utilized either to reduce false positives (FPs) or to characterize abnormalities. Figure 9.23 shows a typical CAD scheme. In the stages of this method, spot-like characteristics in the original X-ray image are enhanced before undergoing border detection and filling to distinguish them from the background. A description of the methods employed in each stage is given below.

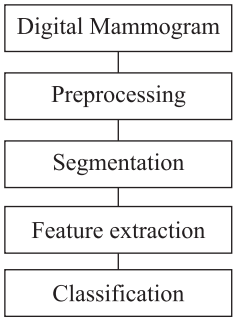


Fig. 9.23. CAD architecture.

9.4.2.1. Processing Procedures.

At the first stage, the subtle features of interest are enhanced and the unwanted characteristics of the MG image are de-emphasized. The enhancement procedure results in a better description of the objects of interest (MCs). The enhancement of the

contrast of the regions of interest, the sharpening of the abnormalities boundaries, and suppression of noise are considered in this stage.

The MG images are extracted from the MIAS database [21] where images have the detailed information, which includes the characteristics of background tissue (fatty, fatty-glandular, or dense-glandular), and a class of abnormality (calcification, masses, and speculated masses). It has been used in analysis of 30 MG images with presence of MCs. The procedure that consists of employing of the WT analysis using *Daubechies (db), Symlets (sym), Coiflets (coif), and Biorthogonals (bior)* filters has been proposed.

The algorithm that consists of the creation of the negative image starting from the original MG and applying WT to the negative image is used. We employ WT families: *db2, db4, db8*, and *db16* filters, *sym1, sym2, and sym4* filters, *coif1, coif2, and coif4* filters, *bior1.1, bior2.2*, and *bior4.4* filters with one decomposition level for negative image. In this step, the approximation image coefficients are obtained and they are denoted as an AC image. Figure 9.24 shows one case of original MG image and their correspondent negative image, and AC image of Wavelet decomposition.

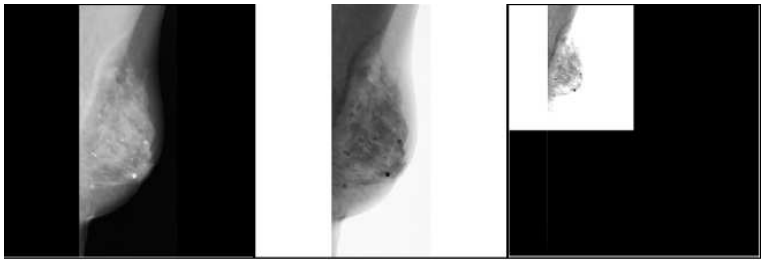


Fig. 9.24. Original MG image (left), Negative MG image (center), and wavelet decomposition of negative MG image (right).

Segmentation is a decision process based on the preceding image processing task to extract suitable features, and this is the second stage of the proposed algorithm. Traditional approaches to segment images include three classes of techniques: pixel-based, region-based, and edge-based segmentation techniques [40]. The AC image is segregated into separate parts, each of which has similar properties. The image background, the tissue area, and other areas can be separated because they are characterized using generic features.

After the image enhancement, the background gray level of the MGs is relatively constant. This facilitates the segmentation of the individual MCs from the background.

Region based methods focus our attention on an important aspect, these techniques classify a pixel as an object pixel judging solely on its gray value independent of the context. This means that any isolated points or small areas could be classified as being objects pixels, despite the fact that an important characteristic of an object is its connectivity. These characteristics are important to consider them in the MCs classification.

Segmentation Algorithm. Because the MG images present visual information with diverse textures in the region of the breast, the analysis has been carried out using an algorithm that allows identifying textures. Algorithm is based on neighborhood operations and these tend to blur edge regions, as edge pixels are combined with structural segment pixels. Image features related to texture can be particularly useful in segmentation.

MG shows the regions that have approximately the same average intensity values, but are readily not distinguished visually because of differences in texture. Several

neighborhood-based operations can be used to distinguish textures: the small segment Fourier transform, local variance (or standard deviation), the Laplacian operator, the rank operator (that uses the difference between maximum and minimum pixel values in the neighborhood), the Hurst operator (maximum difference as a function of pixel separation), and the Haralick operator (a measure of distance moment). Here, the nonlinear rank filter to convert the textural patterns into differences in intensity has been applied. The rank operator is a sliding neighborhood procedure in a 3×3 window that uses the difference between the maximum and minimum pixel values with a neighborhood.

The regions are now clearly visible as intensity differences and can be isolated by thresholding. Histogram of MG image after nonlinear filtering provides the applied threshold values. After filtering, the intensity regions are clearly seen.

9.4.2.2. Feature extraction and classification

In this stage, the MCs patterns of the segmented MG image were obtained. In any segmentation approach, a considerable number of normal objects are recognized as pathological, which results in reduced efficiency of the diction system.

To improve the performance of the scheme, several image features are calculated in an effort to describe the specific properties or characteristics of each MC pattern. The most descriptive of these features are processed by a classification system to make an initial characterization of the segmented samples. Although the number of calculated features derived from different feature spaces is quite large, it is difficult to identify the specific discriminative power of each one. Thus, a primary problem is the selection of an effective feature set that has high ability to provide a satisfactory description of the segmented regions.

Many useful image features have been suggested by the image processing and pattern analysis communities [25, 41]. These features can be divided into three categories, *intensity features, geometric features, and texture features,* whose values should calculated from the pixel matrices of the region of interest (ROI). The MCs patterns obtained at segmentation stage are ROIs. In Fig. 5.25, it is presented foreground of a suspected MC.

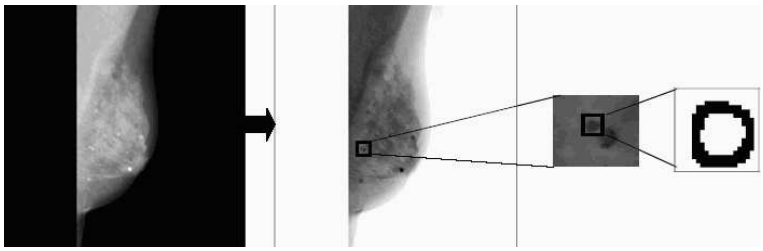


Fig. 9.25. Foreground of a suspected MC.

Table 9.9 provides a list of typical mathematical features of individual MC and their clusters.

The histograms of the feature point distribution extracted from true and false mass regions are studied, and the features that can better separate the true and false mass regions are selected for further study.

From our experience, it has been suggested that three features: *the site area*, *two measures of compactness*, and *difference entropy* led to better discrimination and reliability [42, 43].

Table 9.9. Summary of mathematical features.

Feature subspace	Features
Intensity features	1. Contrast measure of ROI 2. Standard derivation inside ROI 3. Mean gradient of ROI's boundary
Geometric features	1. Area measure 2. Compactness
Texture features	1. Energy measure 2. Correlation 3. Inverse difference moment 4. Sum average 5. Sum entropy 6. Difference entropy

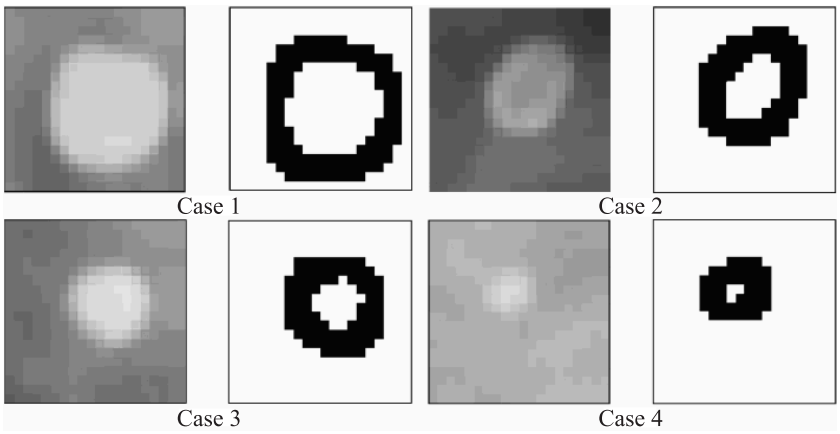


Fig. 9.26. MC pattern original image (left) and MC segmented image (right).

Figure 9.26 exposes five cases of MC patterns of the original images and corresponding MCs segmented ones.

A *classification system* is an essential part of a CAD system. The classifiers that are utilized in the area of the detection of mammographic MCs are those employed in most of the medical image-analysis procedures.

An artificial neural network (ANN) is a structure that can be adjusted to produce a mapping of relationships among the data from a given set of features. The main steps in using ANN are: first, a neural network structure is chosen in a way that should be considered suitable for the type of the specific data and the underlying process to be modeled. Then, the ANN is trained using a training algorithm and a sufficiently representative set of data (training data set). Finally, the trained network is evaluated with different data (test data set), from the same or related sources, to validate that the acquired mapping is of acceptable quality.

The ANN structure was chosen according to the results obtained with another type of images [42] that used ANN of backpropagation with three layers.

A group of patterns of 200 MCs during the segmentation process was obtained. The ANN was trained using a LMS training algorithm and 100 patterns data. The training of the ANN was evaluated with other different set of data of 100 MCs patterns.

9.4.2.3. Simulation Results [42, 43].

The best results applying WT were obtained for *db4, db8, db16, sym1, sym2*, and *sym4* functions. The proposed algorithm for segmentation of MG images has been proven employing 30 images, where separation of the region of interest was obtained with accuracy inside the area breast.

For the classification of MCs, there were considered its segmented patterns. Each segmented pattern of MC indicates the presence of a malignant tumor in the MG images. It has been considered two test types: the first one, considering the features of the MC patterns and the second one, the MC patterns that are used only for classification.

In Table 9.10, the training and test percentages considered from two features to eleven are presented.

Table 9.10. Performance of the MLP classifiers for number of features (training percentage and test percentage).

No. Feature	% Training	% Test
2	75.2	74.0
3	77.0	76.2
4	78.6	76.0
5	82.0	80.0
6	85.4	82.5
7	88.0	85.0
8	92.0	90.0
9	95.2	93.5
10	95.0	94.0
11	100	95.0

Table 9.11. Performance of the MLP classifiers (training percentage, test percentage, and computational iterations).

Architecture	% Training	% Test	Iterations
400:10:2	95.2	91.2	13000
400:20:2	96.4	92.4	10000
400:30:2	97.3	93.5	8000
400:40:2	98.6	94.0	8200
400:50:2	99.0	95.5	7900
400:60:2	99.5	97.0	7500
400:70:2	99.5	96.8	7100
400:80:2	100	97.2	7100
400:90:2	100	98.0	7000
400:100:2	100	97.5	7000
400:500:2	100	97.8	5700

The architectures are presented among brackets with three numbers separated by two points. Here, the first number corresponds to size of the pattern of MC, the second one is a number of the hidden nodes in the structure ANN, and third one is a number of identified classes.

Table 9.11, presenting the MLP type classifiers, exposes that the best results were obtained for [400:60:2] and [400:90:2] architectures. The efficient performance obtained for an MLP classifier was 100% and 98% for training and testing, respectively.

So, the proposed and implemented method is based on WT, segmentation, and MLP classifiers for MG medical image. It permits to reduce the iterations number during the training of the neural network MLP applying WT. The wavelet functions: *Daubechies, Symlet, Coiflet,* and *biorthogonal* have been employed in MLP network for microcalcifications classification in the MG images. The experimental results have shown good performance of the implemented algorithms [37, 38].

References

1. *Kehtarnavaz, N. and Gamadia, M.* Real-Time Image and Video Processing: From Research to Reality. Morgan and Claypool Publ. 2005.
2. *Xilink Inc.*, Xilink Data Book, 2004. http://www.xilink.com
3. *Altera Inc.*, Altera Data Book, 2005. http://www.altera.com
4. *Motorola*. http://www.motorola.com
5. *Texas Instruments Inc.*, TMS320C6000 Code Composer Studio Tutorial, SPRU301C, Texas Instruments Incorporated, 2005.
6. *Charles, E.C. and Marvin, B.* Radar Signals: An introduction to theory and application, Artech House, 1993.
7. *Skolnik, M.I.* Radar handbook, McGraw Hill, 1990.
8. *Natashon, F.E. and Patrick, Reilly J.* Radar design principles, Scitech Publ., INC, 1999.
9. *Nuttall, A.H.* Some windows with very good side-lobe behavior. IEEE Trans. on acoustic, speech and signal processing, ASSP, 1981, vol. 29, no. 1, pp. 84–91.
10. *Harris, F.J.* On the use of window for harmonic analysis with the DFT. *IEEE Proc.*, 1978, vol. 66, no. 1, pp. 51–83.
11. *Kravchenko, V.F.* New synthesized windows. Doklady Physics, 2002, vol. 47, no. 1, pp. 51–60.
12. *Maxfield, C.M.* The design warrior's guide to FPGA's, Newnes, 2004.
13. *Kravchenko, V.F., Ponomaryov, V.I., Smirnov, D.V., Escamilla-Hernandes, E., and Loboda, I.I.* Application of the Kravchenko–Rvachev Functions in Radar Systems for Resolution of the Large Number of Targets (in Russian). Edit. Radioteknika, 2005, no 11, pp. 46–54.
14. *Escamilla-Hernandez, E., Kravchenko, V.F., Ponomaryov, V.I., and Smirnov, D.V.* Radar Processing on Basis of Kravchenko–Rvachev's Functions for Resolving of Numerical Targets. Telecommunications and Radio Engineering, Begell House Edit. 2006, vol. 65, no 3, pp. 209–224.
15. *Escamilla, E., Ikuo, A., Endo, H., and Ponomaryov, V.* Performance of weighting function to reduce side-lobes in pulse compression radar. Proc. of the 10th UEC Int. mini-conference for ES., *UEC*, 2003, Tokyo, Japan, pp. 21–26.
16. *Ponomaryov, V. and Escamilla, E.* Real-Time Windowing in Imaging Radar Using FPGA Technique. Proc. of the Conference Real-time imaging IX, San Jose, California, USA, 2005, Proc. SPIE, vol. 5671, pp. 152–161.
17. *Kravchenko, V.F. and Basarab M.A.* A New Method of Multidimensional-Signal Processing with the Use of R-Functions and Atomic Functions. Doklady Physics, 2002, vol. 47, no. 3, pp. 195–200.
18. *Kravchenko, V.F., et al.* New Constructions of Weight Windows Based on Atomic Functions in Problems of Speech-Signal Processing. Doklady Physics, 2001, vol. 46, no. 3, pp. 166–172.

19. http://www.xilinx.com/products/design_resources/dsp_central/grouping /index.htm.
20. *Jan, J.* Medical Image Processing, Reconstruction and Restoration: Concepts and Methods, New York Taylor & Francis, CRC Press, 2006.
21. Mammographic Image Analysis Society (MIAS). MiniMammography Database; available on-line at: http://www.wiau.man. Ac.uk/services/MIAS /MIASmini.html.
22. *Jähne, B.* Practical Handbook on Image Processing for Scientific and Technical Applications, second Ed., CRC Press 2004.
23. *Antonini, M., Barlaud, M., Mathieu, P., and Daubechies, I.* Image coding using wavelet transform. IEEE Trans. Image Process. 1992, vol. 1, no. 2, pp. 205–220.
24. *Mallat, S.* A theory for multiresolution signal decomposition: The wavelet decomposition. IEEE Trans. Pattern Anal. Mach. Intell., 1989. vol. 11, no. 7, pp. 674–693.
25. *Semmlow, J.L.* Biosignal and Biomedical Image Processing: MATLAB-Based Applications. Signal Processing and Communication series. Ed. Marcel Dekker, 2004.
26. *Ponomaryov, V., Sanchez, J.L., Juarez, C., and Nino-de-Rivera, L.* Evaluation of the JPEG2000 Standard for Compression of Ultrasound and Mammography Images. Proc. of 18th Biennial International EURASIP Conference. Biosignal. 2006, pp. 287–289, Czech Republic.
27. *Ponomaryov, V.I., Sanchez-Ramirez, J.L., and Juarez-Landin, C.* Evaluation and Optimization of the Standard JPEG2000 for Compression of Ultrasound Images. Telecommunications and Radio Engineering, 2006, vol. 65, no. 11, pp. 1005–1017, Begell House Edit. USA.
28. *Sanchez, J.L. and Ponomaryov V.* Wavelet Compression Applying Different Threshold Types for Ultrasound and Mammography Images. GESTS Intern. Trans on Computer Sciences and Engineering, 2007, vol. 39, no. 1, pp. 15–24, South Korea.
29. *Gulyaev, Yu.V., Kravchenko, V.F., and Pustovoit, V.I.* A New Class of WA-Systems of Kravchenko–Rvachev Functions. Doklady Mathematics, 2007, vol. 75, no. 2, pp. 325–332.
30. *Vetterli M. and Kovacevic J.* Wavelets and Subband Coding, Prentice-Hall, Englewood Cliffs, NJ, 1995.
31. *Villemoes L.* Wavelet analysis of refinement equations. SIAM J. Math. Anal., 1994, vol. 25, no. 5, pp. 1433–1460.
32. *Meyer Y.* Ondelettes, Hermann, Paris, France, 1990 (in French).
33. *Unser M. and Aldroubi* A. A general sampling theory for nonideal acquisition devices. IEEE Trans. Signal Process., 1994, vol. 42, no. 11, pp. 2915–2925.
34. *Strang G. and Nguyen T.Q.* Wavelets and Filter Banks. Wellesley-Cambridge Press, Cambridge MA, 1996.
35. *Chan H.P., Doi K., and Vyborny C.J.* Improvement in Radiologists Detection of Clustered Microcalcifications, on Mammograms: the Potential of Computer-Aided Diagnosis. Acad. Radiol., 1990, vol. 25, p. 1102.
36. *Fam B.W., Olson S.L., and Winter P.F.* Algorithm for the Detection of Fine Clustered Calcifications on Film Mammograms. Radiology, 1998, vol. 169, p. 333.
37. *Qian W. and Carke L.P.* Tree-Structured Nonlinear Filter and Wavelet Transform for Microcalcification Segmentation in Mammography. Proc. SPIE Medical Imaging, 1993, vol. 1905, p. 716.
38. *Zhang W., Doi K., and Giger M.L.* Computerized Detection of Clustered Microcalcifications in Digital Mammograms using a Shift-Invariant Artificial Neural Network. Medical Physics, 1994, vol. 21, p. 517.
39. *Chan H.P., Wei D., Helvie M.A, Shiner B., Alder D.D., Goodsitt M.M., and Petrik N.* Computer-Aided Mammographic Screening for Speculated Lesions. Radiology, 1994, vol. 191, pp. 331–337.
40. *Fua J.C., Lee S.K., Wong S.T.C., Yeh J.Y., Wang A.H., and Wu H.K.* Image Segmentation Feature Selection and Pattern Classification for Mammographic Microcalcifications. Computerized Medical Imaging and Graphics, 2005, vol. 29, pp. 419–429.
41. *Hen-Hu, Y. and Hwang, J.N.* Handbook of Neural Network Signal Processing. The Electrical Engineering and Applied Signal Processing Series, N.Y., CRC Press 2002, pp. 12–1 to 12–15.

42. *Juarez, C., Castillo, M.E., and Ponomaryov, V.* Classification of masses in mammography images using wavelet transform and neural networks. In Proc. of 6^{th} International Kharkov Symposium on Physics and Engineering of Microwaves, Millimeter, and Submillimeter Waves (MSMW'07), 2007, vol. 2. pp. 956–958.
43. *Juarez-Landin, C. and Ponomaryov, V.* Classification of Microcalcifications using Wavelet Transform and neural Networks. GESTS Intern. Trans. on Computer Sciences and Engineering. 2007, vol. 39, no. 1, pp. 112–120, South Korea.
44. *Kravchenko, V.F., Ponomaryov, V.I., and Sanchez-Ramirez J.L.* Properties of Different Wavelet Filters used for Ultrasound and Mammography Compression. Telecommunications and Radio Engineering, 2008, vol. 87, no. 10, pp. 853–865.

Chapter 10

TRANSVERSAL ADAPTIVE FILTERS

The transversal FIR adaptive filters have been successfully used in the solutions of many practical problems due to their unconditional stability. The mean-square error surface of such filters is unimodal, which guarantees the convergence to the global minimum. This chapter presents an analysis of transversal FIR adaptive filter structure along with some of its most widely used adaptive algorithms.

10.1. Introduction

An adaptive filter, as shown in Fig. 10.1, is a linear system with two inputs and two outputs whose parameters are updated to minimize some given criterion of the difference between the filter output y(n) and the reference signal d(n). Although several criteria have been proposed in literature, the most commonly used one is the mean-square error because, in this situation, the optimal solution of any FIR adaptive filter converges to the solution of the Wiener–Hopf equation [1–3].

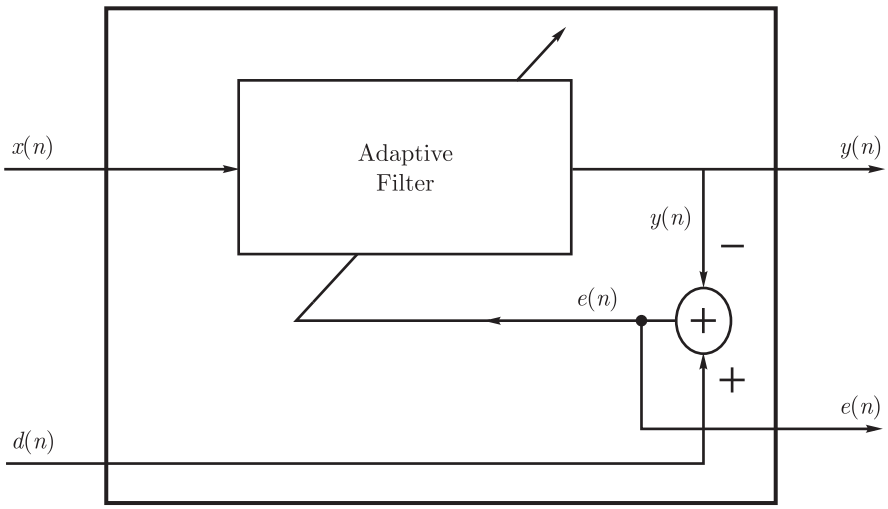


Fig. 10.1. Adaptive filter framework.

Several algorithms have been proposed to update the adaptive filter coefficient vector. They can be divided in two groups. The first group, which is based on the Newton–Rapson approach, provides robustness against additive noise and fast convergence rates, although its computational complexity is too high for most practical applications. The second group, based on gradient-search approach presents a low computational complexity, although its convergence rate is slow and its performance is sensitive to the additive noise. These two approaches and some variants of them are analyzed in the next sections.

10.2. Mean Square Error Surface

A fundamental concept in the development of adaptive filter algorithm is the mean-square error surface (MSE), which is a function of the filter coefficients.

To obtain an expression for the MSE, consider the output signal of a digital filter, which is given by

$$y(n) = \sum_{k=0}^{\infty} w_k x(n-k), \tag{10.1}$$

where the filter coefficients are estimated so that

$$\xi = E\left[|e(n)|^2\right] \tag{10.2}$$

becomes a minimum and

$$e(n) = d(n) - \sum_{k=0}^{\infty} w_k x(n-k). \tag{10.3}$$

From the orthogonality property of the mean square estimation it follows that [1, 3]

$$E\left[\left(d(n) - \sum_{k=0}^{\infty} w_k x(n-k)\right) x(n-j)\right] = 0, \tag{10.4}$$

$$E\left[d(n)x(n-j)\right] = \sum_{k=0}^{\infty} w_k E\left[x(n-k)x(n-j)\right], \tag{10.5}$$

$$\varphi_{xd}(j) = \sum_{k=0}^{\infty} w_k \varphi_{xx}(j-k). \tag{10.6}$$

Then, from (10.6), the MSE is given by

$$MSE = \sigma_d^2 - \sum_{k=0}^{\infty} w_k \varphi_{xd}(k). \tag{10.7}$$

Next, taking the z transform of (10.7), we obtain

$$MSE = r_{dd} + \frac{1}{2\pi j} \oint \left[W(z^{-1}) G_{xx}(z) - 2G_{dx}(z)\right] W(z) \frac{dz}{z}, \tag{10.8}$$

where W(z) is the FIR (finite impulse response) transfer function, $G_{xx}(z)$ is the spectral density function of the input signal, and $G_{dx}(z)$ is the cross-spectral density power between the input and reference signals. If W(z) is an arbitrary pole-zero function, two important situations must be considered: a) The system of poles may move out of the unit circle during the apdatation process and b) the MSE may present local minima or flat zones that may hamper the adaptation process [1, 3]. Figure 10.2 shows the MSE of an arbitrary pole-zero system.

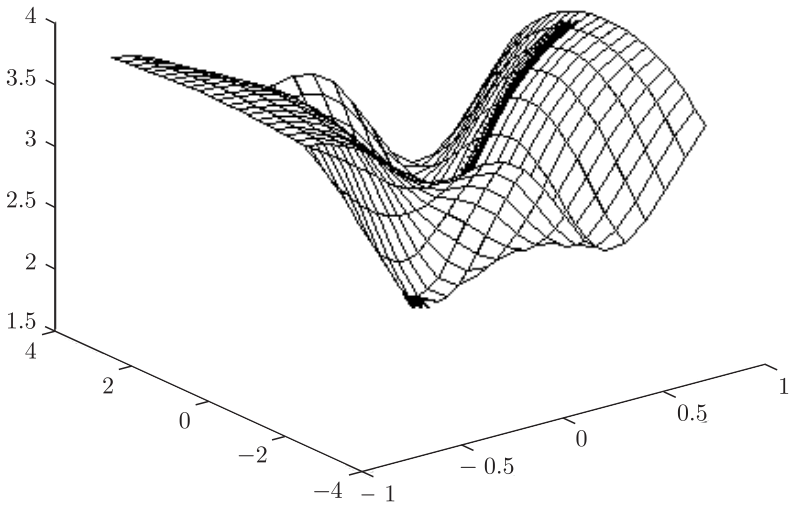


Fig. 10.2. MSE of a second order pole-zero system.

Since the all-zero systems are more widely used than the pole-zero ones, their MSE surface is analysed with more detail in the next subsection.

10.2.1. MSE Surface for all-Zero Systems. The MSE surface of an all-pole system can be obtained by substituting the transfer fuction of an all-pole, or finite impulse response FIR, system of order N, which is given by

$$W(z) = \sum_{k=0}^{N-1} w_k z^{-k} \tag{10.9}$$

into (10.8) to obtain

$$MSE = r_{dd} + \sum_{m=0}^{N-1}\sum_{k=0}^{N-1} w_m w_k \frac{1}{2\pi j}\oint z^m z^{-k} G_{xx}(z)\frac{dz}{z} - \\ - 2\sum_{m=0}^{N-1} w_k \frac{1}{2\pi j}\oint z^{-k} G_{dx}(z)\frac{dz}{z}\,. \tag{10.10}$$

Finally, taking the invesrse z transform, we obtain

$$MSE = r_{dd}(o) + \sum_{k=0}^{N-1}\sum_{m=0}^{N-1} w_k w_m r_{xx}(k-m) - 2\sum_{K=0}^{N-1} w_k r_{dx}(k). \tag{10.11}$$

Equation (10.11) shows that the MSE is a cuadratic function of the filter coefficients, which means that there exists only a global minumum and, therefore, a simple gradient algorithm can be used to obtain the optimum filter coefficients [1]. Figure 10.3 shows the MSE of a FIR system.

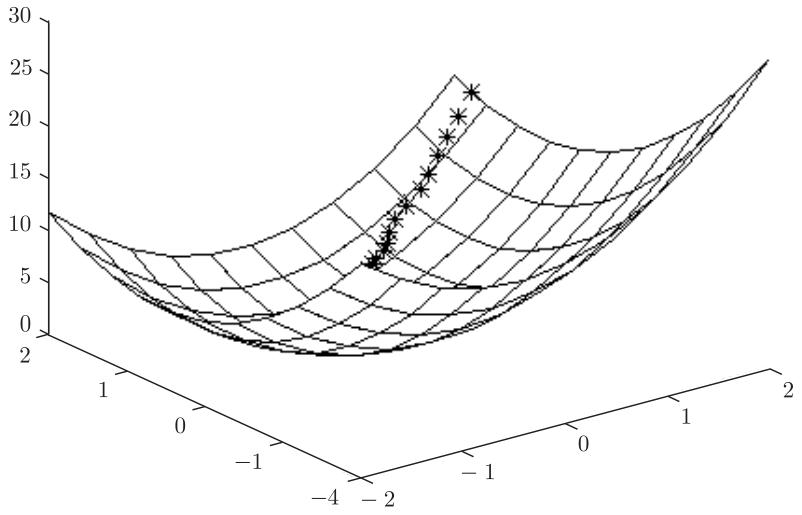


Fig. 10.3. FIR adaptive filter MSE surface.

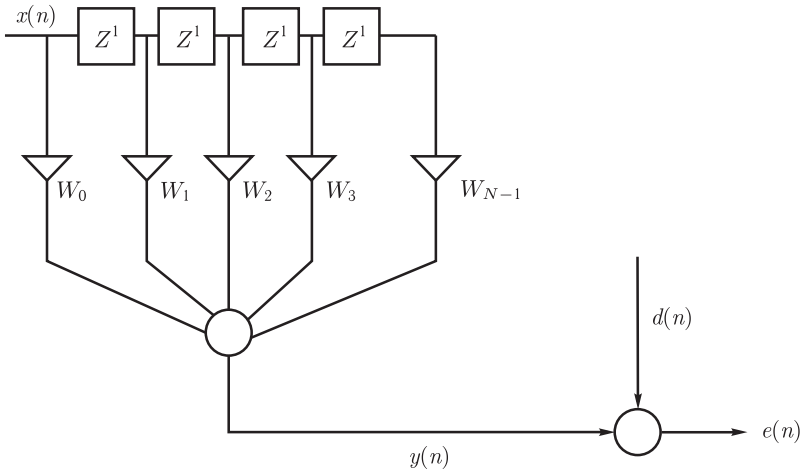


Fig. 10.4. FIR adaptive filter structure.

10.3. Recursive Least-Square Algorithms

One of the most widely used FIR adaptive filter algorithms is the recursive least-square algorithm [1, 2], which directly minimizes the mean-square error. Consider the adaptive filter output, which is given by

$$y(n) = \mathbf{W}^T\mathbf{X}(n) = \mathbf{X}^T(n)\mathbf{W}, \tag{10.12}$$

where

$$\mathbf{X}(n) = [x(n), x(n-1)\ldots, x(n-N+1)]^T, \tag{10.13}$$

$$\mathbf{W} = [w_0, w_1, w_2, \ldots, w_{N-1}]^T \tag{10.14}$$

is the coefficients vector, which will be estimated such that the MSE, $E[e^2(n)]$, attains a minimum, where

$$e(n) = d(n) - y(n), \tag{10.15}$$

$$E[e^2(n)] = E[(d(n) - y(n))^2]. \tag{10.16}$$

To minimize (10.16), we can use the orthogonality principle in MSE estimation, according to which the optimum vector is obtained from the condition that the input vector is orthogonal to the error signal. Thus, from (10.12), (10.13), and (10.16), it follows that [1]

$$E[\mathbf{X}(n)(d(n) - \mathbf{X}(n)^T\mathbf{W})] = 0, \tag{10.17}$$

where

$$E[\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{W}] = E[d(n)\mathbf{X}(n)], \tag{10.18}$$

$$E[\mathbf{X}(n)\mathbf{X}^T(n)]\mathbf{W} = E[d(n)\mathbf{X}(n)]. \tag{10.19}$$

Assuming that the coefficients and input vector are mutually uncorrelated, we obtain

$$\mathbf{RW} = \mathbf{P}, \tag{10.20}$$

where

$$\mathbf{P} = E[d(n)\mathbf{X}(n)] \tag{10.21}$$

is the correlation vector between the reference signal and the input vector $\mathbf{X}(n)$, and

$$\mathbf{R} = E[\mathbf{X}(n)\mathbf{X}^T(n)] \tag{10.22}$$

is the input signal autocorrelation matrix. The next assumption is that the input and reference signals are ergodic. Then, $\mathbf{P}(n)$ can be estimated as follows

$$\begin{aligned} \mathbf{P}(n) &= \sum_{k=0}^{n-1} \lambda^{n-k} d(k)\mathbf{X}(k) + d(n)\mathbf{X}(n), \\ \mathbf{P}(n) &= \sum_{k=0}^{n} \lambda^{n-k} d(k)\mathbf{X}(k), \end{aligned} \tag{10.23}$$

$$\mathbf{P}(n) = \lambda \sum_{k=0}^{n-1} \lambda^{n-k-1} d(k)\mathbf{X}(k) + d(n)\mathbf{X}(n), \tag{10.24}$$

$$\mathbf{P}(n) = \lambda\mathbf{P}(n-1) + d(n)\mathbf{X}(n), \tag{10.25}$$

where λ is the forgetting factor [1, 9]. In a similar way, we find that

$$\mathbf{R}(n) = \lambda\mathbf{R}(n-1) + \mathbf{X}(n)\mathbf{X}^T(n). \tag{10.26}$$

Then, multiplying (10.20) on the left by $\mathbf{R}^{-1}$, from (10.25) and (10.26) it follows that

$$\mathbf{W} = [\lambda\mathbf{R}(n-1) + \mathbf{X}(n)\mathbf{X}^T(n)]^{-1}[\lambda\mathbf{P}(n-1) + d(n)\mathbf{X}(n)]. \tag{10.27}$$

Next, using the matrix inversion lemma [1]

$$[\mathbf{A} + \mathbf{BCD}]^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}[\mathbf{DA}^{-1}\mathbf{B} + \mathbf{C}^{-1}]\mathbf{DA}^{-1} \tag{10.28}$$

with $\mathbf{A} = \lambda\mathbf{R}(n-1)$, $\mathbf{B} = \mathbf{X}(n)$, $\mathbf{C} = 1y\mathbf{D} = \mathbf{X}^T(n)$, we obtain

$$\mathbf{W} = \Big[\frac{1}{\lambda}\mathbf{R}^{-1}(n-1) - \Big[\frac{1}{\lambda}\mathbf{R}^{-1}(n-1)\mathbf{X}(n)\Big]* \\ * \Big[\frac{1}{\lambda}\mathbf{X}^T(n)\mathbf{R}^{-1}(n-1)\mathbf{X}(n) + 1\Big]^{-1}\frac{1}{\lambda}\mathbf{X}^T(n)\mathbf{R}^{-1}(n-1)\Big]* \\ * \Big[\lambda\mathbf{P}(n-1) + d(n)\mathbf{X}(n)\Big], \quad (10.29)$$

$$\mathbf{W} = \frac{1}{\lambda}\Big[\mathbf{R}^{-1}(n-1) - \frac{\mathbf{R}^{-1}(n-1)\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{R}^{-1}(n-1)}{\Big[\lambda + \mathbf{X}^T(n)\mathbf{R}^{-1}(n-1)\mathbf{X}(n)\Big]}\Big]* \\ * \Big[\lambda\mathbf{P}(n-1) + d(n)\mathbf{X}(n)\Big]. \quad (10.30)$$

Let us define

$$\mathbf{Q}(n) = \mathbf{R}^{-1}(n) \quad (10.31)$$

and

$$\mathbf{K}(n) = \frac{\mathbf{R}^{-1}(n-1)\mathbf{X}(n)}{\lambda + \mathbf{X}^T(n)\mathbf{R}^{-1}(n-1)\mathbf{X}(n)}. \quad (10.32)$$

Then, from (10.30) it follows that

$$\mathbf{W}(n) = \mathbf{Q}(n-1)\mathbf{P}(n-1) + \frac{1}{\lambda}d(n)\mathbf{Q}(n-1)\mathbf{X}(n) - \\ - \mathbf{K}(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{P}(n) - \\ - \frac{1}{\lambda}d(n)\mathbf{K}(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n), \quad (10.33)$$

$$\mathbf{W}(n) = \frac{1}{\lambda}[\mathbf{Q}(n-1) - \mathbf{K}(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)][\lambda\mathbf{P}(n-1) + d(n)\mathbf{X}(n)], \quad (10.34)$$

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \frac{1}{\lambda}d(n)\mathbf{Q}(n-1)\mathbf{X}(n) - \frac{\mathbf{Q}(n-1)\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{W}(n)}{\lambda + \mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)} - \\ - \frac{1}{\lambda}\frac{d(n)\mathbf{Q}(n-1)\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)}{\lambda + \mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)}, \quad (10.35)$$

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \\ + \frac{1}{\lambda}\frac{\mathbf{Q}(n-1)\mathbf{X}(n)}{[\lambda + \mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)]}\big[\lambda d(n) + d(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n) - \\ - \lambda\mathbf{X}^T(n)\mathbf{W}(n-1) - d(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)\big] \quad (10.36)$$

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \frac{1}{\lambda}\frac{\mathbf{Q}(n-1)\mathbf{X}(n)}{[\lambda + \mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)]}\lambda[d(n) - \mathbf{X}^T(n)\mathbf{W}(n-1)]. \quad (10.37)$$

Finally, we get

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \mathbf{K}(n)e(n), \quad (10.38)$$

where

$$\mathbf{K}(n) = \frac{\mathbf{Q}(n-1)\mathbf{X}(n)}{\lambda + \mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)} \quad (10.39)$$

An alternative expression can be obtained form (10.34) as follows [1, 5, 9]:

$$\mathbf{Q}(n) = \frac{1}{\lambda}[\mathbf{Q}(n-1) - \mathbf{K}(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)]. \tag{10.40}$$

After multiplying both sides of (10.40) by $\mathbf{X}(n)$ on the left, we make the following transformations:

$$\mathbf{Q}(n)\mathbf{X}(n) = \frac{1}{\lambda}\Big[\mathbf{Q}(n-1)\mathbf{X}(n) - \mathbf{K}(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)\Big], \tag{10.41}$$

$$\mathbf{Q}(n)\mathbf{X}(n) = \frac{1}{\lambda}\Big[\lambda+\mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)\Big]\mathbf{K}(n)-\mathbf{K}(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)\Big], \tag{10.42}$$

$$\mathbf{Q}(n)\mathbf{X}(n) = \mathbf{K}(n)+\frac{1}{\lambda}\mathbf{K}(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)-\frac{1}{\lambda}\mathbf{K}(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n), \tag{10.43}$$

$$\mathbf{Q}(n)\mathbf{X}(n) = \mathbf{K}(n). \tag{10.44}$$

Finally, substituting (10.44) in (10.38), we obtain

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \mathbf{Q}(n)e(n)\mathbf{X}(n), \tag{10.45}$$

where

$$\mathbf{Q}(n) = \frac{1}{\lambda}[\mathbf{Q}(n-1) - \frac{\mathbf{Q}(n-1)\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)}{\lambda + \mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)}]. \tag{10.46}$$

10.3.1. Convergence Performance with Stationary and non-Stationary Input Signals. To evaluate the convergence performance of RLS algorithm, an identifying time invariant is required as well as for time varying systems using several forgetting factors λ [9]. Figure 10.5 shows the convergence performance of a FIR adaptive filter whose coefficients are updated using the RLS adaptive algorithm with three different convergence factors, when it is required to identify a time invariant system. The input signal was a white noise sequence with a signal-to-noise ratio of 20 dB. As expected, in this situation, the convergence performance improves as the forgetting factor becomes close to one. On the other hand, Figs 10.6 and 10.7 show the MSE obtained when the RLS algorithm is required to identify a time varying system whose coefficients are given by

$$\begin{aligned} A[i,k] &= A[i,k]\cos(2\pi k/120) \qquad \text{k is odd,} \\ A[i,k] &= A[i,k]\cos(2\pi k/200) \qquad \text{k is even,} \end{aligned}$$

where A[i,j] are constant real numbers.

From this figures, it follows that when the unknown system is a time varying one, a forgetting factor close to one does not provide the best results because it does not allow one to track the system variations. Therefore, a smaller forgetting factor must be used as shown in Figs. 10.6 and 10.7.

10.4. Least-Mean-Square Algorithm

The least-mean-square (LMS) is the most widely used adaptive algorithm due, mainly, to its low computational complexity and robustness. It is based on a gradient approach given by [3]

$$\mathbf{W}(n) = \mathbf{W}(n-1) - \mu\nabla, \tag{10.47}$$

where $\mathbf{W}(n)$ is the coefficients vector,

$$\nabla = E\left[\frac{\partial e^2(n)}{\partial w_0}, \frac{\partial e^2(n)}{\partial w_1}, \ldots, \frac{\partial e^2(n)}{\partial w_{n-1}}\right]^T \tag{10.48}$$

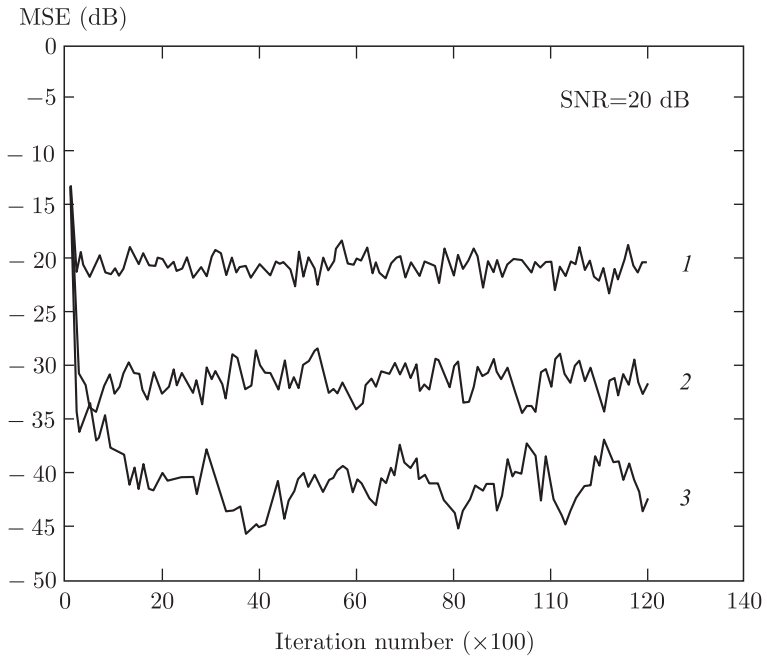


Fig. 10.5. Convergence performance of an RLS algorithm using three different forgetting factors (1) 0.9, (2) 0.99 and (3) 0.999, when it is required to identify a time invariant unknown system.

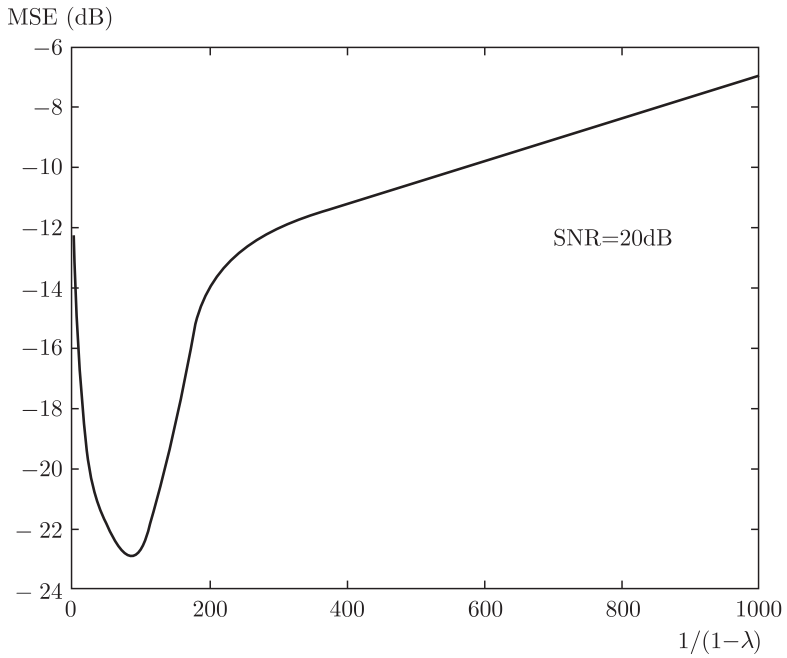


Fig. 10.6. MSE obtained when the RLS algorithm is required to identify a time varying system when different convergence factors. The input signal is a white noise signal.

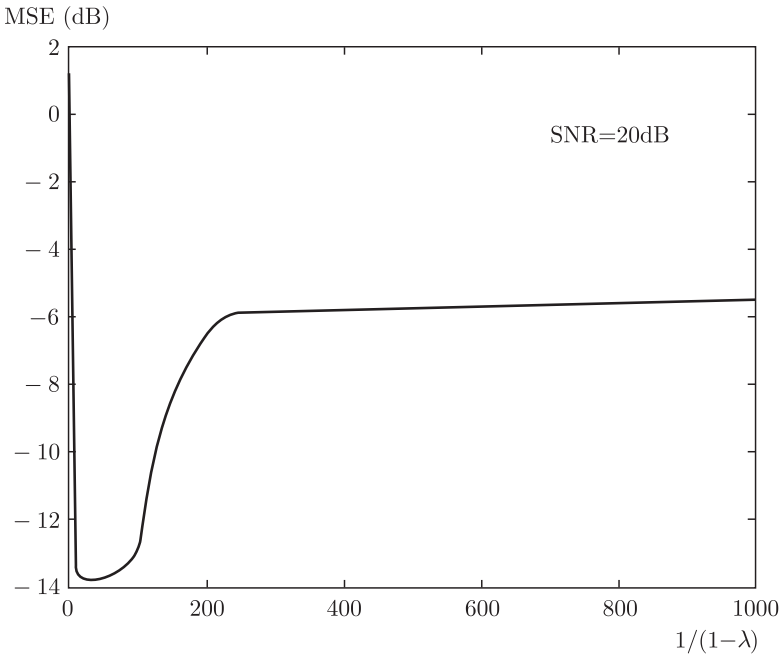


Fig. 10.7. MSE obtained when the RLS algorithm is required to identify a time varying system with different convergence factors. The input signal is an autoregressive process.

is the gradient of the mean square error surface, and μ is the convergence factor that controls the stability and convergence. From Fig. 10.1 it follows that

$$e(n) = d(n) - \mathbf{W}^T\mathbf{X}(n) \tag{10.49}$$

and therefore

$$\nabla = -2E\left[e(n)x(n), e(n)x(n-1), \ldots, e(n)x(n-N+1)\right]^T. \tag{10.50}$$

However, the gradient estimation is difficult because the MSE surface is unknown and it must be estimated from the input data. This process requires a considerable computational effort. To solve this problem Bernard Widrow [3] proposed to replace the true gradient by an instantaneous one, which is given by

$$\hat{\nabla} = 2\left[\frac{\partial e^2(n)}{\partial w_0}, \frac{\partial e^2(n)}{\partial w_1}, \ldots, \frac{\partial e^2(n)}{\partial w_{n-1}}\right]^T. \tag{10.51}$$

Substituting equations (10.12)–(10.15) into equation (10.51), we have

$$\hat{\nabla} = -2[e(n)x(n), e(n)x(n-1), \ldots, e(n)x(n-N+1)]^T, \tag{10.52}$$

$$\hat{\nabla} = -2e(n)\mathbf{X}(n). \tag{10.53}$$

Finally, substituting equation (10.53) into (10.47), we obtain

$$\mathbf{W}(n) = \mathbf{W}(n-1) + 2\mu e(n)\mathbf{X}(n). \tag{10.54}$$

Equation (10.54) is known as the least-mean-square (LMS) algorithm or Widrow–Hopf adaptive algorithm. This algorithm requires only 2N+1 multiplications and 2N+1 additions. This computational complexity, which is very low for most practical applications, does the LMS algorithm suitable for many practical applications.

10.4.1. Proof of Convergence. Consider the output error of FIR adaptive filter, which is given by

$$e(n) = d(n) - \mathbf{X}^T(n)\mathbf{W}(n-1), \tag{10.55}$$

where $\mathbf{X}(n)$ and $\mathbf{W}$ are the input and coefficients vectors given by (10.13) and (10.14). Substituting (10.55) into (10.54), after some straightforward manipulations, we obtain [3]

$$\mathbf{W}(n) = \mathbf{W}(n-1) + 2\mu d(n)\mathbf{X}(n) - 2\mu\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{W}(n-1). \tag{10.56}$$

Taking the expected value of (10.56), after some straightforward manipulation we get

$$E\left[\mathbf{W}(\mathrm{n})\right] = E\left[\mathbf{W}(\mathrm{n}-1)\right] + 2\mu E\left[d(n)\mathbf{X}(n)\right] - 2\mu E[\mathbf{X}(n)\mathbf{X}^T(n)]E\left[\mathbf{W}(\mathrm{n}-1)\right], \tag{10.57}$$

$$E\left[\mathbf{W}(\mathrm{n})\right] - \mathbf{R}^{-1}\mathbf{P} = E\left[\mathbf{W}(\mathrm{n}-1)\right] - \mathbf{R}^{-1}\mathbf{P} + 2\mu\mathbf{R}[\mathbf{R}^{-1}\mathbf{P} - E\left[\mathbf{W}(\mathrm{n}-1)\right]], \tag{10.58}$$

where

$$\mathbf{R} = \begin{bmatrix} \gamma_{xx}(0) & \gamma_{xx}(1) & \dots & \gamma_{xx}(L-1) \\ \gamma_{xx}(1) & \gamma_{xx}(0) & \dots & \gamma_{xx}(L-2) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{xx}(L-1) & \gamma_{xx}(L-2) & \dots & \gamma_{xx}(0) \end{bmatrix} = E[\mathbf{X}(n)\mathbf{X}^T(n)], \tag{10.59}$$

$$\mathbf{P} = E[d(n)\mathbf{X}(n)] = E[\mathbf{X}(n)d(n)] \tag{10.60}$$

are the input signal autocorrelation matrix and cross-correlation vector between the input and reference signals, respectively. Then, defining

$$\xi(n) = E\left[\mathbf{W}(\mathrm{n})\right] - \mathbf{R}^{-1}\mathbf{P}, \tag{10.61}$$

from (10.58) it follows that [3]

$$\xi(n) = \left(\mathbf{I} - 2\mu\mathbf{R}\right)\xi(n-1). \tag{10.62}$$

Next, from the fact that the autocorrelation matrix $\mathbf{R}$ can be decomposed using an orthogonal transformation $\mathbf{K}$. as follows

$$\mathbf{R} = \mathbf{K}^T\mathbf{Q}\mathbf{K}, \tag{10.63}$$

where

$$\mathbf{Q} = \mathrm{diag}\left[\lambda_1, \lambda_2, \dots \lambda_{N-1}\right] \tag{10.64}$$

and λ_i is the i-th eigenvalue of $\mathbf{R}$, alter some manipulations, it follows that

$$\xi(n) = \left(\mathbf{I} - 2\mu\mathbf{K}^{\mathrm{T}}\mathbf{Q}\mathrm{K}\right)\xi(n-1), \tag{10.65}$$

$$\xi(n) = \mathbf{K}^T\left(\mathbf{I} - 2\mu\mathbf{Q}\right)\mathbf{K}\xi(n-1). \tag{10.66}$$

Next, multiplying by $\mathbf{K}$ both sides and defining

$$\mathbf{V}(n) = \mathbf{K}\xi(n) \tag{10.67}$$

we find that [3, 9]

$$\mathbf{V}(n) = (\mathbf{I} - 2\mu\mathbf{Q})\,\mathbf{V}(n-1), \tag{10.68}$$

$$\begin{bmatrix} v_0(n) \\ v_1(n) \\ \vdots \\ v_{N-1}(n) \end{bmatrix} = \begin{bmatrix} (1-2\mu\lambda_1)^n & & & \\ & (1-2\mu\lambda_2)^n & & \\ & & \ddots & \\ & & & (1-2\mu\lambda_N)^n \end{bmatrix} \begin{bmatrix} v_0(0) \\ v_1(0) \\ \vdots \\ v_{N-1}(0) \end{bmatrix}, \tag{10.69}$$

Thus, using the LMS algorithm, we find that the expected value of $\mathbf{W}(n)$ will converge to the optimal solution of Wiener–Hopf equation if [3]

$$|1 - 2\mu\lambda_i| < 1, i = 1, 2, \ldots, N-1. \tag{10.70}$$

That is the LMS algorithm will converge if and only if

$$0 < \mu < \frac{1}{\lambda_{máx}}. \tag{10.71}$$

The eigenvalue estimation requires a considerable computational effort, so it is useful to find a practical boundary for the convergence factor μ. To this end, we can use the fact that

$$\sum_{i=0}^{N-1} x^2(n-i) = N\overline{x^2(n)} > \lambda_{\text{máx}}. \tag{10.72}$$

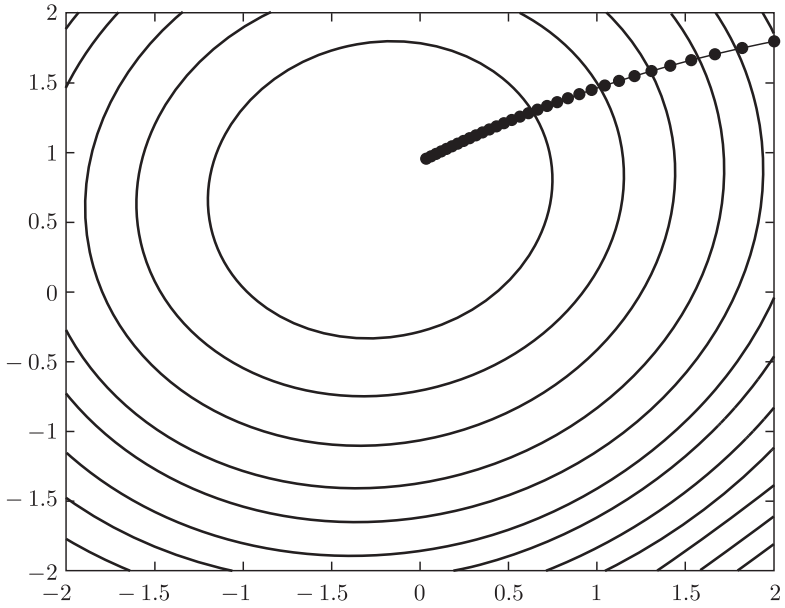


Fig. 10.8. Convergence characteristics of a second-order adaptive filter using a gradient-search based approach.

Then, the LMS Hill algorithm converges to the optimal solution if [3]

$$0 < \mu < \frac{1}{N\overline{x^2(n)}}, \tag{10.73}$$

where x^2 is the input signal power. Figure 10.8 shows the error surface and the gradient search trajectory for a second order adaptive filter with the convergence factor μ=0.03. This figure demonstrates that, if the convergence factor satisfies (10.73), the system converges to the optimal solution.

10.4.2. Properties of the Instantaneous Gradient. In the previous sections it was shown that the use of the instantaneous gradient instead of the real one provides the convergence of the mean value of the coefficients vector to the optimal solution of the Wiener–Hopf equation. Thus, in order to understand this behavior, it is useful to analyze the properties of the instantaneous gradient. To this end, let us consider (55), which can be written as [2, 9]

$$\mathbf{W}(n) = \mathbf{W}(n-1) - \mu(2\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{W}(n-1)) - 2d(n)\mathbf{X}(n)), \tag{10.74}$$

$$\mathbf{W}(n) = \mathbf{W}(n-1) - \mu\widehat{\nabla}, \tag{10.75}$$

where

$$\widehat{\nabla} = 2\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{W}(n-1) - 2d(n)\mathbf{X}(n). \tag{10.76}$$

Taking the expectation of (10.76), we obtain

$$E\left[\widehat{\nabla}\right] = 2E[\mathbf{X}(n)\mathbf{X}^T(n)]E\left[\mathbf{W}(\mathrm{n}-1)\right] - 2E\left[d(n)\mathbf{X}(n)\right], \tag{10.77}$$

$$E\left[\widehat{\nabla}\right] = 2\mathbf{R}E\left[\mathbf{W}(\mathrm{n}-1)\right] - 2\mathbf{P}. \tag{10.78}$$

Next, using the fact that the mean value of the coefficients vector converges to the optimal solution of Wiener–Hopf equation, that is $E[\mathbf{W}(n)] = \mathbf{W}^*$, and that $\nabla = 2\mathbf{R}\mathbf{W}^* - 2\mathbf{P}$, from (10.78) it follows that [3, 9]

$$E\left[\widehat{\nabla}\right] = \nabla. \tag{10.79}$$

Next, consider $\widehat{\nabla} \cdot \nabla$[8]:

$$\widehat{\nabla} \cdot \nabla = \widehat{\nabla}^T\nabla. \tag{10.80}$$

Taking the expectation of (10.80), we obtain

$$\widehat{\nabla} \cdot \nabla = E\left[\widehat{\nabla}^T\nabla\right], \tag{10.81}$$

$$\widehat{\nabla} \cdot \nabla = E\left[\widehat{\nabla}^T\right]\nabla. \tag{10.82}$$

Then, from (10.82), we find

$$\widehat{\nabla} \cdot \nabla = \nabla \cdot \nabla. \tag{10.83}$$

Equation (10.83) means that the projection of the instantaneous gradient on ∇ is equal to ∇. In the statistical sense it means that the instantaneous gradient always contains a component that is equal to the true gradient. However, the magnitude of the instantaneous gradient is larger than that of the tree one and its direction presents a deviation from the true one, as shown in Fig. 10.9. This deviation is denoted by the angle θ, which, from (10.79), satisfies the condition $E[\theta] = 0$.

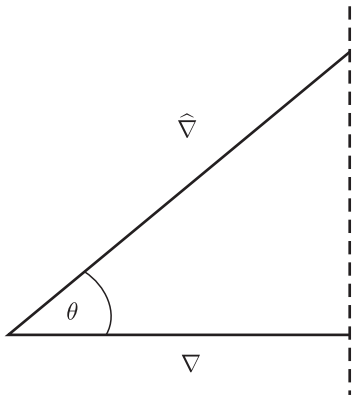


Fig. 10.9. Comparison between the instantaneous and true gradient vectors.

10.5. Normalized LMS Algorithm

A slightly different approach is the called normalized LMS algorithm. To analyze it, consider the LMS algorithm given by [1]

$$\mathbf{W}(n+1) = \mathbf{W}(n) + 2\mu e(n)\mathbf{X}(n). \tag{10.84}$$

Next, define

$$\mathbf{V}(n) = \mathbf{W}^* - \mathbf{W}(n), \tag{10.85}$$

where $\mathbf{W}^*$ and $\mathbf{W}(n)$ are the optimum and actual coefficients vector. Then, from (10.85) the identification error is given by [4]

$$e(n) = \mathbf{V}^T(n)\mathbf{X}(n). \tag{10.86}$$

Next, decomposing the error vector in its orthogonal and parallel components, we obtain

$$\mathbf{V}(n) = \mathbf{V}_0(n) + \mathbf{V}_p(n), \tag{10.87}$$

where $\mathbf{V}_o(n)$ and $\mathbf{V}_p(n) = C\mathbf{X}(n)$ are the component of $\mathbf{V}(n)$ orthogonal and parallel to the input vector $\mathbf{X}(n)$. After some straightforward manipulations, we find [3]

$$e(n) = [\mathbf{V}_0(\mathrm{n}) + \mathbf{V}_\mathrm{p}(\mathrm{n})]^T\,\mathbf{X}(n), \tag{10.88}$$

$$e(n) = [\mathbf{V}_o(n) + C\mathbf{X}(n)]^T\mathbf{X}(n), \tag{10.89}$$

$$e(n) = C\mathbf{X}^T(n)\mathbf{X}(n). \tag{10.90}$$

Thus, from (10.90) we get

$$C = \frac{e(n)}{\mathbf{X}^T(n)\mathbf{X}(n)}. \tag{10.91}$$

Then, from (10.91) the $\mathbf{V}(n)$ component parallel to the input vector $\mathbf{X}(n)$ is given by [4]

$$\mathbf{V}_p(n) = \frac{e(n)\mathbf{X}(n)}{\mathbf{X}^T(n)\mathbf{X}(n)}. \tag{10.92}$$

Next, to estimate the optimum coefficients vector, we can subtract from $\mathbf{V}(n)$ a component proportional the $\mathbf{V}_p(\mathrm{n})$ such that the magnitude of $\mathbf{V}(n)$ becomes smaller, that is

$$\mathbf{V}(n+1) = \mathbf{V}(n) - \alpha\mathbf{V}_p(n). \tag{10.93}$$

Thus, from (10.85) and (10.93) it follows that [4]

$$\mathbf{W}^* - \mathbf{W}(n+1) = \mathbf{W}^* - \mathbf{W}(n) - \alpha \frac{e^{(}n)\mathbf{X}(n)}{\mathbf{X}^T(n)\mathbf{X}(n)}, \tag{10.94}$$

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \alpha \frac{e(n)\mathbf{X}(n)}{\mathbf{X}^T(n)\mathbf{X}(n)}, \tag{10.95}$$

where, on order to reduce the magnitude of $\mathbf{V}(n)$ in each iteration, α must satisfy the condition [4]

$$0\langle\alpha\langle 2. \tag{10.96}$$

Thus, the NLMS algorithm is equivalent to the LMS algorithm if [4, 9]

$$2\mu = \frac{\alpha}{\mathbf{X}^T(n)\mathbf{X}(n)}. \tag{10.97}$$

10.5.1. Misadjustment and Convergence Rate of NLMS Algorithm. To analyze the behavior of the NLMS algorithm with different convergence factors, consider the adaptive filter output error, which is given by [9]

$$e(n) = \mathbf{X}^T(n)\left[\mathbf{H} - \mathbf{W}(\mathrm{n})\right] + r(n), \tag{10.98}$$

where $\mathbf{H}$ is the unknown system coefficients vector and r(n) is a noise signal not correlated with $\mathbf{X}(n)$, and $\mathbf{W}(n)$ is the adaptive filter coefficients vector given by [4]

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \frac{\alpha}{\mathbf{X}^T(n)\mathbf{X}(n)}\mathbf{X}(n)e(n). \tag{10.99}$$

Substituting (10.98) into (10.99), we obtain

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \frac{\alpha\mathbf{X}(n)}{\mathbf{X}^T(n)\mathbf{X}(n)}\left[\mathbf{X}^{\mathrm{T}}(\mathrm{n})[\mathbf{H} - \mathbf{W}(\mathrm{n})] + \mathrm{r}(\mathrm{n})\right], \tag{10.100}$$

$$\mathbf{W}(n+1) - \mathbf{H} = \mathbf{W}(n) - \mathbf{H} + \frac{\alpha\mathbf{X}(n)\mathbf{X}^T(n)}{\mathbf{X}^T(n)\mathbf{X}(n)}\left[\mathbf{H} - \mathbf{W}(\mathrm{n})\right] + \frac{\alpha r(n)\mathbf{X}(n)}{\mathbf{X}^T(n)\mathbf{X}(n)}. \tag{10.101}$$

Next, defining

$$\varepsilon(n) = \mathbf{X}^T(n)\left[\mathbf{W}(\mathrm{n}) - \mathbf{H}\right], \tag{10.102}$$

multiplying on the left by $\mathbf{X}^T(\mathrm{n})$, and assuming that [7]

$$\mathbf{W}(n+1) \approx \mathbf{W}(n). \tag{10.103}$$

from (10.101)–(10.103) we obtain [9]

$$\varepsilon(n) = \mathbf{X}^T(n)\left[[\mathbf{W}(n) - \mathbf{H}] + \frac{\alpha\mathbf{X}(n)\mathbf{X}^T(n)}{\mathbf{X}^T(n)\mathbf{X}(n)}[\mathbf{H} - \mathbf{W}(n)]\right] + \frac{\alpha r(n)\mathbf{X}^T(n)\mathbf{X}(n)}{\mathbf{X}^T(n)\mathbf{X}(n)}, \tag{10.104}$$

$$\varepsilon(n) = \left[\mathbf{X}^{\mathrm{T}}(\mathrm{n}) - \frac{\alpha\mathbf{X}^T(n)\mathbf{X}(n)\mathbf{X}^T(n)}{\mathbf{X}^T(n)\mathbf{X}(n)}\right][\mathbf{W}(n) - \mathbf{H}] + \alpha r(n), \tag{10.105}$$

$$\varepsilon(n) = (1-\alpha)\mathbf{X}^T(n)[\mathbf{W}(n) - \mathbf{H}] + \alpha r(n), \tag{10.106}$$

$$\varepsilon(n) = (1-\alpha)\varepsilon(n) + \alpha r(n), \tag{10.107}$$

Taking the mean square value of (10.107) and assuming that r(n) is uncorrelated with $\varepsilon(n)$, we obtain [8]

$$E\left[\varepsilon^2(n)\right] = (1-\alpha)^2 E\left[\varepsilon^2(n)\right] + \alpha^2 E\left[r^2(n)\right], \tag{10.108}$$

$$(1-1+2\alpha-\alpha^2)E\left[\varepsilon^2(n)\right] = \alpha^2 E\left[r^2(n)\right], \tag{10.109}$$

$$E\left[\varepsilon^2(n)\right] = \frac{\alpha}{2-\alpha} E\left[r^2(n)\right]. \tag{10.110}$$

Finally, from (10.110), we find for MSE [9]:

$$MSE_{dB} = 10\log_{10} E\left[r^2(n)\right] + 10\log_{10}\frac{\alpha}{2-\alpha}. \tag{10.111}$$

Equation (10.111) shows that, for α=1, the MSE converges to the noise level, for α <1, the MSE converges below the noise level, while, for 1< α < ysg thegystem converges above the noise level. The convergence performance of NLMS algorithm with several convergence factors is shown in Fig. 10.10.

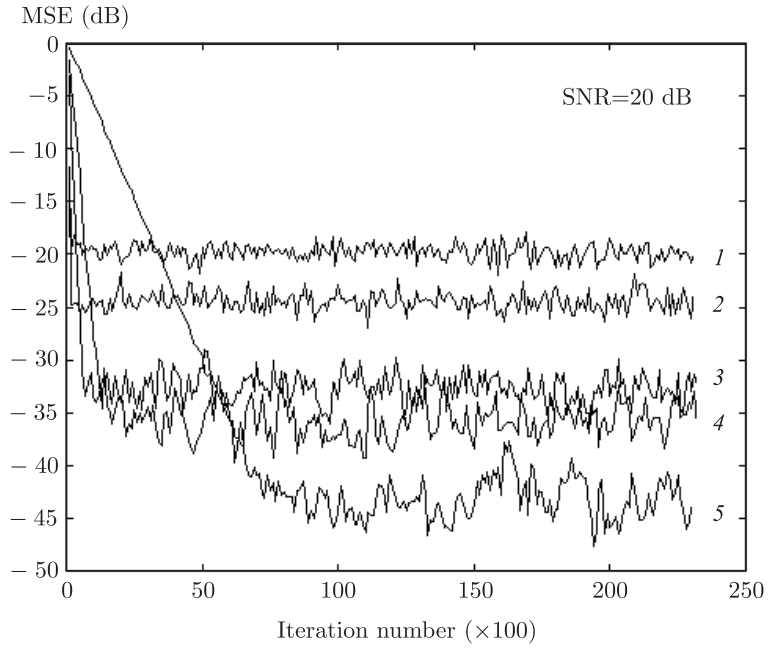


Fig. 10.10. Convergence performance of NLMS algorithm with convergence factors (1) α=1.0, (2) α=0.5, (3) α=0.1, (4) α=0.05, and (5) α=0.01. The input signal is a white noise sequence with zero mean and unit variance.

10.5.2. Time Constant. It is important to define the time constant of adaptive filter coefficients vector. Thus, consider the error vector given by [3]

$$\begin{bmatrix} v_0(n) \\ v_1(n) \\ \vdots \\ v_{N-1}(n) \end{bmatrix} = \begin{bmatrix} (1-\beta\lambda_1)^n & & & \\ & (1-\beta\lambda_2)^n & & \\ & & \ddots & \\ & & & (1-\beta\lambda_N)^n \end{bmatrix} \begin{bmatrix} v_0(0) \\ v_1(0) \\ \vdots \\ v_{N-1}(0) \end{bmatrix}, \tag{10.112}$$

where

$$\beta = \frac{\alpha}{\mathbf{X}^T(n)\mathbf{X}(n)}. \tag{10.113}$$

Next, using the first two terms of the series representation of exponential functions, we find [3], [14]

$$e^{(-\frac{n}{\tau_k})} = (1 - \frac{1}{\tau_k})^n. \tag{10.114}$$

Next, from [3]

$$(1 - \frac{1}{\tau_k})^n = (1 - \beta\lambda_k)^n \tag{10.115}$$

we obtain

$$\frac{1}{\tau_k} = \beta\lambda_k, \tag{10.116}$$

$$\tau_k = \frac{1}{\beta\lambda_k}. \tag{10.117}$$

Finally, substituting (10.113) into (10.117), we find that the time constant for the k-th mode is given by

$$\tau_k = \frac{\mathbf{X}^T(n)\mathbf{X}(n)}{\alpha\lambda_k}. \tag{10.118}$$

10.6. Time Varying Step Size LMS Algorithms

Equation (10.118) shows that, when the convergence factor decreases, that is α <1, the time constant increases, which results in a slower convergence rate, whereas, according to (10.111), in this situation the misadjustment decreases. Thus, with a constant convergence factor α it is not possible to achieve simultaneously small misadjustment and relatively fast convergence rates. In order to achieve small misadjustment and a reasonable fast convergence rate simultaneously, several time varying convergence factor LMS algorithms have been proposed [9–13]. Some of the most successful algorithms of this kind are described in the next sections.

10.6.1. TVSLMS Algorithm. From (10.111) it follows that, in order to keep a MSE constant, the factor α/(2-α) must decrease in the same scale that the additive noise power increases. To achieve this goal, the TVSLMS algorithm, which is suitable for echo cancellation applications, uses a time varying convergence factor given by [9, 13, 16]

$$\alpha(n) = \frac{\varepsilon\overline{x^2(n)}}{\varepsilon\overline{x^2(n)} + \overline{e^2(n)}}, \tag{10.119}$$

where e(n) is the output error and ε is a constant that satisfies the condition

$$\varepsilon\overline{x^2(n)} > \overline{e^2(n)} \tag{10.120}$$

when the noise power is low, that is equal or less than to 10^{-4}. It can be shown that, after convergence, the MSE becomes less than or equal to $\varepsilon\overline{x^2(n)}$. However, when the degradation is due to changes in the statistics of reference signal, the convergence factor becomes so small, resulting in a slow convergence rate, that it may limit its use in

several practical applications. To solve this problem, the convergence factor is modified in the following form [13, 16]:

$$\alpha(n) = \begin{cases} 1.0 & \text{if} \quad x^2(n) > d^2(n), \\ & \text{and} \quad d^2(n) < ke^2(n), \\ \dfrac{\varepsilon\overline{x^2(n)}}{\varepsilon\overline{x^2(n)} + \overline{e^2(n)}} & \text{otherwise,} \end{cases} \tag{10.121}$$

where

$$\overline{x^2(n)} = (1-\gamma)\overline{x^2(n-1)} + \gamma x^2(n), \tag{10.122}$$

$$\overline{e^2(n)} = (1-\gamma)\overline{e^2(n-1)} + \gamma e^2(n), \tag{10.123}$$

$$\overline{d^2(n)} = (1-\gamma)\overline{d^2(n-1)} + \gamma d^2(n), \tag{10.124}$$

and $1/\gamma$ is approximately equal to the number of data used for power estimation. The main idea behind this modification is the fact that, in many practical applications,

$$\overline{d^2(n)} < \overline{x^2(n)} \tag{10.125}$$

even if the noise power is moderately large. On the other hand, when the adaptation starts, when a significant change on the statistics of reference signal occurs, or when the additive noise power increases, the following condition is satisfied

$$\overline{d^2(n)} < k\overline{e^2(n)}. \tag{10.126}$$

However, both conditions are satisfied simultaneously only when a change on the statistics of input signal occurs and power additive noise is low. It is the only case in which a convergence factor equal to one is desirable. Thus, the convergence factor given by (10.121) satisfies the requirement of providing a small convergence factor when the noise power is large and a large convergence factor when the noise power is small, keeping at the same time an adequate performance when the degradation is due to changes on the statistics of reference signal, as shown in Fig. 10.11.

10.6.2. VSLMS Algorithm. One of the most widely used time varying step size LMS algorithms is the VSLMS in which the convergence factor is given by [9, 13, 15]

$$\alpha(n) = \begin{cases} 1 & \alpha'(n) \geqslant 1, \\ \alpha_{\min} & \alpha'(n) < \alpha_{\min}, \\ \alpha'(n) & \alpha_{\min} < \alpha'(n) < 1 \ldots 0, \end{cases} \tag{10.127}$$

where

$$\alpha'(n) = \gamma\alpha'(n-1) + (1-\gamma)e^2(n). \tag{10.128}$$

This algorithm presents several problems because the step size depends on the additive noise power and, when it is large and time varying, the performance of VSLMS algorithm degrades. However, its low computational complexity makes it an attractive alternative in many practical applications.

10.6.3. VECLMS Algorithm. A slightly different approach, which intend to reduce the algorithm dependency on the additive noise characteristics, is the VECLMS algorithm whose time varying step size is given by [13]

$$\alpha(n) = \begin{cases} 1 & \alpha'(n) \geqslant 1, \\ \alpha_{\min} & \alpha'(n) < \alpha_{\min}, \\ \alpha'(n) & \alpha_{\min} < \alpha'(n) < 1 \ldots 0, \end{cases} \tag{10.129}$$

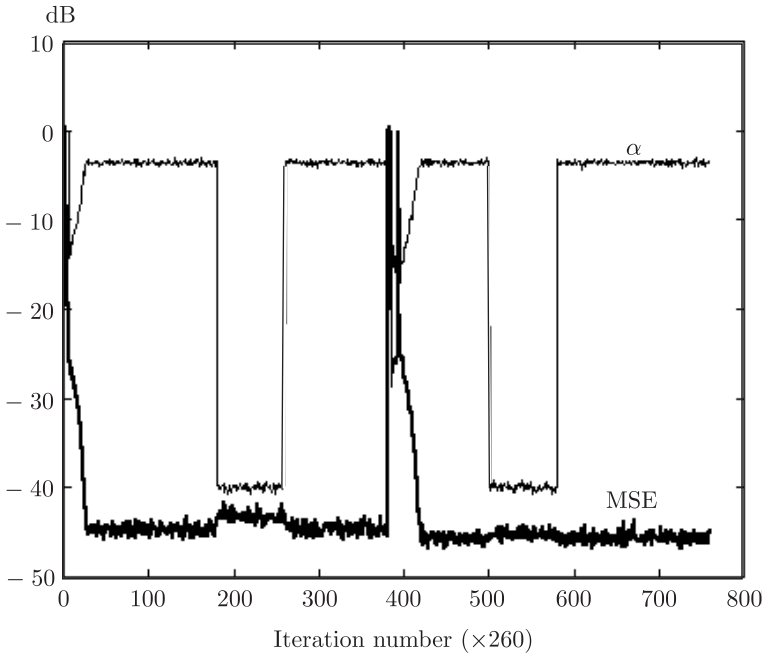


Fig. 10.11. Convergence performance of TVSLMS algorithm and the corresponding convergence factor when it is required to identify an unknown system of order 260. The input signal is a white noise with a time varying signal-to-noise ratio.

where

$$\alpha'(n) = \gamma\alpha'(n-1) + (1-\gamma)p^2(n), \tag{10.130}$$

and

$$p(n) = \gamma p(n-1) + (1-\gamma)e(n)e(n-1). \tag{10.131}$$

This algorthm presents similar characterstics as the VSLMS, except when the addive noise is white.

10.6.4. CC_LMS Algorithm. The CC_LMS algorithm intends to solve the problems still present in the VSLMS and VCLMS, using the cross-correation between the output error and the filter output sequences, respectively. The main idea behind the CC_LMS algorithm is the fact that, under the assumption that the additive noise and the input signals are mutually uncorrelated, we have [13, 17]

$$\overline{e(n)\widehat{y}(n)} = \overline{r(n)\widehat{y}(n)}. \tag{10.132}$$

Then, the step size of CC_LMS algorithm is independent of the additive noise power, providing in this form a better performance than the VSLMS and VECLMS algorithms. Thus, using the cross-correlation criterion, the CC_LMS step size is given by

$$\alpha(n) = \begin{cases} 1 & \text{if} \quad |\alpha'(n)| \geqslant 1, \\ \alpha_{\min} & \text{if} \quad |\alpha'(n)| < \alpha_{\min}, \\ |\alpha'(n)| & \text{if} \quad \alpha_{\min} < |\alpha'(n)| < 1 \ldots 0, \end{cases} \tag{10.133}$$

where

$$\alpha'(n) = \gamma\alpha'(n-1) + (1-\gamma)e(n)\widehat{y}(n). \tag{10.134}$$

10.6.5. ACFLMS Algorihm. Other way to overcome the limitations of the VSLMS algorithm is to use the ACFLMS algorithm, in which the convergence factor is given by [13]

$$\alpha(n) = 1.0 - \frac{1.0}{1.0 + \varepsilon \left|\alpha'(n)\right|}, \tag{10.135}$$

where α'(n) is given by (132). Here, when the cross-correlation between the output error and the filter output approach to one, the convergence factor becomes close to zero, keeping a low misadjustment. On the other hand, when the cross correlation approaches to zero, from (133) it follows that the convergence factor becomes close to one, providing a fast convergence rate.

10.6.6. NACFLMS Algorithm. The NACFLMS is a modification of ACFLMS algorithm in which the convergence factor is given by [13]

$$\alpha(n) = 1.0 - \frac{1.0}{1.0 + \varepsilon e_p(n)}, \tag{10.136}$$

where $e_p(n)$ is given by

$$e_p(n) = \frac{e^2(n)\,|\mathbf{X}(n)|^2_{\max}}{e^2(n)\Big|_{\max}\,|\mathbf{X}(n)|}, \tag{10.137}$$

$e(n)$ is the output error, and $\mathbf{X}(n)$ is the input vector. This algorithm has similar convergence properties toh those of the ACFLMS algorithm.

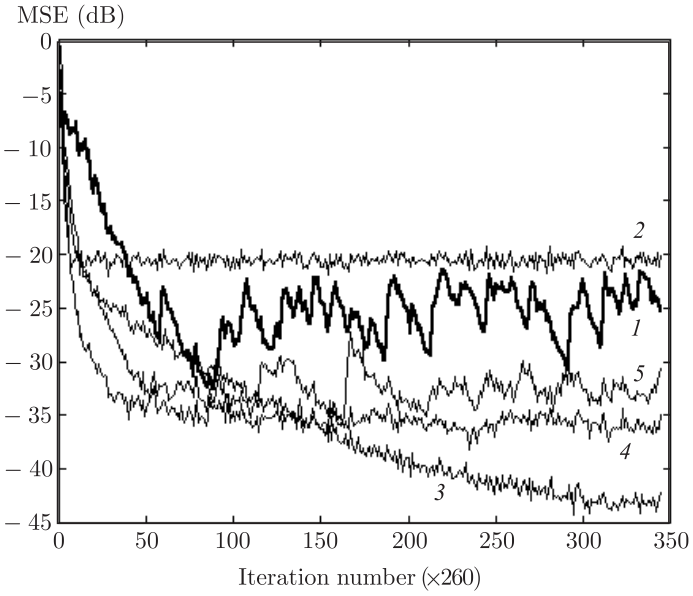


Fig. 10.12. Convergence performance of (1) TVSLMS, (2) NLMS [7], (3) VSLMS [10], (4) NACFLMS[12], and (5) VECLMS [13] when they are used to identify an unknown system of order 260. The input signal is white noise sequence with a SNR equal to 20 dB.

10.6.7. Simulation Results. Figures 10.12–10.16 show the convergence performance of the TVSLMS, NLMS [7], (3) VSLMS [10], (4) NACFLMS [12], and VECLMS [13] adaptive algorithms when they are used to identify an unknown system

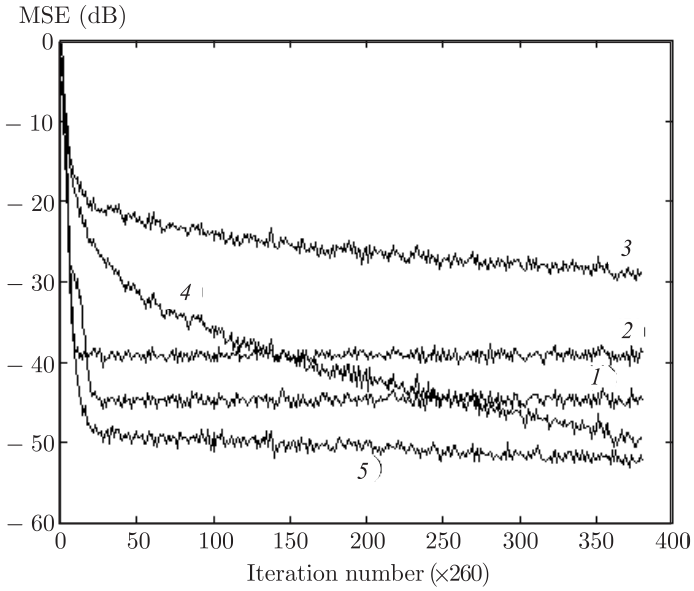


Fig. 10.13. Convergence performance of (1) TVSLMS, (2) NLMS [7], (3) VSLMS [10], (4) NACFLMS[12], and (5) VECLMS [13] when they are used to identify an unknown system of order 260. The input signal is white noise sequence with a SNR equal to 40 dB.

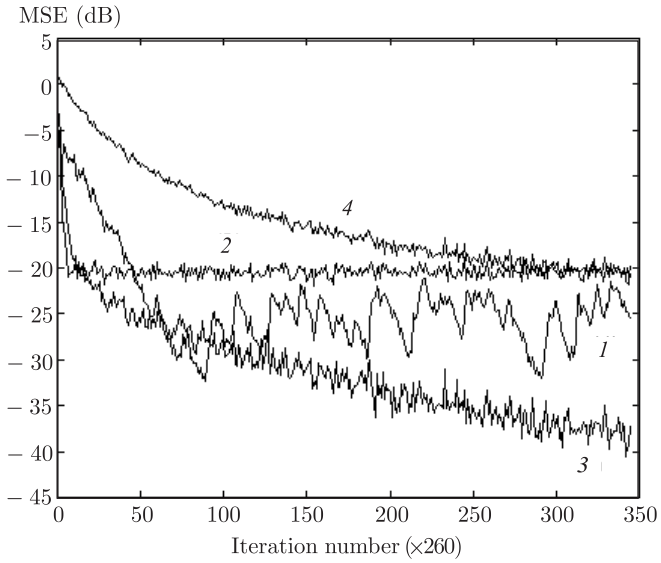


Fig. 10.14. Convergence performance of (1) TVSLMS, (2) NLMS [7], (3) VSLMS [10], (4) NACFLMS[12], and (5) VECLMS [13] when they are used to identify an unknown system of order 260. The input signal is an actual speech signal with a SNR equal to 20 dB.

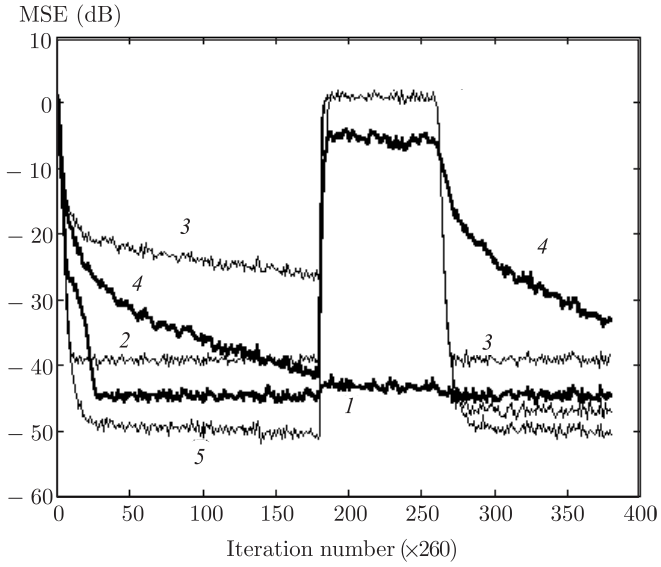


Fig. 10.15. Convergence performance of (1) TVSLMS, (2) NLMS [7], (3) VSLMS [10], (4) NACFLMS[12], and (5) VECLMS [13] algorithms when they are used to identify an unknown system of order 260. The input signal is white noise sequence. Here, firstly the SNR is equal to 40 dB; next, the SNR decays to 0 dB; and, finally, the SNR increases again to 40 dB.

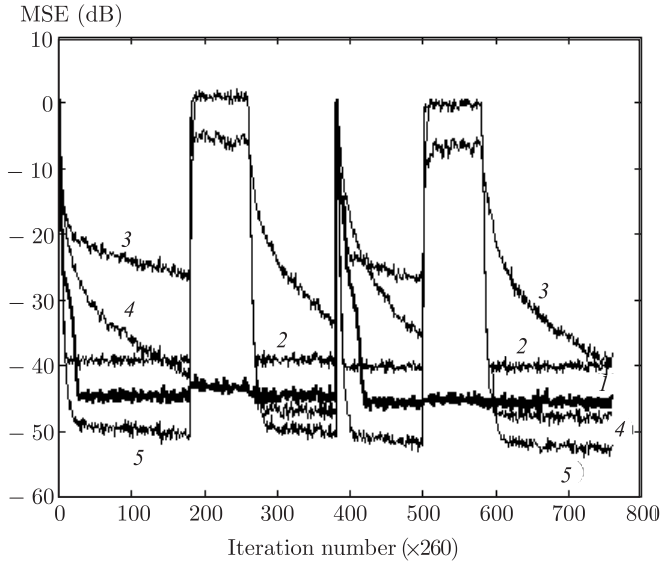


Fig. 10.16. Convergence performance of (1) TVSLMS, (2) NLMS [7], (3) VSLMS [10], (4) NACFLMS[12], and (5) VECLMS [13] algorithms when they are used to identify a time varying unknown system of order 260. The input signal is white noise sequence. Here, the SNR decays twice to 0 dB, increasing after some time interval to 40 dB.

of order 260 under several different conditions. In Figs. 10.12 and 10.13, the input signal is a white noise sequence with a signal-to-noise ratio equal to 20dB and 40dB, respectively.

Figure 10.14 shows the convergence performance of the abovementioned adaptive algorithms when the input signal is an actual speech signal with a signal-to-noise ratio equal to 20 dB. Figure 10.15 shows the convergence performance of time varying step size algorithms when they are used to identify and unknown system of order 260. Here, the input signal is a white noise sequence with a time varying SNR; firstly, equal to 40dB; next, during a time interval n decreases to 0dB; and, finally, increases again, becoming equal to 40 dB. Finally, Fig. 10.16 shows the convergence performance of above mentioned algorithms when they are used to identify a time varying system whose input signal is a with noise signal with time varying SNR.

Conclusions

The FIR adaptive filter algorithms are the most widely used due to their unconditional stability and unimodality of their MSE, which ensures the convergence to the global minimum. Two different approaches have been analyzed: the recursive least square (RLS) and the least mean square (LMS) approaches. The first one provides fast convergence rates and is robust to additive noise, although its computational complexity is very high for many practical applications. The second one, on the other hand, has a very low computational complexity, although its convergence rate is slow and it is highly sensitive to additive noise.

Several approaches that intend to provide a compromise between the RLS and LMS algorithms have been proposed and analyzed in this chapter. Evaluation results have demonstrated that, in general, the new algorithms provide better performance than the LMS with a slightly increase in their computational complexity.

References

1. *Cowan, C. and Grant, P.* Adaptive Filters, Prentice Hall, Englewood Cliff, NJ, 1985.
2. *Gritton, C. W. and Lin A. W.* Echo Cancellation Algorithms. IEEE ASSP Magazine, pp. 30–37, abril de 1984.
3. *Widrow, B. and Wallach, E.* On Statistical Efficiency of The LMS Algorithm with Not Stationary Input Signals. IEEE Trans. on Information Theory, vol. IT-30, pp. 211–221, marzo de 1984.
4. *Nagumo, J. and Noda, A.* A Learning Method for System Identification. IEEE Trans on Automatic Control, vol. AC-12, pp. 282–287, 1967.
5. *Lung, L. and Soderstrom, T.* Theory and Practice of Recursive Identification, MIT Press, Cambridge Ma, 1985.
6. *Perez Meana, H. and Tsujii, S.* A System Identification Using Orthogonal Functions. IEEE Transactions on Signal Processing, vol. 39, no. 3, pp. 752–7555, March 1991.
7. *Tsujii, S.* Handbook of Digital Signal Processing. The IEICE Press, 1992.
8. *Ferrara, E.R.* A Fast Implementation of LMS Adaptive Filters. IEEE Trans. on ASSP, vol. ASSP-28, pp. 474–475, August 1990.
9. *Nakano Miyatake, M.* Adaptive Systems for Filtering and Detection. Ph. D dissertation, Metropolitan University, Mexico City, Dec. 1998 (in Spanish).
10. *Chao, J., Pérez Meana, H., and Tsujii, S.* A Jumping Algorithm for Adaptive Signal Processing. Trans. of the IEICE, vol. J73-A, pp. 1196–1206, July de 1990.
11. *Chao J., Pérez Meana, H., and Tsujii, S.* A Fast Adaptive Filter Algorithm Using Eigenvalue Reciprocals as Step Size. IEEE Trans. on Signal Processing, vol. SP-38, pp. 1343–1352, October 1990.

12. *Asharif M. R. and Amano, F.* Frequency Bin Adaptive Filtering (FBAF) Algorithm and Its Applications to Acoustic Echo Cancelling. Trans. of The IEICE, vol. E74, no. 8, pp. 2276–2283, August 1991.
13. *Perez Meana, H. and Nakano Miyatake, M.*, Algoritmos LMS con factores de convergencia variables en el tiempo. Cientifica, vol. 8, no. 3, pp. 139–150, 2004.
14. *Honig, M. and Messerschmitt, D.* Adaptive Filters: Structures, Algorithms and Applications, Kluwer Academic Publishers, Boston Mass. USA, 1984.
15. *Evans, J. B.* A New Variable Step Size Method Suitable for VLSI Implementation, Proc. of The International Conference on Acoustic, Speech and Signal Processing, pp. 2105–2108, 1991.
16. *Nakano, M. and Perez, H.* A Time Varying Step Size Normalized LMS Algorithm for Adaptive Echo Canceler Structure. IEICE Trans. on Fundamentals of Electronics Computer Sciences, vol. E-78-A, pp. 254–258, 1995.
17. *Casco, F., Perez, H., Nakano, M., and Lopez, M.* A Variable Step Size (VSS-CC) NLMS Algorithm. IEICE Trans Fundamentals of Electronics Computer Sciences, vol. E78-A, pp. 1004–1009. 1995.

Chapter 11

FREQUENCY DOMAIN ADAPTIVE FILTERS BASED ON SUBBAND DECOMPOSITION

One important approach to reduce the computational complexity of adaptive filters is that based on subband decomposition in which the input signals are represented in terms of a set of N near orthogonal signal components using an orthogonal transformation. Such representation makes possible processing schemes in which a lower order adaptive filter is inserted in each subband whose coefficients vectors can be independently or jointly updated, allowing the use of efficient algorithms such as the Recursive Least Square (RLS) algorithm with reduced computational complexity, when fast convergence rates are required, as well as efficient versions of Fast Least Mean Square algorithms when large filter orders are needed. This chapter presents the development of such algorithms along with computer simulation showing their convergence performance.

11.1. Introduction

Adaptive filtering has been a subject of active research and considered as a desirable alternative to conventional transversal filters for several practical problems, such as echo and noise canceling, linear prediction, etc. [1]. During this time several efficient algorithms have appeared in literature, most of them using the direct realization form [2–6]. However, the direct realization form of fast convergence adaptive filter algorithms, such as the RLS algorithm, is in general too large for many practical applications [1]. On the other hand, in real time signal processing, a significant amount of computational effort can be saved if the input signals are represented in terms of a set of orthogonal signal components [6]. This is because the representation admits processing schemes in which each of these signal components can be independently processed.

A suitable adaptive filter algorithm, when large filter orders are required, is the fast LMS algorithm. However, these adaptive filter algorithms present slow convergence rates and introduce long processing delays. These facts limit the use of these algorithms to several practical applications. Thus, in order to use these algorithms in a large number of practical applications, the convergence rate must be increased and the processing time reduced as much as possible.

Taking these facts into account, this chapter presents parallel form FIR adaptive filter algorithms based on the subband decomposition approach in which the input signals are split into a set of approximately orthogonal signal components by using the discrete cosine transform. Subsequently, a bank of FIR filters is inserted in each subband, whose parameters are updated to minimize a common error. Computer simulation results show that the subband decomposition-based FIR adaptive filter structures improve the characteristics of conventional adaptive filter structures.

11.2. FIR Adaptive Filter Structure Based on Subband Decomposition Approach

Consider the output signal y(n) of an Nth-order transversal filter, given by

$$y(n) = \mathbf{X}_F^T(n)\mathbf{H}_F, \tag{11.1}$$

where

$$\mathbf{X}_F(n) = \left[\mathbf{X}^{\mathrm{T}}(\mathrm{n}), \mathbf{X}^{\mathrm{T}}(\mathrm{n-M}), \mathbf{X}^{\mathrm{T}}(\mathrm{n-2M}), \ldots, \right.$$
$$\left. \ldots, \mathbf{X}^T(n-(L-2)M), \mathbf{X}^T(n-(L-1)M)\right]^T, \tag{11.2}$$

$$\mathbf{X}(n-kM) = [x(n-kM), x(n-kM-1), x(n-kM-2), \ldots$$
$$, x(n-(k+1)M+2), x(n-(k+1)M+1)]^T, \tag{11.3}$$

is the input vector, and

$$\mathbf{H}_F = \left[\mathbf{H}_0^{\mathrm{T}}, \mathbf{H}_1^{\mathrm{T}}, \mathbf{H}_2^{\mathrm{T}}, \ldots, \mathbf{H}_{\mathrm{L-1}}^{\mathrm{T}}\right]^T, \tag{11.4}$$

$$\mathbf{H}_k = \left[h_{kM}, h_{kM+1}, h_{kM+2}, \ldots, h_{(k+1)M-1}\right]^T \tag{11.5}$$

is the adaptive filter coefficients vector. Substituting (11.2) and (11.4) into (11.1), we obtain that

$$y(n) = \sum_{k=0}^{L-1} \mathbf{X}^T(n-kL)\mathbf{H}_k. \tag{11.6}$$

Next, defining

$$\mathbf{H}_k = \mathbf{C}^T\mathbf{A}_k, \tag{11.7}$$

where $\mathbf{C}$ denotes an orthogonal transformation such as the DFT, DCT, etc., and substituting (11.7) into (11.6), we obtain

$$y(n) = \sum_{k=0}^{L-1} \left(\mathbf{C}\mathbf{X}(n-kM)\right)^T \mathbf{A}_k = \sum_{k=0}^{L-1} \mathbf{U}^T(n-kM)\mathbf{A}_k, \tag{11.8}$$

where $\mathbf{U}^T(n-kM) = (\mathbf{C}\mathbf{X}(n-kM))^T$ and

$$\mathbf{U}(n-kM) = [u_0(n-kM), u_1(n-kM), u_2(n-kM), \ldots,$$
$$\ldots, u_3(n-kM), \ldots, u_{M-1}(n-kM)]^T \tag{11.9}$$

$$\mathbf{A}_k = \left[a_{k,1}, a_{k,2,}, \ldots, a_{k,(M-1)}\right]^T. \tag{11.10}$$

Using (11.9) and (11.10), we can represent $y(n)$ as

$$y(n) = \sum_{k=0}^{L-1}\sum_{r=0}^{M-1} a_{k,r}u_r(n-kM). \tag{11.11}$$

Let $\mathbf{U}(n-kM)$ denote the discrete Fourier transform (DFT) of input signal. Then (11.11) defines the output signal of the short delay fast least mean square, SDFLMS, adaptive filter proposed in [10] and described in Section 11.3. This approach, which is a generalization of the conventional FLMS adaptive filter algorithm, reduces the processing time and increases the convergence rate of conventional FLMS, providing at the same time perfect reconstruction properties. This structure performs fairly well

using block processing with gradient search based algorithms. However, when LMS-Newton type algorithms are required to increase the convergence rate, the computational complexity can be very high, even if the coefficients of the input signal transformation be mutually uncorrelated.

To reduce the computational complexity of the adaptive filter structure when a RLS type adaptation algorithm is used, firstly interchange the summation order as follows:

$$y(n) = \sum_{r=0}^{M-1} \sum_{k=0}^{L-1} a_{k,r} u_r(n - kM) \tag{11.12}$$

and define

$$\mathbf{V}_r(n) = [u_r(n), u_r(n-M), u_r(n-2M), u_r(n-3M), \ldots, \\ \ldots, u_r(n-(L-2)M), u_r(n-(L-1)M)]^T, \tag{11.13}$$

$$\mathbf{G}_r = \left[a_{0,r}, a_{1,r}, a_{2,r}, \ldots, a_{(L-1),r}\right]^T, \tag{11.14}$$

so, that (11.12) becomes

$$y(n) = \sum_{r=0}^{M-1} \mathbf{G}_r^T \mathbf{V}_r(n). \tag{11.15}$$

Equation (11.15) denotes the output signal of the subband decomposition based filter structure proposed in [8] and [9], which also has perfect reconstruction properties without regard for the statistics of the input signal or the adaptive filter order. Figure 11.1 shows that the realization forms given by (11.11) and (11.15) are equivalent.

11.2.1. Adaptation algorithm. Consider the output error of the SBDADF structure, shown in Fig. 11.1, which is given by

$$e(n) = d(n) - \left(\sum_{r=0}^{M-1} \mathbf{G}_r^T \mathbf{V}_r(n) \right), \tag{11.16}$$

where $\mathbf{G}_r$ and $\mathbf{V}_r$ are given by (11.14) and (11.13), respectively.

The performance of the proposed ANC structure strongly depends on the choice of the orthogonal transformation, because in the development of adaptive algorithm it is assumed that the transformation components are fully uncorrelated. Several orthogonal transformations that approximately satisfy this requirement could be used, such as the discrete cosine transform (DCT), the discrete Fourier transform (DFT), the discrete sine transform (DST), the Walsh–Hadamard transform, etc. Among them, the DCT appears to be an attractive alternative because it is a real transformation and has better orthogonalizing properties than other orthogonal transformations. Besides, it can be estimated in a recursive form by using a filter bank whose rth output signal is given by [11, 12] (See appendix)

$$u_r(n) = 2\cos\left(\frac{\pi r}{M}\right) u_r(n-1) - u_r(n-2) - \\ - \cos\left(\frac{\pi r}{2M}\right) \{x(n-M-1) - (-1)^r x(n-1) - \\ - x(n-M) + (-1)^r x(n)\}. \tag{11.17}$$

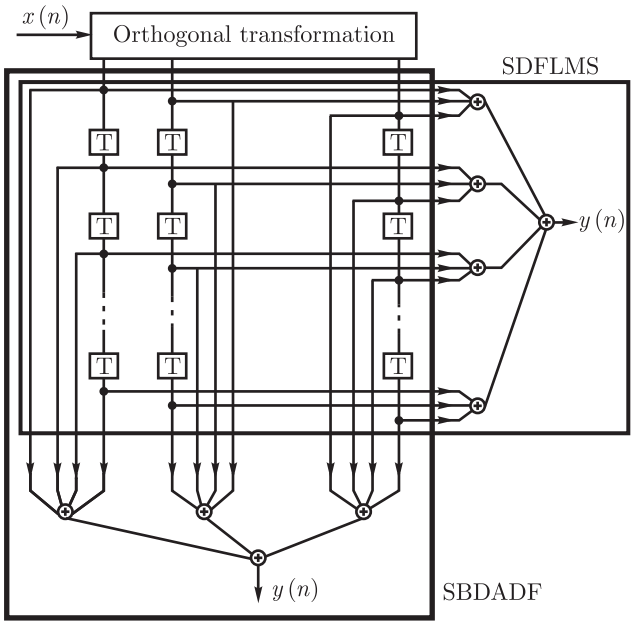


Fig. 11.1. Equivalence between the realization form of SDFLMS [15] and the subband decomposition based ADF [11].

To achieve high convergence rates, the coefficients vector, $\mathbf{G}_r$, $r = 0, 1, 2, \ldots, M-1$, will be estimated so that the sum of squared errors, $\varepsilon(n)$, given by

$$\varepsilon(n) = \sum_{k=1}^{n} \left(d(k) - \sum_{r=0}^{M-1} \mathbf{G}_r^T \widehat{\mathbf{V}}_r(k) \right)^2, \tag{11.18}$$

$$\varepsilon(n) = \sum_{k=1}^{n} \left(d(k) - \mathbf{G}^T \widehat{\mathbf{V}}(k) \right)^2 \tag{11.19}$$

attains a minimum, where

$$\mathbf{G} = \left[\mathbf{G}_1^{\mathrm{T}}, \mathbf{G}_2^{\mathrm{T}}, \mathbf{G}_3^{\mathrm{T}}, \ldots, \mathbf{G}_{\mathrm{L}-1}^{\mathrm{T}} \right]^T, \tag{11.20}$$

$$\widehat{\mathbf{V}}(n) = \left[\mathbf{V}_0^T(n), \mathbf{V}_1^T(n), \mathbf{V}_2^T(n), \ldots, \mathbf{V}_{M-1}^T \right]^T, \tag{11.21}$$

and $\mathbf{V}_r$ and $\mathbf{G}_r$(n) are given by (11.13) and (11.14), respectively.

Multiplying (11.21) on the left by $\mathbf{V}(n)$ and using the orthogonality property of the least square estimation, we obtain

$$\left[\sum_{k=1}^{n} \mathbf{V}(k) \mathbf{V}^T(k) \right] \mathbf{G}(n) = \sum_{k=1}^{n} d(k) \mathbf{V}(k). \tag{11.22}$$

Next, assuming that the DCT coefficients of the input signal are mutually uncorrelated [10–12], we can write equation (11.22) as follows:

$$\left[\sum_{k=1}^{n} \mathbf{V}_r(k) \mathbf{V}_r^T(k) \right] \mathbf{G}_r(n) = \sum_{k=1}^{n} d(k) \mathbf{V}_r(k), \tag{11.23}$$

$$\mathbf{G}_r(n) = \left[\sum_{k=1}^{n} \mathbf{V}_r(k)\mathbf{V}_r^T(k)\right]^{-1} \sum_{k=1}^{n} d(k)\mathbf{V}_r(k), \tag{11.24}$$

where $r = 0, 1, 2, 3, \ldots, M-1$. Equation (11.24) is the solution of the Wiener–Hopf equation, which can be solved by a recursive method based on the Matrix Inversion Lemma as follows:

$$\mathbf{G}_r(n+1) = \mathbf{G}_r(n) + \mu\,\mathbf{K}_r(n)e(n), \tag{11.25}$$

where $e(n)$ is the output error given by equation (11.16), μ is the convergence factor that controls the stability and convergence rate [1],

$$\mathbf{K}_r(n) = \frac{\mathbf{P}_r(n)\mathbf{V}_r(n)}{\lambda + \mathbf{V}_r^T(n)\mathbf{P}_r(n)\mathbf{V}_r(n)}, \tag{11.26}$$

$$\mathbf{P}_r(n+1) = \frac{1}{\lambda}\left[\mathbf{P}_{\mathrm{r}}(\mathrm{n}) - \mathbf{K}_{\mathrm{r}}(\mathrm{n})\mathbf{V}_{\mathrm{r}}^{\mathrm{T}}(\mathrm{n})\mathbf{P}_{\mathrm{r}}(\mathrm{n})\right], \tag{11.27}$$

and $\mathbf{V}_r(\mathrm{n})$ $\widehat{\mathbf{V}}_r(n)$ is given by equation (11.13). Taking into account that [1]

$$\mathbf{K}_r(n) = \mathbf{P}_r(n)\mathbf{V}_r(n), \tag{11.28}$$

we can represent equation (11.25) as follows:

$$\mathbf{G}_r(n) = \mathbf{G}_r(n-1) + \mu\,\mathbf{P}_r(n+1)e(n)\mathbf{V}_r(n). \tag{11.29}$$

Equation (11.29), when $\mu < 1$, defines the so-called LMS-Newton algorithm, which converges to the optimal solution when $0 < \mu < 1$. Figures 11.2 and 11.3 show the subband decomposition adaptive filter structure when it is updated using the modified RLS or LMS-Newton algorithm.

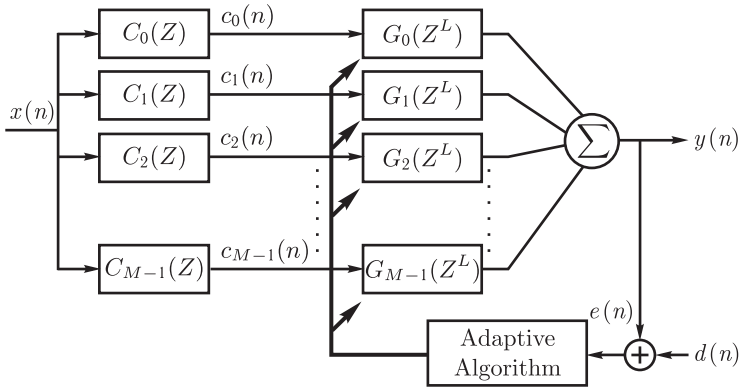


Fig. 11.2. Parallel form FIR filter structure using generalized subband decomposition.

11.2.2. Evaluation Results. To evaluate the actual convergence performance of parallel realization form FIR ADF, we used a system identification configuration. The FIR ADF system is required to identify an unknown system of order 20, using 4 subbands with a sparsity factor L equal to 5. Figures 11.4 and 11.5 show the convergence performance of the parallel form FIR ADF algorithm with a block diagonal autocorrelation matrix with a constant convergence factor and time varying convergence factor, respectively, when the input signal is an autoregressive process of 15th order. The signal-to-noise ratio is equal to 45 dB. The convergence performance obtained by using a conventional RLS algorithm with a full matrix autocorrelation matrix [1] is also

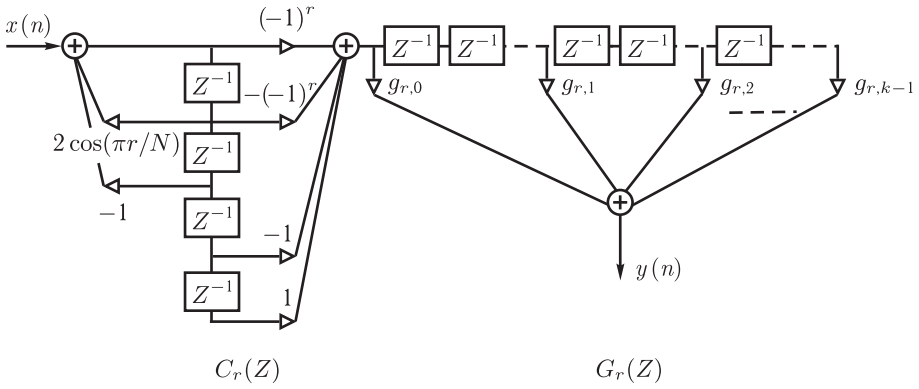


Fig. 11.3. $r-$th stage of proposed structure using the DCT as orthogonal transformation.

shown for comparison. Figures 11.6 and 11.7 show the convergence performance of the parallel FIR ADF algorithm with a constant and time varying convergence factor when the input signal was an actual speech signal with a signal-to-noise ratio of 25 dB. The convergence performance obtained by using a conventional RLS algorithm [1] is also shown for comparison in Fig. 11.7.

Computer simulations show that the parallel form FIR ADF algorithm provides quite similar convergence performance with a much less computations cost than the conventional RLS algorithm, although with a larger misadjustment when a fixed μ is used. The reason is that the DCT does not fully decorrelate the input signal. However the misadjustment decreases when a time varying $\mu(n)$ is used as shown in Fig. 11.7.

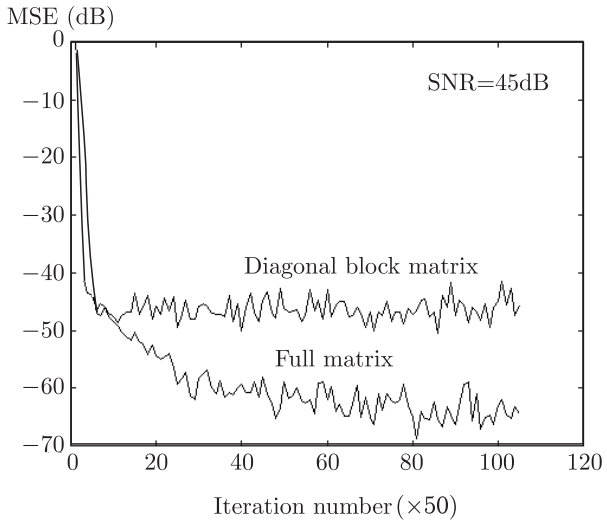


Fig. 11.4. Convergence performance of adaptive algorithm using block diagonal autocorrelation matrix with constant step size, compared with the convergence performance of conventional RLS algorithm with full input signal autocorrelation matrix.

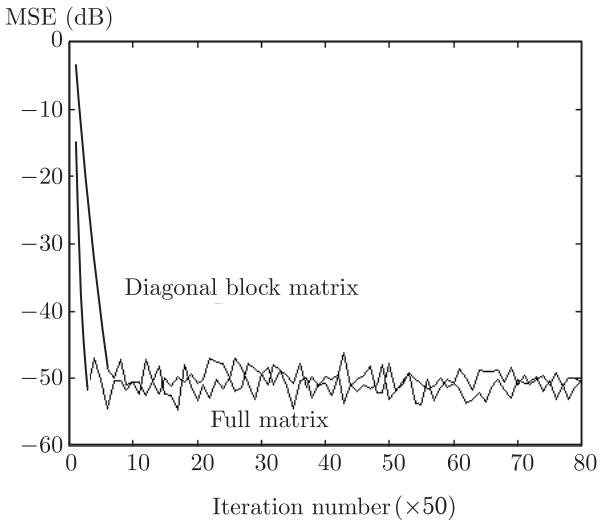


Fig. 11.5. Convergence performance of adaptive algorithm using block diagonal autocorrelation matrix with time varying step size, compared with the convergence performance of conventional RLS algorithm with full input signal autocorrelation matrix.

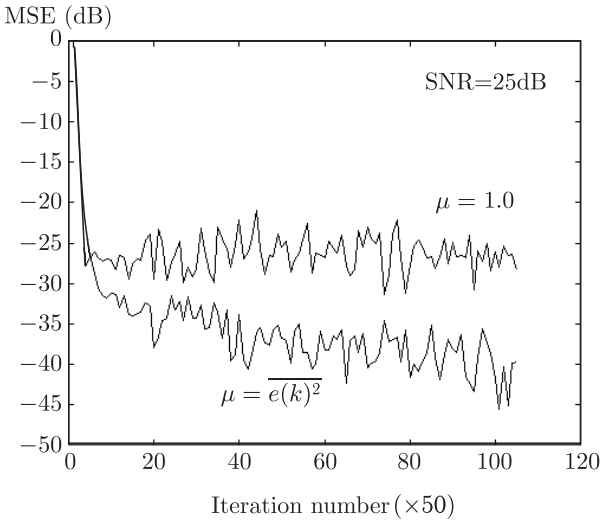


Fig. 11.6. Convergence performance of adaptive algorithm using block diagonal autocorrelation matrix with time varying and constant step sizes.

11.3. Short Delay Fast LMS Algorithm

Two of the most important problems limiting the use of frequency domain adaptive filters in several practical applications are a low convergence rate and long processing delay. Many efforts have been carried out to reduce the processing delay [13]–[18], to increase the convergence rate [20], or both of them [10]. This section presents a review

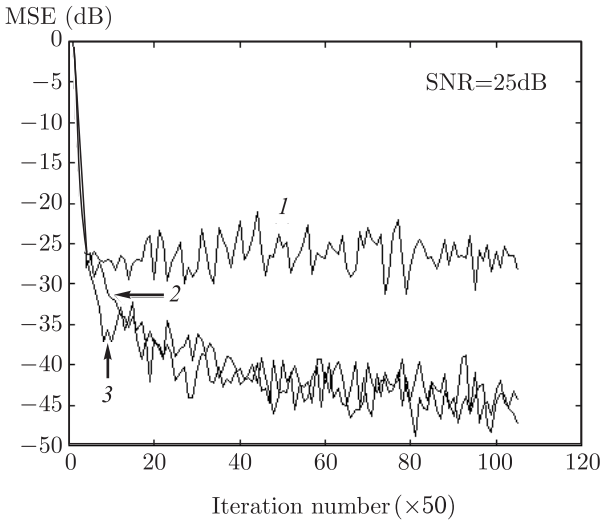


Fig. 11.7. Convergence performance of the FIR adaptive algorithm using a block diagonal autocorrelation matrix and a constant convergence factor equal to 1.0 (1), and a time varying convergence factor (2) respectively. The performance of conventional RLS algorithm is also shown for comparison. The input signal was a white noise sequence.

of the short delay fast LMS algorithm [10] that simultaneously increase the convergence rate and reduce the processing delay.

11.3.1. Short Delay Fast LMS Algorithm. Consider the output signal of a finite impulse response, given by

$$\widehat{y}(kL+j) = \mathbf{W}^T(k)\mathbf{X}(k), \tag{11.30}$$

where k is the block number, *j=0,1,... ,L-1*,

$$\mathbf{W}(k) = [\mathbf{W}_0(k), \mathbf{W}_1(k), \ldots, \mathbf{W}_{M-1}(k)]^T, \tag{11.31}$$

or

$$\mathbf{W}_m(k) = [w_{mL}(k), w_{mL+1}(k), \ldots, w_{mL+L-1}(k)]^T, \tag{11.32}$$

is the vector of coeficientes of the adaptable filter, and

$$\mathbf{X}(k) = [\mathbf{X}_0(k), \mathbf{X}_1(k), \ldots, \mathbf{X}_{M-1}(k)]^T, \tag{11.33}$$

or

$$\mathbf{X}_m(k) = [x((k-m)L+j), x((k-m)L+j-1), \ldots, \\ , \ldots, x((k-m)L+j-L+1)]^T, \tag{11.34}$$

is the input vector. Assuming that the coefficients vector $\mathbf{W}(k)$ remains constant, at least during L sampling periods, the output filter y(kL+j) can be obtained using the fast convolution methods with a 50% overlap. Thus, using the linearity property of the Fourier transform, after some manipulations, from (11.30)–(11.34) we obtain [10], [15]–[17]

$$\widehat{y}(kL+j) = \text{Last L terms of } FFT^{-1}\left[\sum_{m=0}^{M-1} \mathbf{B}_m(k)\mathbf{C}_m(k)\right], \tag{11.35}$$

where FFT^{-1} denotes the inverse Fourier transform,

$$\mathbf{C}_m(k) = FFT\left[\mathbf{W}_{\mathrm{m}}(\mathrm{k}), 0, 0, \ldots, 0, 0\right], \tag{11.36}$$

$$\mathbf{B}_m(k) = \mathrm{diag}\left\{FFT\left[\mathbf{X}_{\mathrm{m-1}}(\mathrm{k}), \mathbf{X}_{\mathrm{m}}(\mathrm{k})\right]\right\}. \tag{11.37}$$

Equations (11.35)–(11.37) produce the SDFLMS structure shown in Figs. 11.1 and 11.8.

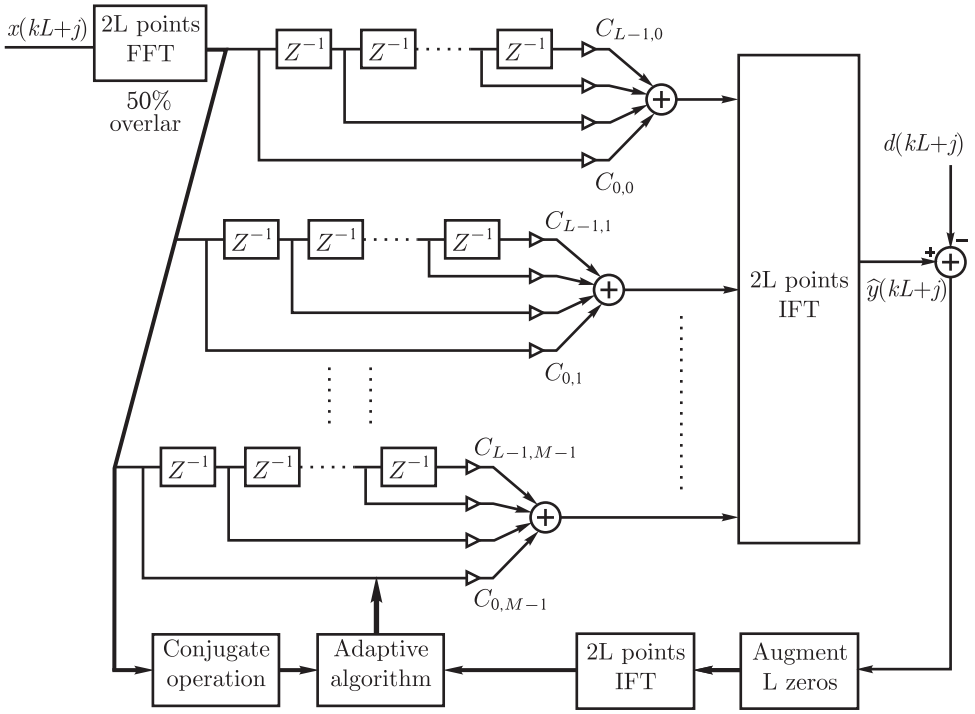


Fig. 11.8. Realization form of short delay FLMS (SDFLMS) adaptive filter.

11.3.2. Adaptation Algorithm. When a block LMS adaptive algorithm is used, the FIR filter coefficients vector in the kth block is given by

$$\mathbf{W}_m(k+1) = \mathbf{W}_m(k) + \beta\nabla_m(k), \tag{11.38}$$

where

$$\nabla_m(k) = \frac{1}{L}\sum_{j=0}^{L-1} e(kL+j)\mathbf{X}_m(kL+j) \tag{11.39}$$

is the estimated gradient in the kth block, e(kL+j) is the output error, and X_m(kL+j) is the input vector. Equation (11.39) defines the cross-correlation between the input vector and the output error, which can be estimated using the overlap-save method with a 50% overlap as follows:

$$\nabla_m(k) = \text{First L terms of } FFT^{-1}[\mathbf{B}_m^*(k)\mathbf{E}(k)], \tag{11.40}$$

where

$$\mathbf{E}(k) = FFT[0, 0, 0, \ldots, 0, e(kL), e(kL+1), e(kL+2), \ldots, e(kN+N-1)]^T, \tag{11.41}$$

and $\mathbf{B}_m^*(k)$ denotes the complex conjugated of $\mathbf{B}_m(k)$ given by (11.37). The frequency domain coefficients vector $\mathbf{C}_m(k)$ can also be adapted directly in the frequency domain. Thus, $\mathbf{C}_m(k+1)$, the frequency transform of $\mathbf{W}_m(k+1)$ is given by

$$\mathbf{C}_m(k+1) = \mathbf{C}_m(k) + \beta\ \mathbf{G}_m(k), m = 0, 1, 2, \ldots, M-1, \tag{11.42}$$

where

$$\mathbf{G}_m(k) = [FFT[\nabla_m(k), 0{,}0 \ldots, 0]]^T. \tag{11.43}$$

11.3.3. Convergence Condition. To derive a convergence factor that takes into account the spectral characteristics of input signals, substitute (11.39) into (11.38) and assume that the input signal is stationary in a wide sense. Thus, after some manipulations we obtain

$$\mathbf{W}(k) = \mathbf{W}(k-1) + \beta L\left[\mathbf{P} - \mathbf{R}\mathbf{W}(\mathrm{k}-1)\right], \tag{11.44}$$

where

$$\mathbf{P} = \frac{1}{L}\sum_{j=0}^{L-1} d(kL+j)\mathbf{X}^T(kL+j) \tag{11.45}$$

is the cross-correlation between the reference and input vector and

$$\mathbf{R} = \frac{1}{L}\sum_{j=0}^{L-1} \mathbf{X}(kL+j)\mathbf{X}^T(kL+j) \tag{11.46}$$

is the input vector autocorrelation matrix. Taking the expectation of (11.46) and subtracting in both sides the optimal solution, we obtain

$$\mathbf{V}(k) = (\mathbf{I} - \beta L\mathbf{Q})^k\mathbf{V}(0), \tag{11.47}$$

where

$$\mathbf{V}(k) = \mathbf{T}(\mathbf{W}(k) - \mathbf{W}_{op}), \tag{11.48}$$

$$\mathbf{Q} = \mathrm{diag}[\lambda_1, \lambda_2, \ldots, \lambda_N], \tag{11.49}$$

$\mathbf{T}$ is an orthogonal transformation, and λ_r is the r-th eigenvalue of the input vector autocorrelation matrix $\mathbf{R}$. From (11.47) it follows that

$$v_r(k) = (1 - N\beta\lambda_r)^k v_r(0). \tag{11.50}$$

Then, $V(k)$ will converge exponentially to zero and $W(k)$ will convergence to the optimal solution if

$$0 < \beta < \frac{2}{L\lambda_{\max}}. \tag{11.51}$$

Equation (11.51) determines the maximum value β allowing the filter convergence. However, of larger interest is to determine the convergence factor providing high convergence rates when gradient search based adaptive algorithms are used. The optimal factor can be obtained when the fastest and slowest mode converge at the same speed. That is, if [18]

$$|1 - L\beta\lambda_{\max}| = |1 - L\beta\lambda_{\min}|, \tag{11.52}$$

then

$$\beta_f = \frac{2}{L(\lambda_{\max} + \lambda_{\min})}. \tag{11.53}$$

To estimate the eigenvalues of the input vector autocorrelation matrix is a difficult task, which requires considerable computational effort because the orthogonal matrix required to this end is signal dependent and, in most cases, there is no fast algorithm to

compute it. In order to reduce the computational cost required for eigenvalue estimation, the DFT can be used. Although the eigenvalues obtained using the DFT are only approximations of the real ones, they are sufficiently good for the solution of many practical problems. Then, assuming that the input signal is stationary in a wide sense, the eigenvalues can be estimated as follows:

$$L\widehat{\mathbf{Q}} = \frac{L}{k}\sum_{i=1}^{k}|\mathbf{CX}(i)|^2, \tag{11.54}$$

where CX(i) denotes the Fourier transform of the input signal, *x(kL+j)*, and $|a|$ is the absolute value of a. Thus, from (11.54) it follows that the maximum and minimum eigenvalues can be obtained by estimating the maximum and minimum values of the input signal power spectral densities. In order to estimate the power spectral density without a significant increase in the computational complexity, the averaged modified periodograms with a 50% overlap can be used, which are estimated as follows

$$L\widehat{\mathbf{Q}}(k) = (1-\gamma)\widehat{\mathbf{Q}}(k-1) + \gamma\left[\sum_{m=0}^{M-1}\mathbf{B}_m(i)\mathbf{B}_m^*(i)\right], \tag{11.55}$$

where γ^{-1} is approximately equal to the number of blocks used for spectral density estimation. Finally, defining

$$L\widehat{\mathbf{Q}}(k) = (\zeta_1, \zeta_2, \ldots, \zeta_N), \tag{11.56}$$

from (11.53), we obtain

$$\beta_f = \frac{2}{\max(\zeta_1, \zeta_2, \ldots, \zeta_N) + \min(\zeta_1, \zeta_2, \ldots, \zeta_N)}, \tag{11.57}$$

where $\max(\zeta_1, \zeta_2, \ldots, \zeta_N)$ is the maximum value of $(\zeta_1, \zeta_2, \ldots, \zeta_N)$, which will be the estimate for $L\lambda_{\max}$, and $\min(\zeta_1, \zeta_2, \ldots, \zeta_N)$ is the minimum value of $(\zeta_1, \zeta_2, \ldots, \zeta_N)$, which will be the estimate for $L\lambda_{\min}$.

11.3.4. Evaluation Results. The simulation results, used to evaluate the convergence performance of SDFLMS algorithm, were obtained using a system identification configuration, although the SDFLMS can be used in any other configuration discussed in the accompanying chapters. The criterion used to evaluate the convergence characteristics was the normalized mean-square error given by

$$MSE = 10\log_{10}\frac{E[(y(kL+j) - \widehat{y}(kL+j))^2]}{E[y^2(kL+j)]}, \tag{11.58}$$

where $y(kL+j)$ denotes the unknown system output signal, which is the reference signal in the absence of noise and $y(kL+j)$ is the adaptive filter output.

Figure 11.9 shows the convergence performance of the SDFLMS algorithm with four different block delays (N/M) when it is required to identify an unknown system of order 128. The signal-to-noise ratio between the unknown system output and the additive noise is equal to 35 dB. Figure 11.10 shows the convergence performance of the SDFLMS when it is required to identify an unknown system of order 128. The input signal was an actual speech signal with a signal-to-noise ratio equal to 40 dBs. The convergence performance of the FLMS is also shown for comparison. Figure 11.11 shows the convergence performance of the SDFLMS when it is required to identify an unknown system of order 128. The input signal was a white nose sequence with a

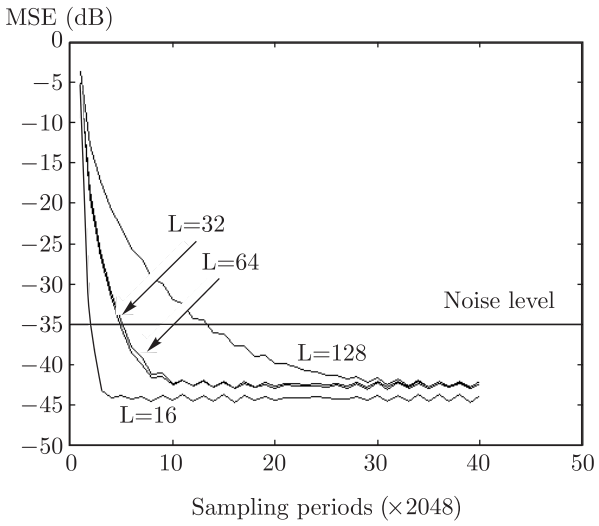


Fig. 11.9. Convergence performance of SDFLMS using 4 different processing delays. The input signal is a white noise sequence with a signal-to-noise ratio equal to 35 dB.

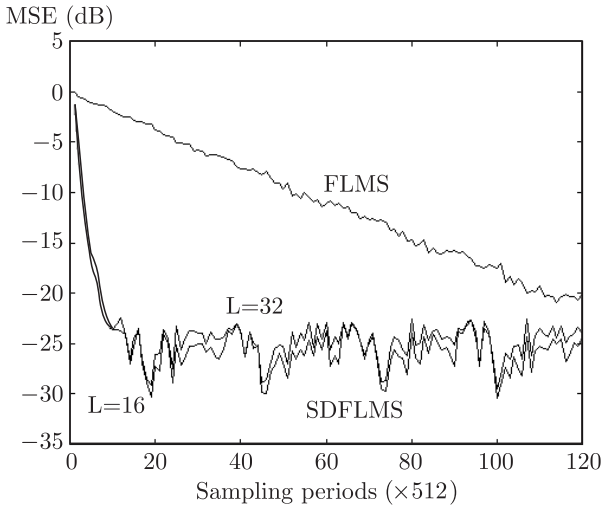


Fig. 11.10. Convergence performance of SDFLMS using 2 different block delays when required to identify an unknown system of order 128. The input signal was an actual speech signal with a signal-to-noise ratio equal to 20 dB. The convergence of FLMS is shown for comparison.

signal-to-noise ratio equal to 40 dB. The convergence performance of other previously proposed algorithms such as the MDF, FBAF, and JALG are also shown for comparison.

The evaluation results show that the SDFLMS provides better convergence performance than the conventional FLMS algorithm and other previously proposed FLMS algorithms with shorter processing delays.

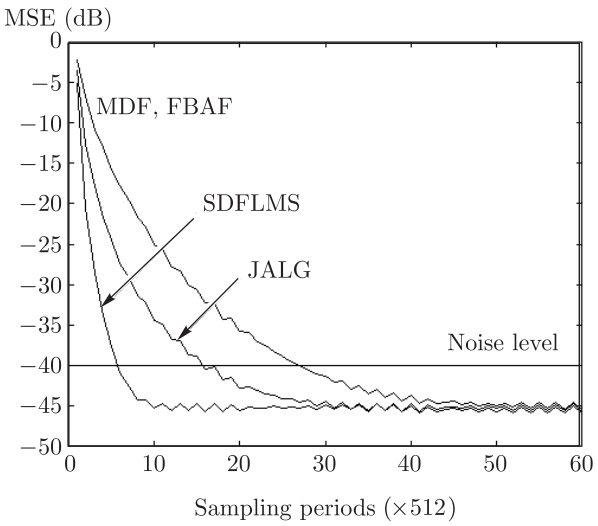


Fig. 11.11. Convergence performance of SDFLMS with 32 samples processing delays, when it is requiered to identify an unknown system of order 128. The input signal was a white noise signal with a sgnal-to-noise ratio equal to 40 dB. The convergence performance of FBAF, MDF and JALG are also shown for comparison.

Conclusions

This chapter presented parallel form adaptive filter structures based on a generalized subband decomposition, in which each subfilter can be updated independently, when RLS algorithms (SDADF) are used, or jointly, when the FLMS-type algorithms are used (SDFLMS). The SDADF structure has no numerical stability problems since the subband filters order updated using the modified RLS or Gauss-Newton algorithm is generally small. Computer simulations show that SDADF algorithms provide a convergence rate very close to that of the standard RLS adaptive algorithm or Gauss-Newton algorithm and a much lower computational cost, although with a greater misadjustment. A time varying convergence factor is used to reduce specially misadjustment, when the additive noise level is relatively large.

The SDFLMS algorithm uses the linear property of FFT to implement the FLMS algorithm using a linear combination of an actual FFT vector of order L and L–1 previously computed FFT vectors. This reduces the processing delay and increases the convergence rate, although the computational complexity increases. Computer simulations show that the SDFLMS excel in convergence rate the FLMS and other previously proposes FLMS-type algorithms.

References

1. *Haykin*, *S.*, Adaptive Filter Theory. Prentice Hall, Englewood Cliffs NJ, 1991.
2. *Messershmitt*, *D.* Echo Cancellation in Speech and Data Transmission. IEEE J. Of Selected Areas in Communications, vol. SAC-2, no. 2, pp. 283–297, March 1992.
3. *Nakano-Miyatake*, *M.*, *Perez-Meana*, *H.*, *Niño-de-Rivera*, *L.*, *Casco-Sanchez*, *F.*, *and Sanchez-Garcia*, *J.* A Time Varying Step Size Normalized LMS Algorithm for Adaptive Echo Canceler Structures. IEICE Trans. on Fundamentals, vol. E-78, no. 2, pp. 254–258, Feb. 1995.

4. *Pérez-Meana, H. and Amano, F.* Acoustic Echo Cancellation Using Multirate Techniques. IEICE Trans., vol. E74, no. 11, pp. 3559–3568, 1991.
5. *Amano, F., Perez-Meana, H., De Luca, A., and Duchen, G.*. A Multirate Acoustic Echo Canceler Structure. IEEE Trans. on Communications, vol. 43, No. 7, pp. 2172–2176, July 1995.
6. *Perez-Meana, H. and Tsujii S.* A System Identification Using Orthogonal Functions. IEEE Trans. on Signal Processing, vol. 39, no. 3, March 1991.
7. *Nakano-Miyatake, M., Perez-Meana, H., Ortiz-Balbuena, L., Niño-de-Rivera, L., and Sanchez, J.* A continuous Time RLS adaptive Filter Structure Using Hopfield Neural Networks, Proc. of ISITA'96, vol. II, pp. 614–617, Sept. 1996.
8. *Mariane R., Petraglia and Sanjit, K. Mitra* Adaptive FIR Filter Structure Based on the Generalized Subband Decomposition of FIR Filters. IEEE Trans. on Circuits and Systems-II, vol. 40, no. 6, pp. 354–362, June 1993.
9. *Tapia Sánchez, D., Bustamante, R., Pérez Meana, H., Nakano Miyatake, M.* Single Channel Active Noise Canceller Algorithm Using Discrete Cosine Transform. Journal of Signal Processing, vol. 9, no. 2 pp. 141–151, 2005.
10. *Perez-Meana, H., Nakano-Miyatake, M., Ortiz-Balbuena, L., Martinez-Gonzalez, A., and Sanchez-Garcia, J.* A Fast Block Type Adaptive Filter Algorithm with Short Processing Delay. IEICE Transactions on Fundamentals of Electronics,Communications and Computer Science, vol. E79 pp. 721–726, 1996.
11. *Perez Meana, H., Nakano Miyatake, N., and Niño de Rivera, L.* Adaptive Filtering Based on Subband Decomposition. Journal of Telecommunications and Radio Engineering, 2001, vol. 56, no. 4, pp. 160–173.
12. *Narayan, S., Peterson, A., and Narasimha, J.*, Transform domain LMS Algorithm. IEEE Trans. Acoust., Speech, Signal Processing, June 1983, vol. ASSP-31, no. 3, pp. 609–615.
13. *Panda, G., Mulgrew, B., Cowan, C., and Grant, P.* Orthogonalizing Efficient Block Adaptive Filter. IEEE Trans. on Acoustic, Speech and Signal Processing, 1986, vol. ASSP-34, pp. 1573–1581.
14. *Asharif M.R. and Amano, F.* Frequency Bin Adaptive Filtering (FBAF) Algorithm and Its Applications to Acoustic Echo Cancelling. Trans. of The IEICE, August 1991, vol. E74, no. 8, pp. 2276–2283.
15. *Asharif, M.R. and Amano, F.*, Acoustic Echo Cancellation Based on Frequency Bin Adaptive Filtering (FBAF), Proc. of the GLOBCOM'87, pp. 1940–1944, November 1987.
16. *Asharif, M.R., Amano, F., Unagami, S., and Murano, K.* A New Structure of Echo Canceller Based on Frequency Bin Adaptive Filtering (FBAF), Proc. of the IEICE Digital Signal Processing Symposium, November 1986, pp. 165–169.
17. *Soo, J.S. and Pang, K.K.* Multidelay Block Frequency Domain Adaptive Filter. IEEE Trans. on Acoustic, Speech and Signal Processing, Feb. 1995, vol. 38, no. 2, pp. 333–376.
18. *Honig M. and Messerschmitt D.*, Adaptive Filters: Structures, Algorithms and Applications, Kluwer Academic Publishers, Boston Mass. USA, 1984.
19. *Chao, J., Pérez Meana, H., and Tsujii, S.* A Jumping Algorithm for Adaptive Signal Processing. Trans. of the IEICE, July de 1990, vol. J73-A, pp. 1196–1206.
20. *Chao, J., Pérez Meana, H., and Tsujii, S.* A Fast Adaptive Filter Algorithm Using Eigenvalue Reciprocals as Step Size. IEEE Trans. on Signal Processing, October 1990, vol. SP-38, pp. 1343–1352.

Appendix

To further reduce the computational complexity of the SBDADF adaptive system shown in Figs. 1 and 2, an on-line implementation of the DCT would be required. To this end, consider the discrete cosine transform of an input signal $x(n)$ at time instants n,

$(n-1)$ and $(n-2)$, given by

$$C(n,r) = \alpha(k) \sum_{m=0}^{N-1} x(n-[m-1]+m) \cos(\pi(2m+1)r/2N), \tag{A.11.1}$$

$$C(n-1,r) = \alpha(k) \sum_{m=0}^{N-1} x(n-1-[m-1]+m) \cos(\pi(2n+1)r/2N), \tag{A.11.2}$$

$$C(n-2,r) = \alpha(k) \sum_{n=0}^{N-1} x(n-2-[m-1]+m) \cos(\pi(2n+1)r/2N). \tag{A.11.3}$$

Next, taking into account that

$$2\cos(a)\cos(b) = \cos(a-b) + \cos(a+b). \tag{A.11.4}$$

After some manipulations we obtain that the r-th component of the DCT of input signal can be estimated as the output signal of a filter bank whose r-th stage has a transfer function given by

$$C_r(z) = \frac{\cos\frac{\pi r}{2N}\left((-1)^r - (-1)^r z^{-1} - z^{-N} + z^{-N-1}\right)}{1 - 2\cos\frac{r\pi}{N} z^{-1} + z^{-2}}. \tag{A.11.5}$$

Chapter 12

IIR ADAPTIVE FILTER ALGORITHMS

12.1. Introduction

The finite impulse response adaptive filters (FIR-ADF) are widely used in most practical applications because they are unconditionally stable and their mean square error surface only has a global minimum [1]. However, the FIR-ADF has several limitations to model systems whose transfer functions have poles as well as zeros. These limitations become particularly important when the adaptive system is required to cancel acoustic echoes and multipath interferences and for several other applications in which the physical processes are properly modeled as the output of an infinite impulse response system [2].

The development of IIR adaptive filter (IIR-ADF) algorithms is motivated mainly by the potential reduction in the computational complexity and the ability to model sharp resonances with a much smaller number of filter coefficients, because they can model rational transfer functions in a more efficient way than the FIR-ADF, providing significantly better performance with same number of coefficients as its FIR counterparts [2, 3]. However, the IIR adaptive filters still present several drawbacks that limit their use in several practical applications, such as stability problems, convergence to one of several local minima of the mean-square error surface when gradient-search-based adaptive algorithms are used, slow convergence rates, etc [2, 3]. These problems must be solved before reliable IIR adaptive structures can be used in practical applications. To this end, over the last three decades, substantial research has been carried out to develop IIR-ADF algorithms capable to solve some of the above-mentioned problems. Among them, we have the equation error method based IIR adaptive algorithm [3], which has unimodal mean square error (MSE) surface, although its coefficients are based on the presence of additive noise and present stability problems [3]. To avoid the coefficients bias, several output error based methods have been proposed [3] such as the so-called recursive Gauss–Newton algorithm [3, 4], the recursive maximum likelihood algorithms (RML) [4, 5], the recursive instrumental variable method (RIV) [4, 5], the recursive prediction error algorithm [3], the approximate gradient methods [3], etc. All of them solve the bias coefficients problem, using the inverse of Hessian matrix, although they still present stability problems and convergence to a local minimum of the MSE surface. To reduce the convergence problems, the HARF (Hyperstable adaptive recursive filter) and SHARF (Simplified hyperstable recursive filter) were proposed [2, 3]. The HARF and SHARF algorithms guarantee the convergence if the strictly positive real (SPR) condition is satisfied. However, in most cases the SPR condition is not satisfied [3]. All of these algorithms use a direct realization form with a difficult stability test. Besides, the computational complexity is due, mainly, to the estimation of the inverse of Hessian matrix.

Several other approaches have been proposed to solve the stability problems without using a direct realization form. One of them is an IIR structure with fixed poles derived using a set of orthogonal functions [6]. However, although this structure does

not have stability problems and its MSE surface is unimodal, it provides a suboptimal solution. IIR adaptive filter structures using parallel and lattice structures have also been proposed [3, 7]. These structures, although with a trivial stability test, does not guarantee the convergence to the global minimum of the MSE surface. Polyphase IIR adaptive structures [8] have also been proposed, with convergence much higher than the achieved by conventional IIR-ADF structures, although the global convergence and stability problems still remain. Thus, although there have been proposed a relatively large number of IIR-ADF, several problems associated with the IIR-ADF that must be solved still remain, namely, slow convergence, potential filter instability, mean square-error function with multiple local minima, etc. [9–13].

Finally, to overcome some of the above-mentioned problems, a family of adaptive algorithms combining the equation error and output error methods, called Steiglitz-MacBride methods, using either direct or lattice realization forms,have been proposed [11]. Among them, some of the most successful algorithms use a cascade of FIR transversal filter and AR lattice structures. These filter structures, which can be updated using either the Steiglitz–MacBride method or a gradient-based adaptive algorithm, have trivial stability test, although, in general, the lattice-based adaptive filters have a larger computational complexity than the direct realization form IIR-ADF structures.

This chapter presents a review of IIR adaptive filters without stability problems, using orthogonalized structures, as well as a low complexity cascade lattice IIR adaptive filter algorithms using the Steiglitz–MacBride method, and a gradient-search based adaptive algorithm, in which the derivatives involved in the adaptation processes are estimated using the simultaneous perturbation stochastic approximation. Computer simulations are given to show the actual performance of presented IIR adaptive algorithms.

12.2. Adaptive Filters Based on Equation Error Method

The widely used IIR adaptive filter algorithm is based on the equation error in which, during the adaptation process, it is assumed that, after convergence is achieved, the filter output $y(n)$ becomes very close to the reference signal $d(n)$. Thus, using $d(n)$ instead of $y(n)$ during the adaptation process, the MSE surface becomes a quadratic function of the coefficients vector, although, due to the presence of additive noise, this assumption does not always lead in this situation to a biased solution.

Consider the adaptive filter output given by [4]

$$y(n) = \mathbf{W}^T(n)\mathbf{X}(n), \tag{12.1}$$

where

$$\mathbf{X}(n) = [x(n), x(n-1), x(n-2), \ldots, \ldots, x(n-N+1), y(n-1), y(n-2), \ldots, y(n-M+1)]^T \tag{12.2}$$

is the input vector and

$$\mathbf{W}(n) = [a_0(n), a_1(n), \ldots, a_{N-1}(n), b_1(n), b_2(n), \ldots, b_{M-1}(n)]^T \tag{12.3}$$

is the coefficients vector.

12.2.1. Adaptive Algorithm. Consider the output error given by

$$e(n) = d(n) - y(n), \tag{12.4}$$

where $y(n)$ is given by (12.1), $d(n)$ is the reference signal, and $\mathbf{W}(n)$ is the coefficients vector which will be updated so that the mean-square value of $e(n)$ attains a minimum.

However a direct minimization is a nonlinear problem because the input vector $\mathbf{X}(n)$ is a function of $\mathbf{W}(n)$. To solve this problem, we assume that after convergence the filter output $y(n)$ becomes close to the reference signal $d(n)$ and then $y(n)$ is becomes approximately given by [4]

$$y(n) = \mathbf{W}^T(n)\mathbf{\Psi}(n), \tag{12.5}$$

where,

$$\mathbf{\Psi}(n) = [x(n), x(n-1), x(n-2), \ldots, \\ \ldots, x(n-N+1), d(n-1), d(n-2), \ldots, d(n-M+1)]^T. \tag{12.6}$$

The optimal coefficients vector $\mathbf{W}(n)$ will be obtained from the Wiener–Hopf equation, that is

$$\frac{\partial}{\partial \mathbf{W}} E\left[\left(d(n) - \mathbf{W}^T(n)\Psi(n)\right)^2\right] = 0. \tag{12.7}$$

This expression is the same equation used to update the coefficients vector in the FIR filter. Then $\mathbf{W}(n)$ can be updated using either the RLS or LMS algorithms presented in other chapter. Thus, if the RLS algorithm is used, $\mathbf{W}(n)$ becomes

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \mathbf{Q}(n)e(n)\Psi(n), \tag{12.8}$$

where

$$\mathbf{Q}(n) = \frac{1}{\lambda}[\mathbf{Q}(n-1) - \frac{\mathbf{Q}(n-1)\mathbf{X}(n)\mathbf{X}^T(n)\mathbf{Q}(n-1)}{\lambda + \mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)}]. \tag{12.9}$$

On the other hand, if the LMS algorithm is used, the coefficients vector becomes [4]

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \mu e(n)\Psi(n). \tag{12.10}$$

12.3. Adaptive Filters Based on the Output Error Method

Consider that $d(n)$ and $y(n)$ denote the reference and adaptive filter output, respectively, such that the output error is given by [4]

$$e(n) = d(n) - y(n), \tag{12.11}$$

where

$$y(n) = \mathbf{W}^T(n)\mathbf{X}(n) \tag{12.12}$$

denotes the output signal,

$$\mathbf{X}(n) = [x(n), x(n-1), x(n-2), \ldots, \\ \ldots, x(n-N+1), y(n-1), y(n-2), \ldots, y(n-M+1)]^T \tag{12.13}$$

is the input vector,

$$\mathbf{W}(n) = [a_0(n), a_1(n), \ldots, a_{N-1}(n), b_1(n), b_2(n), \ldots, b_{M-1}(n)]^T \tag{12.14}$$

is the coefficients vector, which will be estimated so that [4]

$$\varepsilon\left(\mathbf{W}, n\right) = \frac{1}{2n}\sum_{k=1}^{n} e^2(k) \tag{12.15}$$

attains a minimum. However, since $\varepsilon(\mathbf{W},n)$ is a nonlinear function of $\mathbf{W}$, $\varepsilon(\mathbf{W},n)$ must be numerically minimized. To this end, consider the Taylor series expansion of $\varepsilon(\mathbf{W},n)$ around $\mathbf{W}(n-1)$ which is given by [4]

$$\varepsilon(\mathbf{W}(n)) = \varepsilon(\mathbf{W}(n-1)) + \varepsilon'(\mathbf{W}(n-1))[\mathbf{W}-\mathbf{W}(n-1)]+ \\ + (1/2)[\mathbf{W}-\mathbf{W}(n-1)]^T\varepsilon''(\mathbf{W}(n-1))[\mathbf{W}-\mathbf{W}(n-1)]+ \\ + o\left[|\mathbf{W}-\mathbf{W}(n-1)|^2\right]. \quad (12.16)$$

Minimizing (12.16) with respect to $\mathbf{W}(n)$, we obtain [4]

$$\mathbf{W}(n) = \mathbf{W}(n-1) - \left[\varepsilon''(\mathbf{W}n-1))\right]^{-1}\left[\varepsilon'(\mathbf{W}(n-1))\right]^T + o\left[|\mathbf{W}-\mathbf{W}(n-1)|\right]. \quad (12.17)$$

Next, define $\mathbf{\Psi}(n)$ as the derivative of $e(n)$ with respect to the coefficients vector, that is

$$\Psi(n) = -\left[\frac{\partial e(n)}{\partial \mathbf{W}}\right]^T, \quad (12.18)$$

and then

$$\varepsilon'(\mathbf{W}(n)) = \frac{1}{2n}\sum_{k=1}^{n}\frac{\partial e^2(k)}{\partial \mathbf{W}}, \quad (12.19)$$

$$\varepsilon'(\mathbf{W}(n)) = \frac{1}{n}\sum_{k=1}^{n} e(k)\frac{\partial e(k)}{\partial \mathbf{W}}, \quad (12.20)$$

$$\varepsilon'(\mathbf{W}(n)) = -\frac{1}{n}\sum_{k=1}^{n} e(k)\Psi(\mathbf{W},k). \quad (12.21)$$

In a similar form, from (12.21) it follows that [4]

$$\varepsilon'(\mathbf{W}(n-1)) = -\frac{1}{n-1}\sum_{k=1}^{n-1} e(k)\Psi(\mathbf{W},k) \quad (12.22)$$

and then, from (12.21) and (12.22), it follows that

$$\varepsilon'(\mathbf{W}(n)) = -\frac{1}{n-1}.\frac{n-1}{n}\sum_{k=1}^{n-1} e(k)\Psi(\mathbf{W},k) - \frac{1}{n}e(n)\Psi(\mathbf{W},n), \quad (12.23)$$

$$\varepsilon'(\mathbf{W}(n)) = -\frac{1}{n-1}\sum_{k=1}^{n-1} e(k)\Psi(\mathbf{W},k)- \\ - \frac{1}{n}\left\{-\frac{1}{n-1}\sum_{k=1}^{n-1} e(k)\Psi(\mathbf{W},k)+e(n)\Psi(\mathbf{W},n)\right\}, \quad (12.24)$$

$$\varepsilon'(\mathbf{W}(n)) = \varepsilon'(\mathbf{W}(n-1) - \frac{1}{n}\left\{\varepsilon'(\mathbf{W}(n-1)) + e(n)\Psi(\mathbf{W},n)\right\}. \quad (12.25)$$

Now, taking the derivative of (12.25) with respect to $\mathbf{W}$, we obtain [4]

$$\varepsilon''(\mathbf{W}(n)) = \varepsilon''(\mathbf{W}(n-1))- \\ - \frac{1}{n}\left\{\varepsilon''(\mathbf{W}(n-1)) - \Psi(\mathbf{W}(n))\Psi^T(\mathbf{W}(n)) + e(n)e''(n)\right\}. \quad (12.26)$$

Next, to evaluate (12.26), the following assumptions must be done [4]:

1. Assuming that $\mathbf{W}(n)$ is in the neighborhood of $\mathbf{W}(n-1)$, this approach must be good enough for large values of n. In this situation, this approach also leads to the following assumptions: Neglect $o\left[\|\mathbf{W}-\mathbf{W}(n-1)\|\right]$ in (12.16) and (12.17) and assume that $\varepsilon''(\mathbf{W}(n))=\varepsilon''(\mathbf{W}(n-1))$.
2. Assume that $\mathbf{W}(n-1)$ is the optimal value estimated at time n-1, such that $\varepsilon'(\mathbf{W}(n-1))=0$.
3. Finally, assume that $e''(n)e(n)=0$. The reasoning behind this is the fact that, close to the optimum, the output error becomes nearly white and then the expectation of $e(n)e(n-1)$ becomes close to zero.

With these assumptions, from (12.26) it follows that [4]

$$\varepsilon''(\mathbf{W}(n))=\varepsilon''(\mathbf{W}(n-1))+\frac{1}{n}\left\{\Psi(\mathbf{W},n)\Psi^T(\mathbf{W},n)-\varepsilon''(\mathbf{W}(n-1))\right\}, \tag{12.27}$$

$$\mathbf{R}(n)=\mathbf{R}(n-1)+\frac{1}{n}\left\{\Psi(\mathbf{W},n)\Psi^T(\mathbf{W},n)-\mathbf{R}(n-1)\right\}, \tag{12.28}$$

where $\mathbf{R}(n)$ denotes an approximation of $\varepsilon''(\mathbf{W}(n))$. Next, using assumption 2 in (12.25), we obtain

$$\varepsilon'(\mathbf{W}(n))=-\frac{1}{n}e(n)\Psi(\mathbf{W},n). \tag{12.29}$$

Finally usign assuption 1, from (17) we obtain [4]

$$\mathbf{W}(n)=\mathbf{W}(n-1)+\mathbf{R}(n)^{-1}\Psi(\mathbf{W},n)e(n), \tag{12.30}$$

where [4]

$$\mathbf{R}(n)=(1-\gamma)\mathbf{R}(n-1)+\gamma\,\Psi(\mathbf{W},n)\Psi^T(\mathbf{W},n), \tag{12.31}$$
$$0<\gamma<1.$$

Several forms for estimating $\mathbf{R}^{-1}(n)$ can be used as the input data instead of estimating $\mathbf{R}(n)$ and then inverting it. One of the most used methods is the matrix inversion lemma which establish that

$$[\mathbf{A}+\mathbf{BCD}]=\left[\mathbf{A}^{-1}+\mathbf{A}^{-1}\mathbf{B}\left[\mathbf{D}\mathrm{A}^{\cdot 1}\mathbf{B}+\mathbf{C}^{-1}\right]^{-1}\right]. \tag{12.32}$$

Then, setting $\mathbf{A}=(1-\gamma)\,\mathbf{R}(n-1)$, $\mathbf{B}=\boldsymbol{\psi}$, $\mathbf{C}=\gamma$, and $\mathbf{D}=\boldsymbol{\psi}$, from (12.31) and (12.32) we obtainn [4]

$$\mathbf{R}^{-1}(n)=\frac{1}{(1-\gamma)}\left(\mathbf{R}^{-1}-\frac{\mathbf{R}^{-1}\boldsymbol{\psi}(\mathbf{W},n)\boldsymbol{\psi}^T(\mathbf{W},n)\mathbf{R}^{-1}}{\boldsymbol{\psi}^T(\mathbf{W},n)\mathbf{R}\boldsymbol{\psi}(\mathbf{W},n)-(1-\gamma)/\gamma}\right), \tag{12.33}$$

where

$$\psi_k(\mathbf{W},n)=x(n-k)+\sum_{m=1}^{M-1}b_m\psi_k(\mathbf{W},n-m)\quad k=0,2,\ldots,N-1, \tag{12.34}$$

$$\psi_{N+k}(\mathbf{W},n)=y(n-k)+\sum_{m=1}^{M-1}b_m\psi_{N+k}(\mathbf{W},n-m)\quad k=1,2,\ldots,M-1 \tag{12.35}$$

and b_m is the (N+m)-th coefficient of vector $\mathbf{W}$ given by (12.14).

12.4. Orthogonalized IIR Adaptive Filter Algorithms

IIR adaptive filtering has been a subject of active research during the last three decades and considered as a desirable alternative to conventional adaptive filters struc-

tures for the solution of several practical problems, such as echo and noise canceling, linear prediction, etc.[14]. During this time, several algorithms have appeared in literature, most of them using the direct realization form [15–17]. Unfortunately, the stability test generally required after each filter update can be computationally expensive [1, 2]. This problem was solved using parallel and cascade realization forms that allow a trivial stability test, although, in some situations, algorithms could experience some convergence problems and slow convergence rate. However, their low computational complexity to perform their stability test does them a desirable alternative to IIR direct realization form.

On the other hand, in real time signal processing, a significant amount of computational effort can be saved if the input signals are represented in terms of a set of orthogonal signal components [7, 21]. The reason is that the representation admits processing schemes in which each of these signal components can be independently processed.

Taking this fact into account, this chapter presents an IIR structure based on the output error method in which the input signals are split into a set of approximately orthogonal signal components by using the discrete cosine transform. Subsequently an IIR-ADF is inserted in each subband whose parameters are independently updated to minimize the total error. The computer simulation results show that all the proposed types of subband adaptive filter structures reduce computational complexity considerably with a fairly good convergence property.

12.4.1. Adaptive filter structure. Consider the transfer function H(z) of the direct form IIR filter which is given by

$$H(z) = \frac{A(z)}{B(z)}, \tag{12.36}$$

where $A(z)$ and $B(z)$ are polynomials of order L and N, respectively, $N > L$ and $N/2$ is even. Using the partial fraction expansion technique, $H(z)$ can be rewritten as

$$H(z) = H_1(z) + H_2(z), \tag{12.37}$$

$$H(z) = \frac{A_1(z)}{B_1(z)} + \frac{A_2(z)}{B_2(z)}, \tag{12.38}$$

where $B_1(z)$ and $B_2(z)$ are polynomials of order $N/2$. Next, taking into account that

$$(1 - z^{-M}) = \prod_{r=0}^{M/2-1} (1 - 2\cos(2\pi/M)z^{-1} + z^{-2}) \tag{12.39}$$

and

$$(1 + z^{-M}) = \prod_{r=0}^{M/2-1} (1 - 2\cos(\pi(2r+1)/M)z^{-1} + z^{-2}), \tag{12.40}$$

after some manipulation, we get the parallel form IIR structure given by [20]

$$H(z) = \sum_{r=0}^{M-1} C_r(z)G_r(z), \tag{12.41}$$

where

$$C_r(z) = \frac{[1 - z^{-1}][1 - (-1)^r z^{-M}]}{1 - 2\cos(\pi r/M)z^{-1} + z^{-2}}, \quad r = 0, 1, \ldots, M-1 \tag{12.42}$$

$$G_r(z) = \frac{a_{0r} + a_{1r}z^{-1} + a_{2r}z^{-2}}{1 - z^{-1}}, \quad r = 0, 1, \tag{12.43}$$

and

$$G_r(z) = \frac{a_{or} + a_{1r}z^{-1} + a_{2r}z^{-2} + a_{3r}z^{-3}}{1 - b_{1r}z^{-1} - b_{2r}z^{-2}}, \quad r = 2, 3, \ldots, M-1. \tag{12.44}$$

The orthogonarized parallel IIR structure is shown in Figs. 12.1 and 12.2, respectively [8].

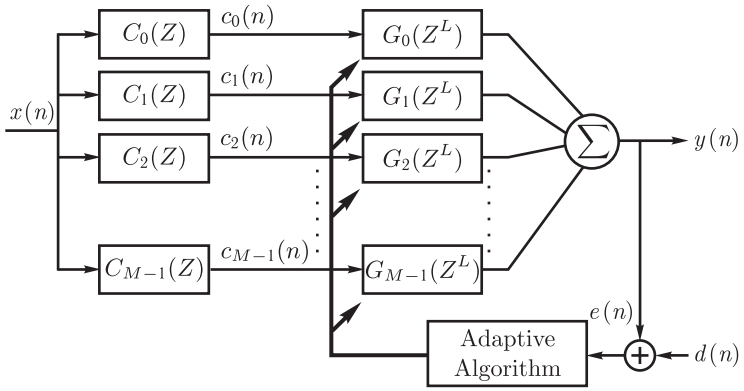


Fig. 12.1. Orthogonalized IIR ADF structure based on subband decomposition using DCT.

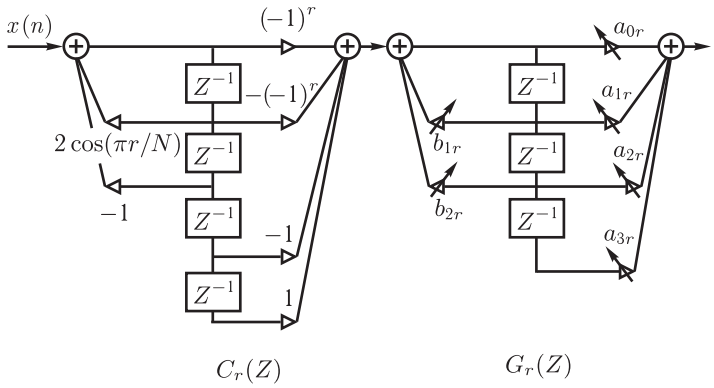


Fig. 12.2. r-th stage of orthogonalized IIR ADF structure based on subband decomposition using DCT.

12.4.2. Adaptive Algorithm. The adaptation algorithm used to update the filter coefficients is a modified form of the Gauss–Newton algorithm presented in Section 12.3, derived under the assumption that the DCT coefficients are fully uncorrelated. Under this assumption, each IIR ADF can be updated independently. Then the $4M \times 4M$ Hessian matrix of the parallel form Gauss–Newton algorithm can be replaced by a block

diagonal matrix where each block has rank= 6, that is

$$\mathbf{R}(n) = \begin{bmatrix} \mathbf{R}_0(n) & 0 & 0 & 0 & \cdot & 0 \\ 0 & \mathbf{R}_1(n) & 0 & 0 & \cdot & 0 \\ 0 & 0 & \mathbf{R}_2(n) & 0 & \cdot & 0 \\ 0 & 0 & 0 & \mathbf{R}_3(n) & 0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & 0 & 0 & \mathbf{R}_{M-1}(n) \end{bmatrix}, \tag{12.45}$$

where

$$\mathbf{R}_r(n) = \sum_{k=1}^{n} \lambda^{n-k} \mathbf{U}_r(k) \mathbf{U}_r^T(k), \tag{12.46}$$

is the r-th Hessian matrix estimated from the input data, λ is the forgetting factor, and $U_r(n)$ is the information vector whose elements correspond to the estimated gradient of $y(n)$ for a_{ir} $(i = 0, 1, 2, 3)$ and b_{jr} $(j = 1, 2)$, respectively. Then the modified parallel form Gauss–Newton algorithm becomes

$$A_r(n) = A_r(n-1) + \mu \mathbf{R}_r^{-1}(n) e(n) \mathbf{U}_r(n), \tag{12.47}$$

where $0 < \mu < 1$ is the convergence factor, which controls the stability and convergence rate,

$$\mathbf{U}_r(n) = [u_{0,r}(n), u_{1,r}(n), u_{2,r}(n), u_{3,r}(n), u_{4,r}(n), u_{5,r}(n)]^T, \tag{12.48}$$

is the information vector, whose elements given by

$$u_{k,r}(n) = \frac{\partial y(n)}{\partial a_{k,r}}, \quad k = 0, 1, 2, 3, \tag{12.49}$$

$$u_{k,r}(n) = \frac{\partial y(n)}{\partial b_{k,r}}, \quad k = 4, 5, \tag{12.50}$$

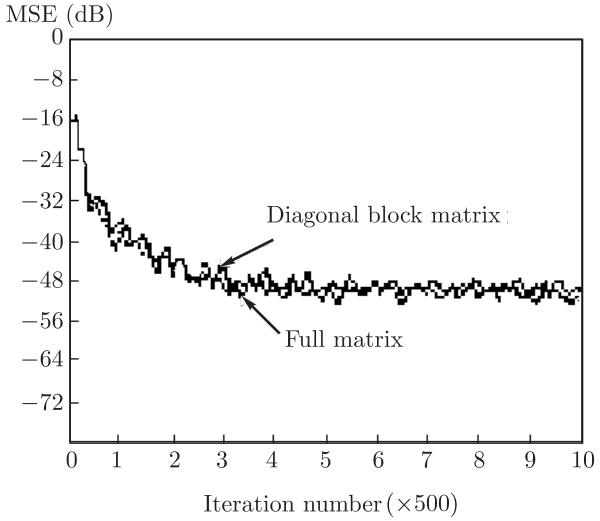


Fig. 12.3. Convergence performance of orthogonalized IIR structure based on output error method when it is required to identify an unknown system of order 32. Input signal is AR process.

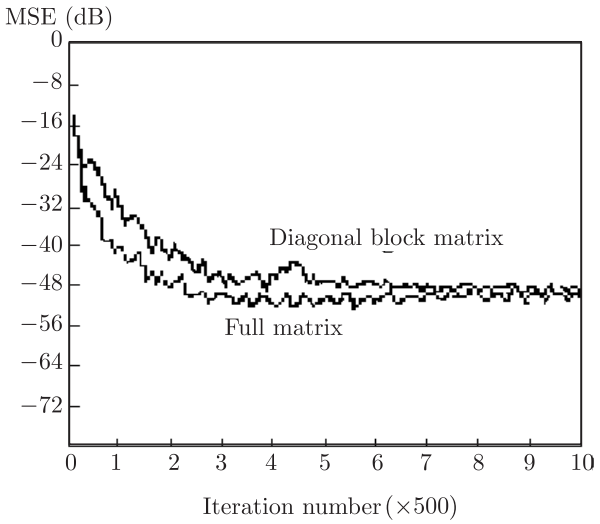


Fig. 12.4. Convergence performance of the orthogonalized IIR structure based on the output error method when it is required to identify an unknown system of order 32. Input signal is speech signal.

can be estimated as follows [3, 7]:

$$u_{k,r}(n) = c_r(n-k) + b_{1,r}(n)u_{k,r}(n-1) + b_{2,r}(n)u_{k,r}(n-2), \quad k = 0,1,2,3, \tag{12.51}$$

$$u_{k,r}(n) = y_r(n-k+3) + b_{1,r}(n)u_{k,r}(n-1) + b_{2,r}(n)u_{k,r}(n-2), \quad k = 4,5, \tag{12.52}$$

and $\mathbf{R}_r^{-1}(n)$ is the inverse of Hessian matrix, given by [4]

$$\mathbf{R}_r^{-1}(n) = \frac{\mathbf{R}_r^{-1}(n-1)}{\lambda} - \frac{\mathbf{R}_r^{-1}(n-1)\mathbf{U}_r(n)\mathbf{U}_r^T(n)\mathbf{R}_r^{-1}(n-1)}{\lambda^2 + \mathbf{U}_r^T(n)\mathbf{R}_r^{-1}(n-1)\mathbf{U}_r(n)}. \tag{12.53}$$

12.4.3. Computer Simulations. Figures 12.3 and 12.4 show the convergence performance of the orthogonalized parallel IIR structure based on the output error method, when it is required to identify an unknown system of order 32. In Fig. 12.3 the input signal was a 12-th order AR process, and in Fig. 12.4 it was an actual speech signal. In both cases the convergence factor was equal to 0.1. The convergence performance obtained using a parallel structure, with a full autocorrelation matrix and convergence factor equal to 1, is also shown for comparison

12.5. Cascade Lattice IIR Adaptive Filter

Figure 12.5 shows the IIR structure which consists of a FIR transversal filter to implement the feed forward sections and a lattice structures to realize the feedback sections. The reason for using a lattice structure to implement the feedback section is the fact that, using it, the IIR structure will remain stable if the absolute value of the reflection coefficients is kept less than one.

To derive the transfer function of the cascade lattice structure shown in Fig.12.5, consider the all-pole lattice structure shown in Fig. 12.6. From this figure we have

$$v_0(n) = y(n), \tag{12.54}$$

$$v_1(n) = v_0(n) + k_1 x_1(n), \tag{12.55}$$

$$v_2(n) = v_1(n) + k_2 x_2(n). \tag{12.56}$$

Substituting (12.53) and (12.55) into (12.56), we obtain

$$v_2(n) = y(n) + k_1 x_1(n) + k_2 x_2(n). \tag{12.57}$$

In a similar manner, from Fig. 12.6 and equations (12.54)–(12.56) it follows that

$$v_3(n) = v_2(n) + k_3 x_3(n), \tag{12.58}$$

$$v_3(n) = y(n) + \sum_{i=1}^{3} k_i x_i(n). \tag{12.59}$$

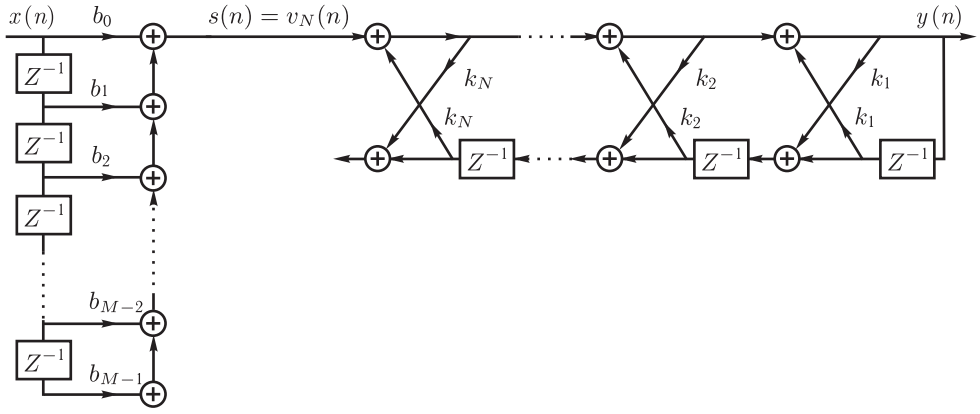


Fig. 12.5. Cascade lattice IIR filter structure.

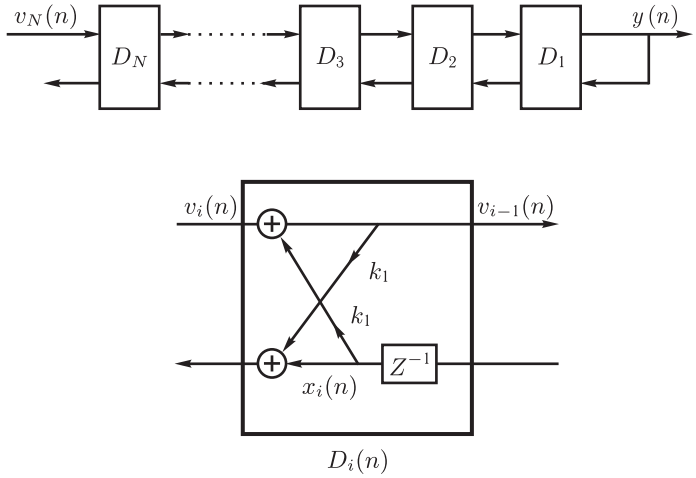


Fig. 12.6. All-pole lattice structure.

Then, in general, it follows that $v_m(n)$is given by

$$v_m(n) = y(n) + \sum_{i=1}^{m} k_i x_i(n). \tag{12.60}$$

Next, consider the expression for$x_i(n)$, i=1,2,..., which is given from Fig. 12.6 by

$$x_1(n) = y(n-1), \tag{12.61}$$
$$x_2(n) = k_1 y(n-1) + y(n-2). \tag{12.62}$$

Equations (12.61) and (12.62) can be written as

$$\begin{pmatrix} x_1(n) \\ x_2(n) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ k_1 & 1 \end{pmatrix} \begin{pmatrix} y(n-1) \\ y(n-2) \end{pmatrix}, \tag{12.63}$$
$$\begin{pmatrix} x_1(n) \\ x_2(n) \end{pmatrix} = \begin{pmatrix} \mathbf{T}_1 & 0 \\ k_1 & \mathbf{T}_1 \end{pmatrix} \begin{pmatrix} y(n-1) \\ y(n-2) \end{pmatrix}, \tag{12.64}$$

where $\mathbf{T}_1 = 1$. Next, consider the expression of $x_3(n)$, which is given by

$$x_3(n) = k_2 y(n-1) + (k_1 k_2 + k_1) y(n-2) + y(n-3) \tag{12.65}$$

which, using equation (12.64), can be written as

$$\begin{pmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ k_1 & 1 & 0 \\ k_2 & k_1 + k_1 k_2 & 1 \end{pmatrix} \begin{pmatrix} y(n-1) \\ y(n-2) \\ y(n-3) \end{pmatrix}, \tag{12.66}$$
$$\begin{pmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} 1 & 0 \\ k_1 & 1 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ k_2, (k_1 k_2, \ 1) & \begin{pmatrix} 1 & 0 \\ k_1 & 1 \end{pmatrix} \end{pmatrix} \begin{pmatrix} y(n-1) \\ y(n-2) \\ y(n-3) \end{pmatrix}, \tag{12.67}$$
$$\begin{pmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \end{pmatrix} = \begin{pmatrix} \mathbf{T_2} & \mathbf{0} \\ k_2, (k_1 k_2, 1) & \mathbf{T_2} \end{pmatrix} \begin{pmatrix} y(n-1) \\ y(n-2) \\ y(n-3) \end{pmatrix}, \tag{12.68}$$

where

$$\mathbf{T}_2 = \begin{pmatrix} 1 & 0 \\ k_1 & 1 \end{pmatrix}. \tag{12.69}$$

In a similar manner, we get

$$\begin{pmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \\ x_4(n) \end{pmatrix} = \begin{pmatrix} \mathbf{T}_3 & \mathbf{0} \\ k_3, (k_3 k_1, k_3 k_1, 1) & \mathbf{T}_3 \end{pmatrix} \begin{pmatrix} y(n-1) \\ y(n-2) \\ y(n-3) \end{pmatrix}, \tag{12.70}$$
$$\mathbf{T}_3 = \begin{pmatrix} \mathbf{T}_2 & \mathbf{0} \\ k_3, (k_3 k_1, k_3 k_1, 1) & \mathbf{T}_3 \end{pmatrix}. \tag{12.71}$$

Thus, from equations (12.61)–(12.71) it follows that

$$\begin{pmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \\ \vdots \\ x_m(n) \end{pmatrix} = \begin{pmatrix} \mathbf{T}_{m-1} & \mathbf{0} \\ k_{m-1} & \mathbf{K}_{m-1}\mathbf{T}_{m-1} \end{pmatrix} \begin{pmatrix} y(n-1) \\ y(n-2) \\ y(n-3) \\ \vdots \\ y(n-m) \end{pmatrix}, \tag{12.72}$$
$$\mathbf{X}_m(n) = \mathbf{T}_m \mathbf{Y}_m(n), \tag{12.73}$$

where $m = 1, 2, 3, \ldots, N$,

$$\mathbf{T}_m = \begin{pmatrix} \mathbf{T}_{m-1} & \mathbf{0} \\ k_{m-1} & \mathbf{K}_{m-1}\mathbf{T}_{m-1} \end{pmatrix}, \tag{12.74}$$

$$\mathbf{K}_{m-1} = [k_{m-1}k_1, k_{m-1}k_2 \ldots k_{m-1}k_{m-2}, 1]^T, \tag{12.75}$$

$$\mathbf{X}_m(n) = [x_1(n), x_2(n), x_3(n), \ldots, x_m(n)]^T, \tag{12.76}$$

$$\mathbf{Y}_m(n) = [y(n-1), y(n-2), \ldots, y(n-m)]^T. \tag{12.77}$$

Finally, from equations (12.60), (12.72), and (12.77) we obtain

$$v_N(n) = y(n) + \mathbf{K}^T\mathbf{X}_N(n). \tag{12.78}$$

Next, substituting $\mathbf{X}_N(n)$, which is given by (12.73)–(12.77) with $m = N$, into (12.78), we obtain

$$v_N(n) = y(n) + \mathbf{K}^T\mathbf{T}_N\mathbf{Y}_N, \tag{12.79}$$

$$y(n) = v_N(n) - \mathbf{K}^T\mathbf{T}_N\mathbf{Y}_N, \tag{12.80}$$

where

$$\mathbf{K} = [k_1, k_2, k_{3,} \ldots, k_N]^T. \tag{12.81}$$

Next, using the delay operator q^{-1}, from Fig. 12.5 and equations (12.77), (12.80), and (12.81) it follows that

$$y(n) = s(n) - [k_1, k_2, \ldots, k_N]\,\mathbf{T}_N \begin{pmatrix} q^{-1} \\ q^{-2} \\ \vdots \\ q^{-N} \end{pmatrix} y(n), \tag{12.82}$$

where s(n) from Fig. 14.5 is given by

$$s(n) = [b_0, b_1, b_2, \ldots, b_{M-1}] \begin{pmatrix} 1 \\ q^{-1} \\ q^{-2} \\ \vdots \\ q^{M-1} \end{pmatrix} x(n). \tag{12.83}$$

Finally, substituting equation (12.83) into (12.82), we get

$$y(n) = B(q^{-1})x(n) - A(q^{-1})y(n), \tag{12.84}$$

where

$$A(q^{-1})y(n) = [k_1, k_2, \ldots, k_N]\,\mathbf{T}_N \begin{pmatrix} y(n-1) \\ y(n-2) \\ \vdots \\ y(n-N) \end{pmatrix} \tag{12.85}$$

and

$$B(q^{-1})x(n) = [b_0, b_1, \ldots, b_{M-1}] \begin{pmatrix} x(n) \\ x(n-1) \\ x(n-2) \\ \vdots \\ x(n-M+1 \end{pmatrix}. \tag{12.86}$$

Several algorithms have been proposed to update the IIR adaptive filters coefficients vector. Among them the gradient-search-based adaptive algorithm and the Steiglitz–McBride type IIR adaptive algorithm are two of the most widely used adaptive algorithms, which are derived under the assumption that the adaptive filter is of strictly sufficient order, the filter structure is persistently excited, the measurement noise has bounded variance, it is uncorrelated with the input signal $x(n)$, and, finally, the adaptive filter is assumed to be stable [3, 9–12]. Next sections describe the adaptive algorithms used to update the IIR filter coefficients vector.

12.5.1. IIR gradient algorithm. When a gradient search-based-algorithm is used to update the IIR adaptive filter coefficients vector, the adaptive filter coefficients at time instant n is given by [9]

$$k_i(n+1) = k_i(n) + \mu\, e(n) \frac{\partial y(n)}{\partial k_i}, \tag{12.87}$$

$$b_j(n+1) = b_j(n) + \mu\, e(n) \frac{\partial y(n)}{\partial b_j}, \tag{12.88}$$

where $e(n)$ is the output error and

$$\frac{\partial y(n)}{\partial k_i} = \alpha_i(n) = -x_i(n) - \mathbf{K}^T \mathbf{T}_N(n) \begin{pmatrix} \alpha_i(n-1) \\ a_i(n-2) \\ \vdots \\ \alpha_i(n-N) \end{pmatrix}, \tag{12.89}$$

$$\frac{\partial y(n)}{\partial b_j} = \beta_j(n) = x(n-j) + \mathbf{K}^T \mathbf{T}_N(n) \begin{pmatrix} \beta_j(n-1) \\ \beta_j(n-2) \\ \vdots \\ \beta_j(n-N) \end{pmatrix}, \tag{12.90}$$

where $i = 1, 2, \ldots, N$; $j = 0, 1, 2, \ldots, M-1$; and $\mathbf{T}_N$ and $\mathbf{K}$ are given by equations (12.74) and (12.81), respectively. A summary of the IIR gradient algorithm is shown in Table 12.1.

12.5.2. IIR SPSA-based gradient algorithm. The IIR gradient algorithm describe in Section 12.5.1 is widely used to update the IIR adaptive filter coefficients vectors. However, the gradient estimation requires a considerable computational effort because each derivative requires the computation of the output of an all pole lattice structure, as shown in equations (12.89) and (12.90). To reduce the number of numerical operations required to compute these equations, the simultaneous perturbation stochastic approach can be used.

The simultaneous perturbation stochastic approximation (SPSA) [22, 23] is a very low computational complexity adaptive algorithm that can be used to solve difficult multivariate optimization problem, such as given by equations (12.87)–(12.90), efficiently. To this end, firstly, define a perturbation vector $\mathbf{C}_n$, given at time instant n by

$$\mathbf{C}_n = (r_n^1, r_n^2, \ldots, r_n^N, p_n^0 \ldots p_n^{M-1})^T. \tag{12.91}$$

This vector will be used to estimate the derivatives required by equations (12.87) and (12.88), where n denotes the time index. Here, the perturbation vector components r_n^i and p_n^i are random numbers in the interval $([-c_{\max}, -c_{\min}], [c_{\min}, c_{\max}])$, with cero mean and mutually uncorrelated. Next, define the following coefficients vectors [14, 15]:

$$\mathbf{K}_p = \left[k_1 + r_n^1, k_2 + r_n^2, k_{3,} + r_n^3, \ldots, k_N + r_n^N\right]^T, \tag{12.92}$$

Table 12.1. Cascade lattice IIR adaptive algorithm.

For each input data do

$\mathbf{X}_N(n) = [x_1(n), x_2(n), x_3(n), \ldots, x_N(n)]^T$

$\mathbf{K}_{m-1} = [k_{m-1}, (k_{m-1}k_1, k_{m-1}k_2, \ldots, k_{m-1}k_{m-2}, 1)]^T$

$\mathbf{X}(n) = [x(n), x(n-1), x(n-2), \ldots, x(n-M-1)]^T$

$\mathbf{Y}_N(n) = [y(n-1), y(n-2), \ldots, y(n-N)]^T$

$\mathbf{B} = [b_0, b_2, \ldots, b_{M-1}]^T$

$\mathbf{K} = [k_1, k_2, k_3, \ldots, k_N]^T$

$\mathbf{A}_i(n) = [\alpha_i(n-1), a_i(n-2), \ldots, \alpha_i(n-N)]^T$

$\mathbf{B}_i(n) = [\beta_i(n-1), \beta_i(n-2), \ldots, \beta_i(n-M+1)]^T$

$$\mathbf{T}_2(n) = \begin{pmatrix} 1 & 0 \\ k_1 & 1 \end{pmatrix}$$

Compute filter output

For $m = 3$ to N

$$\mathbf{T}_m(n)_= \begin{pmatrix} \mathbf{T}_{m-1}(n) & \mathbf{0} \\ k_{m-1} & \mathbf{K}_{m-1}\mathbf{T}_{m-1}(n) \end{pmatrix}$$

$\mathbf{X}_N(n) = \mathbf{T}_N\mathbf{Y}_N(n)$

$y(n) = \mathbf{B}^T\mathbf{X}(\mathbf{n}) - \mathbf{K}^T\mathbf{T}_N\mathbf{Y}_N(n)$

Coefficients update

$e(n) = d(n) - y(n)$

For $i = 1$ to N

$\alpha_i(n) = -x_i(n) - \mathbf{K}^T\mathbf{T}_N(n)\mathbf{A}$

$k_i(n+1) = k_i(n) + \mu\, e(n)\alpha_i(n)$

For $j = 1$ to $M-1$

$\beta_j(n) = x(n-j) + \mathbf{K}^T\mathbf{T}_N(n)\mathbf{B}$

$b_j(n+1) = b_j(n) + \mu\, e(n)\beta_j(n)$

$$\mathbf{B}_p = \left[b_0 + p_n^1, b_2 + p_n^2, \ldots, b_{M-1} + p_n^{M-1}\right]^T, \tag{12.93}$$

and compute the perturbed IIR output signal which, from equation (12.84) is given by

$$y_p(n) = B_p(q^{-1})x(n) - A_p(q^{-1})y(n), \tag{12.94}$$

where

$$A_p(q^{-1})x(n) = \mathbf{K}_p^T\mathbf{T}_N(n)\,\mathbf{Y}(n), \tag{12.95}$$

$$B_p(q^{-1})x(n) = \mathbf{B}_p^T\,\mathbf{X}(n), \tag{12.96}$$

$$\mathbf{Y}(n) = [y(n-1), y(n-2), \ldots, y(n-N)]^T, \tag{12.97}$$

$$\mathbf{X}(n) = [x(n), x(n-1), \ldots, x(n-M+1)]^T, \tag{12.98}$$

Table 12.2. Cascade lattice IIR adaptive algorithm using SPSA method.

Define

$\mathbf{C}_n = (r_n^1, r_n^2, \ldots, r_n^N, p_n^0 \ldots p_n^{M-1})^T$

$\mathbf{B} = [b_0, b_2, \ldots, b_{M-1}]^T$

$\mathbf{K} = [k_1, k_2, k_3, \ldots, k_N]^T$

$\mathbf{K}_p = \mathbf{K} + \mathbf{C}$

$\mathbf{B}_p = \mathbf{B} + \mathbf{C}$

$\mathbf{K}_{m-1} = [k_{m-1}, (k_{m-1}k_1, k_{m-1}k_2, \ldots, k_{m-1}k_{m-2}, 1)]^T$

$\mathbf{K}_{m-1}^p = \left[k_{m-1}^p, \left(k_{m-1}^p k_1^p, k_{m-1}^p k_2^p \ldots k_{m-1}^p k_{m-2}^p, 1\right)\right]^T$

$\mathbf{X}(n) = [x(n), x(n-1), x(n-2), \ldots, x(n-M-1)]^T$

$\mathbf{Y}_N(n) = [y(n-1), y(n-2), \ldots, y(n-N)]^T$

$$\mathbf{T}_2(n) = \begin{pmatrix} 1 & 0 \\ k_1 & 1 \end{pmatrix}$$

$$\mathbf{T}_2(n) = \begin{pmatrix} 1 & 0 \\ k_1^p & 1 \end{pmatrix}$$

Compute filter output

For $m = 3$ to N

$$\mathbf{T}_m(n) = \begin{pmatrix} \mathbf{T}_{m-1}(n) & \mathbf{0} \\ k_{m-1} & \mathbf{K}_{m-1}\mathbf{T}_{m-1}(n) \end{pmatrix}$$

$$\mathbf{T}_m^p(n) = \begin{pmatrix} \mathbf{T}_{m-1}^p(n) & \mathbf{0} \\ k_{m-1}^p & \mathbf{K}_{m-1}^p\mathbf{T}_{m-1}^p(n) \end{pmatrix}$$

$y(n) = \mathbf{B}^T\mathbf{X}(\mathbf{n}) - \mathbf{K}^T\mathbf{T}_N\mathbf{Y}_N(n)$

$y_p(n) = \mathbf{B}_p^T\mathbf{X}(\mathbf{n}) - \mathbf{K}_p^T\mathbf{T}_N^p\mathbf{Y}_N(n)$

Coefficients update

$e(n) = d(n) - y(n)$

For $i = 1$ to N

$$k_i(n+1) = k_i(n) + \mu\, e(n)\frac{y(n) - y_p(n)}{r_i}$$

For $j = 1$ to $M-1$

$$b_j(n+1) = b_j(n) + \mu\, e(n)\frac{y(n) - y_p(n)}{p_j}$$

Finally, using the SPSA, the IIR coefficients are updated as follows

$$k_i(n+1) = k_i(n) + \mu\, e(n)\frac{(y(n) - y_p(n))}{r_n^i}, \tag{12.99}$$

$$b_j(n+1) = b_j(n) + \mu\, e(n)\frac{(y(n) - y_p(n))}{p_n^i}. \tag{12.100}$$

If we compare the number of operations required to compute equations (12.87)–(12.88) with the number of operations required to compute equations (12.92)–(12.100), we will

find that the computational complexity of the proposed IIR algorithm using the SPSA is much lower than that of the conventional IIR gradient algorithm.

12.5.3. Steiglitz–McBride type IIR adaptive filter algorithms. Figure 12.7 shows the IIR output error adaptive structure proposed by Steiglitz and McBride [10]. Based on this structure, several IIR adaptive algorithms have been proposed. Among them, some of the most successful algorithms use a cascade of FIR transversal filters and an AR lattice structure to avoid stability problems. Both structures are updated using an LMS-type adaptive algorithm to minimize the mean square error criterion, which is given as

$$\varepsilon(n) = E\left[e^2(n)\right], \tag{12.101}$$

where $e(n)$ is the output error given by [9]

$$e(n) = A(q^{-1}, n)\left(\frac{d(n)}{\widehat{A}(q^{-1}, n-1)}\right) - B(q^{-1}, n)\left(\frac{x(n)}{\widehat{A}(q^{-1}, n-1)}\right), \tag{12.102}$$

where $A(q^{-1}, n)$ and $B(q^{-1}, n)$ are given by equations (12.85) and (12.33), respectively. Assuming that $A(q^{-1}, n-1)$ remain constant, the minimization can be done with respect to the parameters $k_1(n), k_2(n), \ldots, k_N(n), b_0(n), b_1(n), b_2(n), \ldots, b_{M-1}(n)$. Thus, using an LMS type adaptive algorithm, $k_j(n+1)$, $j = 1, 2, \ldots, N$, and $b_m(n+1)$, $m = 1, 2, \ldots, M-1$ are updated as follows:

$$k_j(n+1) = k_j(n) - \mu e(n)\frac{\partial f_N(n)}{\partial k_j}, \tag{12.103}$$

$$b_j(n+1) = b_j(n) + \mu e(n)\frac{\partial y_a(n)}{\partial b_j}, \tag{12.104}$$

where $m = 0, 1, 2, \ldots, M-1$; $j = 1, 2, \ldots, N$ and $e(n)$ is given by equation (12.102).

To update the filter coefficients using equations (12.101) and (12.102), it is necessary to compute $y_a(n)$ and $f_N(n)$. Since these two signals are the result of filtering the input and reference signals $x(n)$ and $d(n)$ with an all-pass lattice filter of order N, as shown in Fig. 12.6, from equation (12.80) it follows that

$$v_0(n) = x(n) - \mathbf{K}^T\mathbf{T}_N(n-1)\mathbf{V}_N, \tag{12.105}$$

$$p_0(n) = d(n) - \mathbf{K}^T\mathbf{T}_N(n-1)\mathbf{P}_N, \tag{12.106}$$

where

$$\mathbf{V}_N(n) = \left[v_1(n), v_2(n), v_3(n), \ldots, v_N(n)\right]^T, \tag{12.107}$$

$$\mathbf{P}_N(n) = \left[p_1(n), p_2(n), p_3(n), \ldots, p_N(n)\right]^T, \tag{12.108}$$

and, $\mathbf{T}_N(n-1)$ and $\mathbf{K}_T$ are given by equations (12.74) and (12.75), respectively, with $m = N$. Subsequently, the signal $v_0(n)$ is filtered by FIR transversal filter structure, whose output signal is given by

$$y_a(n) = \mathbf{B}^T\mathbf{V}(n), \tag{12.109}$$

where

$$\mathbf{V}(n) = \left[v_0(n), v_0(n-1), \ldots, v_0(n-N)\right]^T. \tag{12.110}$$

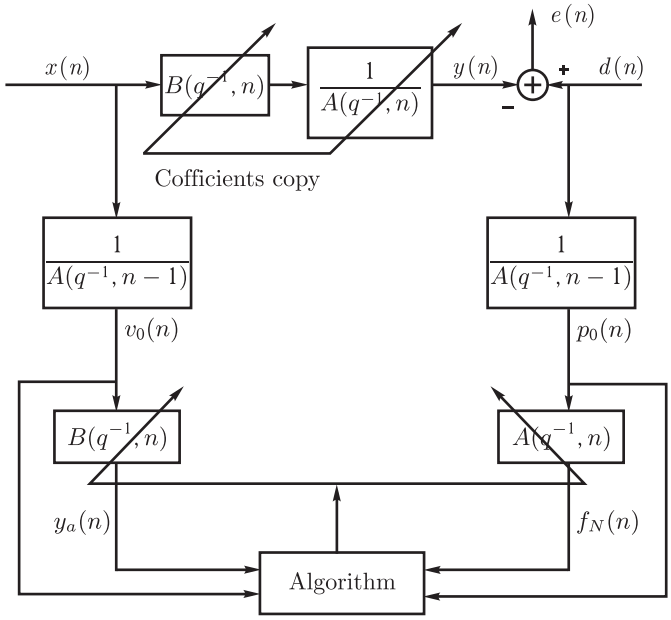


Fig. 12.7. Steiglitz–McBride type IIR adaptive filter structure.

To compute $f_N(n)$, consider the FIR lattice stage shown in Fig. 12.8. Comparing this figure with the all-pole lattice shown in Fig. 12.6 and doing manipulations similar to those described in equations (12.54)–(12.90), we obtain

$$f_N(n) = p_0(n) + \mathbf{K}^T \mathbf{T}_N(n) \mathbf{P}_0, \tag{12.111}$$

where

$$\mathbf{P}_0 = [p_0(n-1), p(n-2) \ldots, p_0(n-M+1)]^T. \tag{12.112}$$

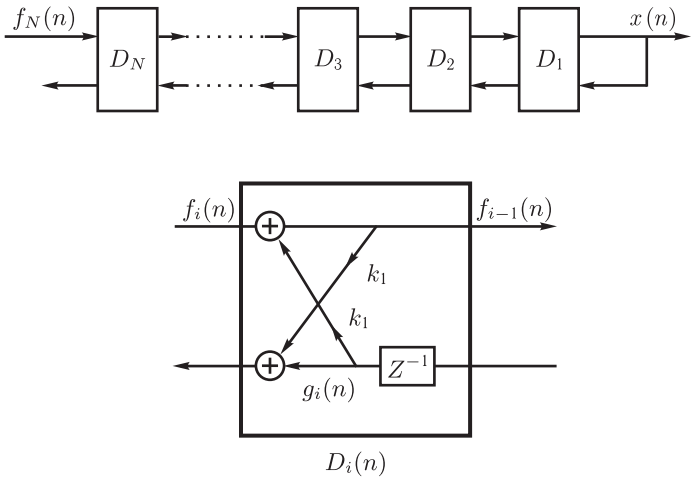


Fig. 12.8. FIR Lattice stage.

Table 12.3. Steiglitz–McBride IIR algorithm.

For each input data do

$\mathbf{B} = [b_0, b_2, \ldots, b_{M-1}]^T$

$\mathbf{K} = [k_1, k_2, k_3, \ldots, k_N]^T$

$\mathbf{K}_{m-1} = [k_{m-1}, (k_{m-1}k_1, k_{m-1}k_2, \ldots, k_{m-1}k_{m-2}, 1)]^T$

$\mathbf{X}(n) = [x(n), x(n-1), x(n-2), \ldots, x(n-M-1)]^T$

$\mathbf{Y}_N(n) = [y(n-1), y(n-2), \ldots, y(n-N)]^T$

$\mathbf{V}_N(n) = [v_1(n), v_2(n), v_3(n), \ldots, v_N(n)]^T$

$\mathbf{P}_N(n) = [p_1(n), p_2(n), p_3(n), \ldots, p_N(n)]^T$

$\mathbf{V}(n) = [v_0(n), v_0(n-1), \ldots, v_0(n-N)]^T$

$\mathbf{P}_0 = [p_0(n-1), p(n-2) \ldots, p_0(n-M+1)]^T$

$\mathrm{A}_i(n) = [\alpha_i(n-1), a_i(n-2), \ldots, \alpha_i(n-N)]^T$

$$\mathbf{T}_2(n) = \begin{pmatrix} 1 & 0 \\ k_1 & 1 \end{pmatrix}, \quad \mathbf{T}_2(n) = \begin{pmatrix} 1 & 0 \\ k_1^p & 1 \end{pmatrix}$$

Compute filter output

For $m = 3$ to N

$$\mathbf{T}_m(n) = \begin{pmatrix} \mathbf{T}_{m-1}(n) & \mathbf{0} \\ k_{m-1} & \mathbf{K}_{m-1}\mathbf{T}_{m-1}(n) \end{pmatrix}$$

$v_0(n) = x(n) - \mathbf{K}^T\mathbf{T}_N(n-1)\mathbf{V}_N$

$p_0(n) = d(n) - \mathbf{K}^T\mathbf{T}_N(n-1)\mathbf{P}_N$

$y_a(n) = \mathbf{B}^T\mathbf{V}(n)$

$f_N(n) = p_0(n) + \mathbf{K}^T\mathbf{T}_N(n)\mathbf{P}_0$

$\mathbf{A}(n) = -\mathbf{T}_\mathbf{N}(n)\mathbf{P}_0(n)$

Coefficients update

$e(n) = f_N(n) - y_a(n)$

$\mathbf{K}(n+1) = \mathbf{K}(n) - 2\mu\mathbf{A}(n)$

$\mathbf{B}(n+1) = \mathbf{B}(n) - 2\mu\mathbf{V}(n)$

Next. using equations (12.58) and (12.59), we can estimate the derivative required to update the filter coefficients, which, since $\mathbf{T}_N(n)$ and $\mathbf{K}$ at the time n do not depend on the parameters at the time $n+1$, are given by [9]

$$\frac{\partial y_a(n)}{\partial b_j} = \beta_j(n) = v_0(n-j), \tag{12.113}$$

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{pmatrix} = -\mathbf{T}_\mathbf{N}(n) \begin{pmatrix} p_0(n-1) \\ p_0(n-2) \\ \vdots \\ p_0(n-N) \end{pmatrix}. \tag{12.114}$$

Thus, finally, the adaptive filter coefficients are updated as follows:

$$k_j(n+1) = k_j(n) - 2\mu e(n)\frac{\partial f_N(n)}{\partial k_j}, \tag{12.115}$$

$$b_j(n+1) = b_j(n) - 2\mu e(n)\beta_j. \tag{12.116}$$

12.5.4. SPSA based Steiglitz–McBride algorithm. The gradient estimation required by equations (12.101) and (12.102), especially the first one that involves the lattice stages, demands a relatively large number of operations. To reduce the computational complexity of the cascade lattice IIR filter while keeping its desirables properties, the IIR coefficients vector can be updated using the simultaneous perturbation stochastic approximation SPSA, in which, firstly, the perturbation vector, $\mathbf{C}_n$, is defined as in equation (12.91). Next, we estimate the unperturbed output error signal, which from equations (12.102), (12.108), and (12.110) is given by

$$e(n) = f_N(n) - y_a(n), \tag{12.117}$$

where $f_N(n)$ and $y_a(n)$ are defined by equations (12.108) and (12.110), respectively. Next define the perturbed vector $\mathbf{K}_p$ and $\mathbf{B}_p$ as follows:

$$\mathbf{K}_p = \left[k_1 + r_n^1, k_2 + r_n^2, k_{3,} + r_n^3, \ldots, k_N + r_n^N\right]^T, \tag{12.118}$$

$$\mathbf{B}_p = \left[b_0 + p_n^1, b_2 + p_n^2, \ldots, b_{M-1} + p_n^{M-1}\right]^T, \tag{12.119}$$

and compute the perturbed output error, which is given by

$$e^p(n) = f_N^p(n) - y_a^p(n), \tag{12.120}$$

where the perturbed signal $y_a^p(n)$ is given by

$$y_a^p(n) = \mathbf{B}_p^T \mathbf{V}(n) \tag{12.121}$$

and $\mathbf{V}(n)$ is given by equation (12.107). Now consider the perturbed signal $f_N^p(n)$, given by

$$f_N^p(n) = p_0(n) + \mathbf{K}_p^T \mathbf{T}_N^p(n)\mathbf{P}_0, \tag{12.122}$$

where $\mathbf{T}_N^p(n)$ is the perturbed matrix, which can be recursively estimated for $m = 3, \ldots$ $\ldots, N$ as follows:

$$\mathbf{T}_m^p(n) = \begin{pmatrix} \mathbf{T}_{m-1}^p(n) & \mathbf{0} \\ k_{m-1} + p_n^{m-1} & \mathbf{K}_p^{m-1}\mathbf{T}_{m-1}^p(n) \end{pmatrix}, \tag{12.123}$$

$$\mathbf{K}_p = \left[k_1 + r_n^1, k_2 + r_n^2, \ldots, k_{m-1} + r_n^{m-1}\right]^T. \tag{12.124}$$

This equation is iteratively estimated with

$$\mathbf{T}_2^p(n) = \begin{pmatrix} 1 & 0 \\ k_1 + p_n^1 & 1 \end{pmatrix}. \tag{12.125}$$

Thus, using equations (12.117) and (12.120), the IIR filter coefficients are updated as follows:

$$k_j(n+1) = k_j(n) - 2\mu e(n)\frac{e(n) - e_p(n)}{r_n^j}, \tag{12.126}$$

$$b_m(n+1) = b_m(n) - 2\mu e(n)\frac{e(n) - e_p(n)}{p_n^m} \tag{12.127}$$

for $j = 1, 2, \ldots N$ and, $m = 0, 1, 2 \ldots, M-1$, respectively.

Table 12.4. SPSA based Steiglitz–McBride algorithm.

For each input data do

$\mathbf{C}_n = (r_n^1, r_n^2, \ldots, r_n^N, p_n^0 \ldots p_n^{M-1})^T$

$\mathbf{B} = [b_0, b_2, \ldots, b_{M-1}]^T$

$\mathbf{K} = [k_1, k_2, k_3, \ldots, k_N]^T$

$\mathbf{K}_{m-1} = [k_{m-1}, (k_{m-1}k_1, k_{m-1}k_2, \ldots, k_{m-1}k_{m-2}, 1)]^T$

$\mathbf{K}_{m-1}^p = \left[k_{m-1}^p, \left(k_{m-1}^p k_1^p, k_{m-1}^p k_2^p \ldots k_{m-1}^p k_{m-2}^p, 1\right)\right]^T$

$\mathbf{X}(n) = [x(n), x(n-1), x(n-2), \ldots, x(n-M-1)]^T$

$\mathbf{Y}_N(n) = [y(n-1), y(n-2), \ldots, y(n-N)]^T$

$\mathbf{V}_N(n) = [v_1(n), v_2(n), v_3(n), \ldots, v_N(n)]^T$

$\mathbf{P}_N(n) = [p_1(n), p_2(n), p_3(n), \ldots, p_N(n)]^T$

$\mathbf{P}_0 = [p_0(n-1), p(n-2) \ldots, p_0(n-M+1)]^T$

$\mathbf{K}_p = \mathbf{K} + \mathbf{C}$

$\mathbf{B}_p = \mathbf{B} + \mathbf{C}$

$$\mathbf{T}_2(n) = \begin{pmatrix} 1 & 0 \\ k_1 & 1 \end{pmatrix}, \quad \mathbf{T}_2(n) = \begin{pmatrix} 1 & 0 \\ k_1^p & 1 \end{pmatrix}$$

Compute filter output

For $m = 3$ to N

$$\mathbf{T}_m(n) = \begin{pmatrix} \mathbf{T}_{m-1}(n) & \mathbf{0} \\ k_{m-1} & \mathbf{K}_{m-1}\mathbf{T}_{m-1}(n) \end{pmatrix}$$

$$\mathbf{T}_m^p(n) = \begin{pmatrix} \mathbf{T}_{m-1}^p(n) & \mathbf{0} \\ k_{m-1}^p & \mathbf{K}_{m-1}^p\mathbf{T}_{m-1}^p(n) \end{pmatrix}$$

Unperturbed output error signal

$v_0(n) = x(n) - \mathbf{K}^T\mathbf{T}_N(n-1)\mathbf{V}_N$

$p_0(n) = d(n) - \mathbf{K}^T\mathbf{T}_N(n-1)\mathbf{P}_N$

$y_a(n) = \mathbf{B}^T\mathbf{V}(n)$

$f_N(n) = p_0(n) + \mathbf{K}^T\mathbf{T}_N(n)\mathbf{P}_0$

$e(n) = f_N(n) - y_a(n)$

Perturbed output error signal

$y_a^p(n) = \mathbf{B}_p^T\mathbf{V}(n)$

$e^p(n) = f_N^p(n) - y_a^p(n)$

Coefficients update, $j = 1$ to N, $m = 0$ to $M-1$

$k_j(n+1) = k_j(n) - 2\mu e(n)\left(e(n) - e_p(n)\right)/r_n^j$

$b_m(n+1) = b_m(n) - 2\mu e(n)\left(e(n) - e_p(n)\right)/p_m^j$

12.6. Simulation Results

The actual performance of the proposed algorithm was evaluated using system identification configuration, in which the unknown system used is the same as reported in [1], whose transfer function is given by

$$H(z) = \frac{-0.097 - 1.337z^{-1} + 1.6z^{-2}}{1.0 - 1.19z^{-1} + 0.7z^{-2}}, \tag{12.128}$$

where the input signal was a white noise sequence. The convergence factor was equal to 0.001, which minimizes the use of the stabilization mechanism that keeps the poles inside the unit circle. The variance of the measurement noise was -10.0 dB. The filter weights are initialized using N uniformly distributed random numbers with zero mean and unit variance, and the perturbation factors used to update the filter weights are uniformly distributed random numbers in the interval [–0.01 +0.01] except [–0.001, 0.001], i.e., $Cmax = 0.01$ and $Cmin = 0.001$.

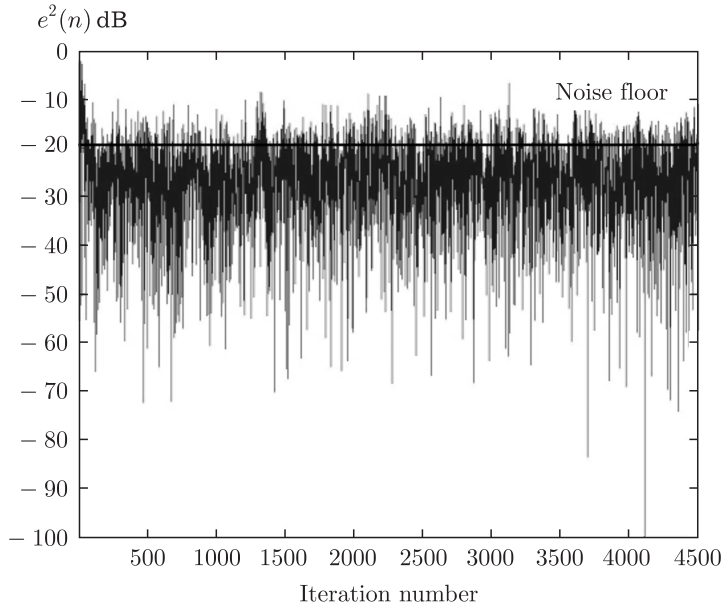


Fig. 12.9. Convergence performance of proposed LMS-IIR-SPSA algorithm.

Figure 12.9 shows the convergence performance of the proposed LMS-based IIR adaptive algorithm using the simultaneous perturbation stochastic approach (SPSA). The convergence performance of the conventional LMS-based IIR adaptive algorithm is shown for comparison in Fig. 12.10.

Figures 12.11 and 12.12 show the convergence performance of the proposed LMS-SPSA-based SM type IIR adaptive algorithm. Fig. 12.13 shows, for comparison, the convergence performance of the previously proposed algorithm LMS based SM IIR adaptive algorithm. Simulation results have shown that the proposed algorithm performs fairly well with a far lower computational complexity than the previously proposed similar structures. Simulation results show that the proposed algorithms using

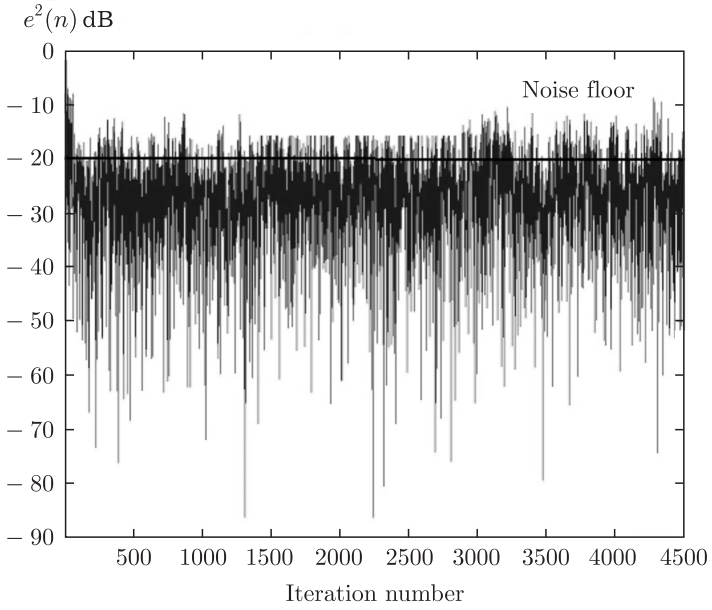


Fig. 12.10. Convergence performance of conventional LMS-IIR-SPSA algorithm.

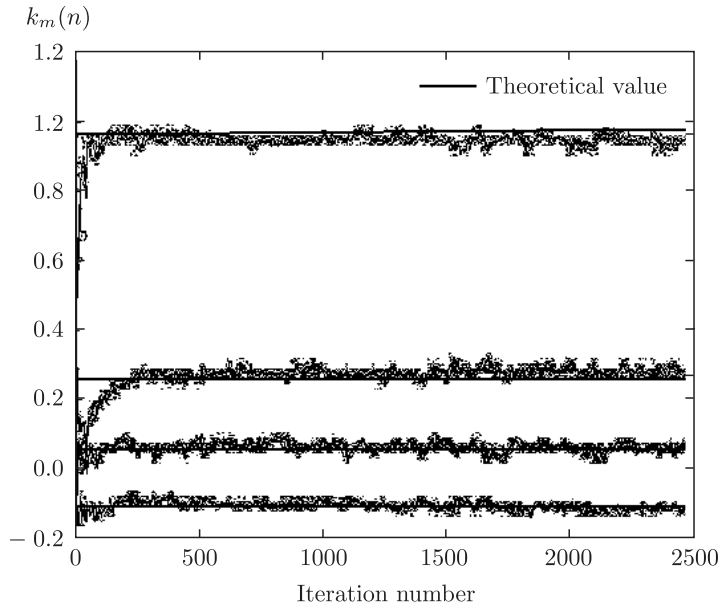


Fig. 12.11. Tap parameters of proposed SM type algorithm.

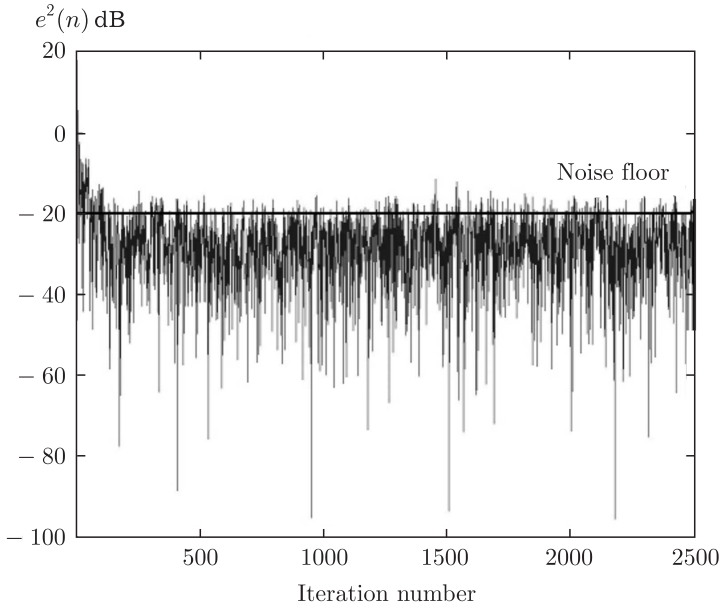


Fig. 12.12. Convergence performance of proposed SM type adaptive algorithm with SPSA.

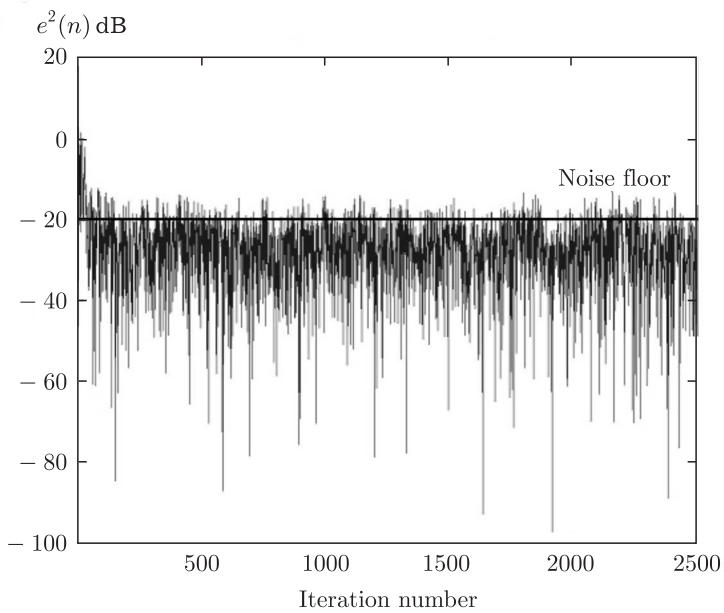


Fig. 12.13. Convergence performance of conventional SM Cascade type IIR adaptive algorithm.

the simultaneous perturbation method provide similar convergence performance as the previously proposed IIR algorithms, with a lower computational complexity.

Conclusions

This chapter presented the development of IIR adaptive filter structures with a trivial stability test. Firstly, an IIR adaptive filter with a parallel structure based on the output error method was presented, in which the input signal is orthogonalized using the DCT. This allows each stage to be independently updated to minimize the total error, using the Gauss–Newton algorithm. Computer simulations have shown that the orthogonalized IIR algorithm provides similar convergence performance with much lower computational complexity.

Two different adaptive algorithms for updating the IIR structure, consisting of a cascade of transversal and lattice stages whose parameters are updated using the SPSA method, are also proposed. The first one uses the IIR-LMS approach and the second one uses the Steiglitz–McBride type IIR adaptive algorithm. The main advantage of the SPSA-based approaches over the previously proposed LMS-IIR and Steiglitz–McBride based IIR filter structures is a significant reduction of the computational complexity. Computer simulations have shown that SPSA-based algorithms provide similar convergence performance as the previously proposed algorithms, but with much lower computational complexity.

References

1. *Widrow*, *B. and Stearns*, *S.* Adaptive signal processing, Prentice Hall, Englewood Cliffs. NJ, 1985.
2. *Treichler*, *J.* Adaptive algorithms for infinite impulse response, Adaptive Filters, Ed. Cowan and P. Grant, Adaptive filters, Prentice Hall, Englewood Cliffs, NJ, 1985.
3. *Shynk*, *J.* Adaptive IIR filtering. IEEE ASSP Magazine, 1989, no. 2, pp. 4–21.
4. *Ljung*, *L. and Soderstrom*, *T.* Theory and practice of recursive identification, MIT Press, Cambridge, 1983.
5. *Friedlander*, *B.* System identification techniques for adaptive noise canceling. IEEE Trans. on Acoustic, Speech and Signal Processing, 1982, vol. 1, no. 1, pp. 699–709.
6. *Héctor Pérez and Shigeo Tsujii*. A System Identification Algorithm using Discrete Orthogonal Functions. IEEE Trans. on Signal Processing, 1991, vol. 39, no. 3, pp. 752–755.
7. *Héctor Pérez and Shigeo Tsujii*. A Fast Parallel IIR Adaptive Filter Algorithm. IEEE Trans. on Signal Processing, Septiembre. 1991, vol. 39, no. 9, pp. 2118–2122.
8. *Burt*, *P. and Gerken*, *M.* A polyphase IIR adaptive filter. Trans. on Circuits and Systems II, 2002, vol. 49, no. 5, pp. 356–359.
9. *Miao*, *K.*, *Fan*, *H.*, *and Doroslovacki*, *M.* Cascade lattice IIR adaptive filters. IEEE Trans. on Signal Processing, 2002, vol. 42, no. 4, pp. 721–742.
10. *Pasquato*, *L. and Kale*, *I.* Adaptive IIR initialization via Hybrid FIR/IIR Adaptive combination. IEEE Trans on Instrumentation and Measurements, 2001, vol. 50, no. 6, pp. 1830–1835.
11. *Regalia*, *P. and Mboup*, *M.* Undermodeled adaptive filtering,: An a priori error bound for the Steiglitz-MacBride Method. IEEE Trans. on Circuit and Systems- II Analog and Digital Signal Processing, 1996, vol. 41, no. 2, pp. 105–116.
12. *Regalia*, *P.* Stable and Efficient lattice algorithms for IIR filtering. IEEE Trans. on Signal Processing, 1992, vol. 40, no. 2, pp. 375–388.
13. *Cousseau*, *J. and Diniz*, *P.* New adaptive IIR filtering algorithms based on the Steiglitz–McBride method. IEEE Transactions on Signal Processing, 1997, vol. 45, no. 5, pp. 1367–1371.
14. *Haykin*, *S.* Adaptive Filter Theory Prentice Hall, Englewood Cliffs NJ, 1991.

15. *Messershmitt, D.* Echo Cancellation in Speech and Data Transmission. IEEE J. Of Selected Areas in Communications, March 1992, vol. SAC-2, no. 2, pp. 283–297.
16. *Nakano-Miyatake, M., Perez-Meana, H., Niño-de-Rivera, L., Casco-Sanchez, F., and Sanchez-Garcia, J.* A Time Varying Step Size Normalized LMS Algorithm for Adaptive Echo Canceler Structures. IEICE Trans. on Fundamentals, Feb. 1995, vol. E-78, no. 2, pp. 254–258.
17. *Amano, F., Perez-Meana, H.A., De Luca, and Duchen G.* A Multirate Acoustic Echo Canceler Structure. IEEE Trans. on Communications, July 1995, vol. 43, no. 7, pp. 2172–2176.
18. *Nakano-Miyatake, M., Perez-Meana, H., Ortiz-Balbuena, L., Niño-de-Rivera, L., and Sanchez J.* A continuous Time RLS adaptive Filter Structure Using Hopfield Neural Networks, Proc. of ISITA'96, Sept. 1996, vol. II, pp. 614–617.
19. *Zelinski, R. and Noll, P.* Adaptive transform coding of speech. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-25, no. 4, pp. 299–309.
20. *Narayan, S., Peterson, A., and Narasimha, J.* Transform domain LMS Algorithm. IEEE Trans. Acoust., Speech, Signal Processing, June 1983, vol. ASSP-31, no. 3, pp. 609–615.
21. *Petraglia, M.R. and Mitra, S.K.* Adaptive FIR Filter Structure Based on the Generalized Subband Decomposition of FIR Filters. IEEE Trans. on Circuits and Systems-II, June 1993, vol. 40, no. 6, pp. 354–362.
22. *Shnnk, J.* Adaptive IIR Filtering Using Parallel Realizations Forms. IEEE Trans. on ASSP, April 1989, vol. 37, no. 4, pp. 517–537.
23. *Perez-Meana, H., Nakano-Miyatake, M., and Nino-de-Rivera, L.* Adaptive Filtering Based on Subband Decomposition, Proc. of The International Workshop of Mathematical Modeling of Physical Processes in Homogeneous Media, pp. 14–16, Guanajuato, Mex, March, 2001.
24. *Maeda, Y. and Kanata, Y.* Learning rules for neuro-controller via simultaneous perturbation. IEEE Trans. on Neural Networks, 1997, vol. 8, no. 5, pp. 1119–1130.
25. *Maeda, Y. and Kanata, Y.* Learning rules for recurrent neural networks using simultaneous perturbation and their application to neuro-control. Trans of The IEICE, 1993, vol. 113-C, no. 4, pp. 402–408.

Chapter 13

ACTIVE NOISE CANCELLING USING THE DISCRETE COSINE TRANSFORM

This chapter presents a low complexity single-channel Active Noise Cancellation (ANC) algorithms with system identification and predictive configurations, which are based on decomposing the filter input signal into a finite number of mutually near orthogonal signal components in which each signal component can be independently processed by a FxRLS adaptive algorithm. Computer simulation results show that the subband decomposition-based ANC structures provide similar convergence performance to conventional ANC structures with FxRLS adaptive algorithms with much lower computational complexity.

13.1. Introduction

Acoustic noise problem becomes more and more important as the use of large industrial equipment, such as engines, blowers, fans, transformers, air conditioners, motors, etc., increases. Due to its importance, several methods have been proposed to solve this problem [1, 2], such as enclosures, barriers, silencers, and other passive techniques that attenuate the undesirable noise [2–4]. There are mainly two types of passive techniques. The first type uses the concept of impedance change caused by a combination of baffles and tubes to silence the undesirable sound. These types of passive techniques, usually called reactive silencers, are commonly used as mufflers in internal combustion engines.

The second type, called resistive silencers, uses energy loss caused by sound propagation in a duct lined with sound-absorbing material [1–8]. These silencers are usually used in ducts for fan noise [1]. Both types of passive silencers have been successfully used for many years in several applications, however the attenuation of passive silencers is low when the acoustic wavelength is large compared with the silencers' dimension [5]. In an effort to overcome these problems, single- and multichannel active noise cancellation (ANC), which uses a secondary noise source that destructively interferes with the unwanted noise, has received considerable attention during the last several years [1], [5]. In addition, because the characteristics of the environment, acoustic noise source, as well as the amplitude, phase, and sound velocity of the undesirable noise are nonstationary, the ANC system must be adaptive in order to cope with these variations [1, 6].

The most commonly used ANC systems are the single channel ANC systems which typically use two microphones. The first microphone is used to measure the noise signal and the second microphone, to measure the attenuated noise or error signal. Both signals are then used to update the ANC parameters so that error power attains a minimum (Fig. 13.1) [1, 6]. In this kind of ANC systems, the adaptive filter $\mathbf{W}(z)$ estimates the time varying unknown acoustic path from the reference microphone to the point where the noise attenuation must be achieved, $P(z)$.

The active noise canceller system is quite similar to the traditional noise canceller system proposed by Widrow and Stearns [9], because in both cases the purpose of the adaptive filter is to minimize the power of the residual error so that the filter output

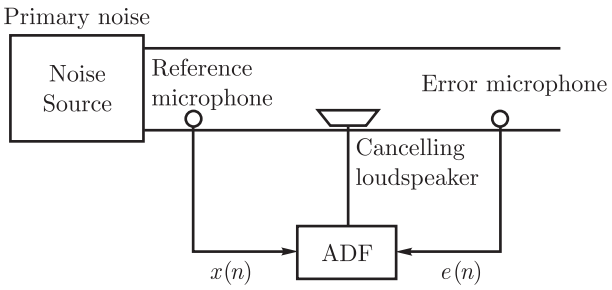


Fig. 13.1. Single-channel broadband feed forward ANC in a duct.

$\widehat{y}(n)$ becomes the best estimate of the disturbance signal d(n) in the mean square sense. However, in the active noise canceller system, an acoustic summing point is used instead of the electrical subtraction of signals. Then, after the primary noise is picked up by the reference microphone, the adaptive filter will require some time to estimate the right output of the canceling loudspeaker. Thus, if the electrical delay becomes longer than the acoustic delay from the reference microphone to the canceling loudspeaker, the system performance will be substantially degraded, because in this situation, the ANC impulse response becomes noncausal. However, when the causality condition is met, the ANC system is able to cancel the broadband random noise [1–6]. Note that, if it is not possible to meet the causality condition, the system can effectively cancel only narrowband or periodic noise.

To avoid this problem, it is necessary to compensate the secondary-path transfer function $S(z)$ from $y(n)$ to $\widehat{y}(n)$ (Fig. 13.2); which includes the digital-to-analog (D/A) converter, the reconstruction filter, the power amplifier, and the loudspeaker; and the acoustic path from the loudspeaker to the error microphone; as well as the error microphone, the preamplifier, the antialiasing filter, and the analog-to-digital (A/D) converter. A key advantage of this approach is that with a proper model of the plant, $P(z)$, and the secondary-path, $S(z)$, the ANC system can respond instantaneously to change in the statistics of the noise signal [1, 6].

Another widely used adaptive structure for active noise cancellation generates internally its own input signal using the adaptive filter output and the error signals, as shown in Figs. 13.3 and 13.4. This approach, if the disturbing noise samples are strongly mutually correlated and the secondary-path S(z) is properly estimated, provides a fairly good cancellation of the disturbing noise [1, 6, 10]. However, the system performance will degrade if the correlation between consecutive samples of noise signal weakens, because in this situation the prediction of the disturbing signal becomes less accurate [1, 6, 10].

In many cases, the ANC structure with a system identification configuration presents better cancellation performance than the ANC using a predictive configuration, because the first one is able to cancel both narrow- and broadband noise. However, in some situations, the signal produced by the canceling speaker is also captured by the reference microphone, which leads to ANC system performance degradation [1, 2]. On the other hand, the ANC with predictive configuration uses only one microphone and therefore it does not present the feedback problem, making it suitable for applications in which the position of error and reference microphone is close to each other and the noise to be cancelled is narrowband.

Most active noise canceller systems use the FxLMS adaptive algorithm or some variation of it, mainly due to its low computational complexity. However, the convergence of the FxLMS is slow when the input signal autocorrelation matrix presents

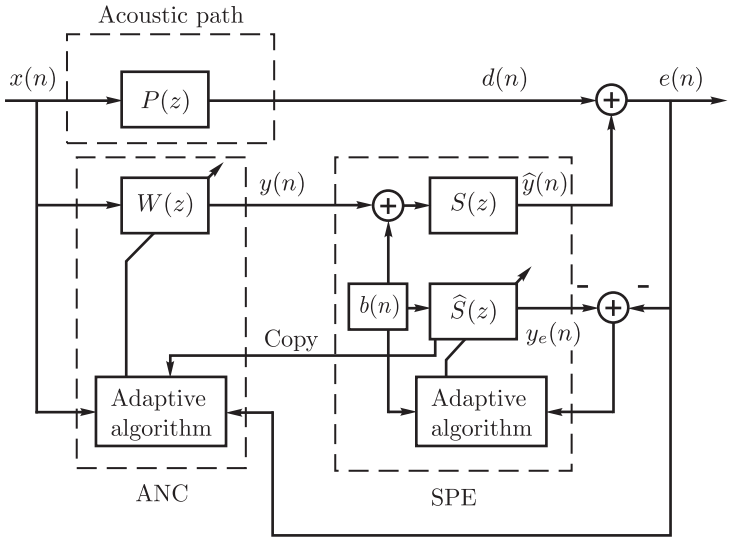


Fig. 13.2. Block diagram of an active noise canceling (ANC) structure with system identification configuration and a secondary path estimation (SPE) stage.

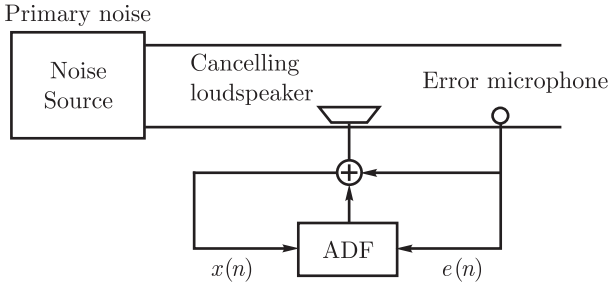


Fig. 13.3. A single channel ANC system in a duct using a predictive structure.

a large eigenvalue spread. In addition, the FxLMS algorithm is sensitive to additive noise. These facts may limit the use of the FxLMS adaptive algorithm when high convergence rates and low sensitivity to additive noise are required [9, 11]. On the other hand, the FxLMS-Newton adaptive algorithms have the potential to provide a much higher convergence rate with a much lower sensitivity to additive noise than the FxLMS algorithm, while its computational complexity is very high. Thus, due to the desirable properties of the FxLMS-Newton algorithm, several efforts have been carried out to reduce the computational complexity of LMS-Newton-based algorithms while keeping its desirables properties.

On the other hand, in real time signal processing, a significant amount of computational effort can be saved if the input signals are represented in terms of a set of orthogonal signal components [12, 13]. That is because the system admits processing schemes in which each signal component can be processed independently. Taking this fact into account, we propose a parallel form active noise cancellation algorithm using a single sensor, with system identification and predictive configurations in which the input signal is split into a set of approximately orthogonal signal components by using the discrete cosine transform. Subsequently, these signal components are fed into a bank

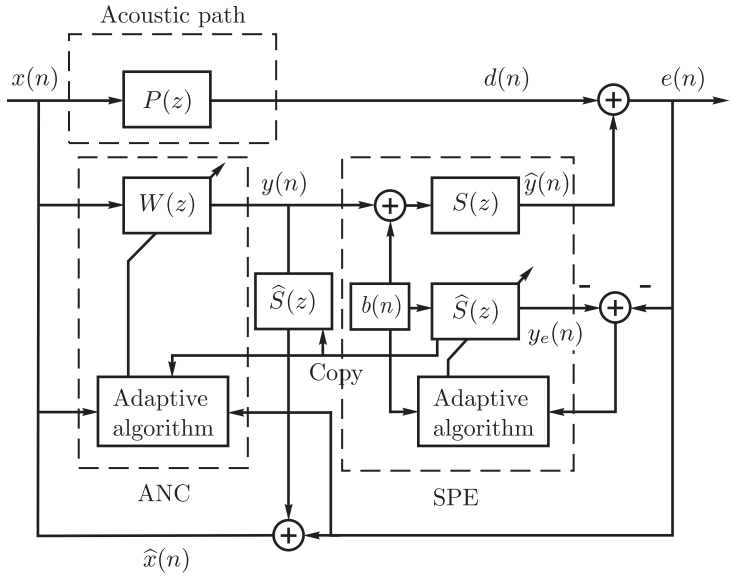


Fig. 13.4. Block diagram of an active noise canceling (ANC) algorithm with a predictive structure and a secondary path estimation (SPE) stage.

of adaptive transversal filters (FIR–ADF) whose parameters are independently updated to minimize the total error [13–15]. The proposed schemes are attractive alternatives to the conventional filtered-x recursive least squares transversal algorithms, FxLMS-Newton, because they reduce the computational complexity of conventional algorithms and keep similar convergence performance.

13.2. ANC Structures Based on Subband Decomposition Approach

Some of the most widely used active noise canceling structures use either the system identification or predictive configurations, shown in Figs. 13.2 and 13.4, which differ from each other only in the way used to derive the input signal. Thus, the filter structure will be developed without assuming any particular configuration.

Consider the output signal $y(n)$ of an Nth-order transversal filter, given by

$$y(n) = \mathbf{X}_F^T(n)\mathbf{H}_F, \tag{13.1}$$

where

$$\mathbf{X}_F(n) = \left[\mathbf{X}^{\mathrm{T}}(\mathrm{n}), \mathbf{X}^{\mathrm{T}}(\mathrm{n-M}), \mathbf{X}^{\mathrm{T}}(\mathrm{n-2M}), \ldots, \right.$$
$$\left. \ldots, \mathbf{X}^T(n-(L-2)M), \mathbf{X}^T(n-(L-1)M)\right]^T, \tag{13.2}$$

$$\mathbf{X}(n-kM) = [x(n-kM), x(n-kM-1), \ldots,$$
$$\ldots, x(n-(k+1)M+2), x(n-(k+1)M+1)]^T, \tag{13.3}$$

is the input vector, and

$$\mathbf{H}_F = \left[\mathbf{H}_0^{\mathrm{T}}, \mathbf{H}_1^{\mathrm{T}}, \mathbf{H}_2^{\mathrm{T}}, \ldots, \mathbf{H}_{\mathrm{L-1}}^{\mathrm{T}}\right]^T, \tag{13.4}$$

$$\mathbf{H}_k = \left[h_{kM}, h_{kM+1}, h_{kM+2}, \ldots, h_{(k+1)M-1}\right]^T \tag{13.5}$$

is the adaptive filter coefficients vector. Substituting equations (13.2) and (13.4) into equation (13.1), we obtain

$$y(n) = \sum_{k=0}^{L-1} \mathbf{X}^T(n - kL)\mathbf{H}_k. \tag{13.6}$$

Next, defining

$$\mathbf{H}_k = \mathbf{C}^T\mathbf{A}_k, \tag{13.7}$$

where $\mathbf{C}$ denotes an orthogonal transformation, such as the DFT, DCT, etc., and substituting equation (13.7) into equation (13.6), we obtain

$$y(n) = \sum_{k=0}^{L-1} \left(\mathbf{C}\mathrm{X}(\mathrm{n} - \mathrm{kM})\right)^T \mathbf{A}_k = \sum_{k=0}^{L-1} \mathbf{U}^T(n - kM)\mathbf{A}_k, \tag{13.8}$$

where $U^T(n - kM) = (CX(n - kM))^T$ and

$$\mathbf{U}(n - kM) = [u_0(n - kM), u_1(n - kM), u_2(n - kM), u_3(n - kM), \ldots, u_{M-1}(n - kM)]^T, \tag{13.9}$$

$$\mathbf{A}_k = \left[a_{k,1}, a_{k,2,}, \ldots, a_{k,(M-1)}\right]^T. \tag{13.10}$$

Then, from equations (13.9) and (13.10), $y(n)$ can be represented as

$$y(n) = \sum_{k=0}^{L-1} \sum_{r=0}^{M-1} a_{k,r} u_r(n - kM). \tag{13.11}$$

When $\mathbf{U}(n - kM)$ denotes the discrete Fourier transform (DFT) of the input signal, equation (13.11) defines the output signal of the short delay fast least mean square, SDFLMS, adaptive filter structure, proposed in [16]. This approach, which is a generalization of the conventional FLMS adaptive filter algorithm [16], reduces the processing delay and increases the convergence rate of conventional FLMS, providing at the same time perfect reconstruction properties. This structure performs fairly well using block processing with gradient search based algorithms. However, when RLS type algorithms are required to increase the convergence rate, the computational complexity can be very high, even if the coefficients of the input signal transformation be uncorrelated among them.

To reduce the computational complexity of the proposed structure when a RLS type adaptation algorithm is used, firstly interchange the summation order as follows:

$$y(n) = \sum_{r=0}^{M-1} \sum_{k=0}^{L-1} a_{k,r} u_r(n - kM) \tag{13.12}$$

and define

$$\mathbf{V}_r(n) = [u_r(n), u_r(n - M), u_r(n - 2M), u_r(n - 3M), \ldots, u_r(n - (L - 2)M), u_r(n - (L - 1)M)]^T, \tag{13.13}$$

$$\mathbf{W}_r = \left[a_{0,r}, a_{1,r}, a_{2,r}, \ldots, a_{(L-1),r}\right]^T, \tag{13.14}$$

so that equation (13.12) takes the form

$$y(n) = \sum_{r=0}^{M-1} \mathbf{W}_r^T \mathbf{V}_r(n). \tag{13.15}$$

Equation (13.15) denotes the output signal of the subband decomposition based filter structure proposed in [11], which also has perfect reconstruction properties without regarding the statistics of the input signal or the adaptive filter order. Figure 13.5 shows that the realizations forms given by equations (13.11) and (13.15) are equivalent.

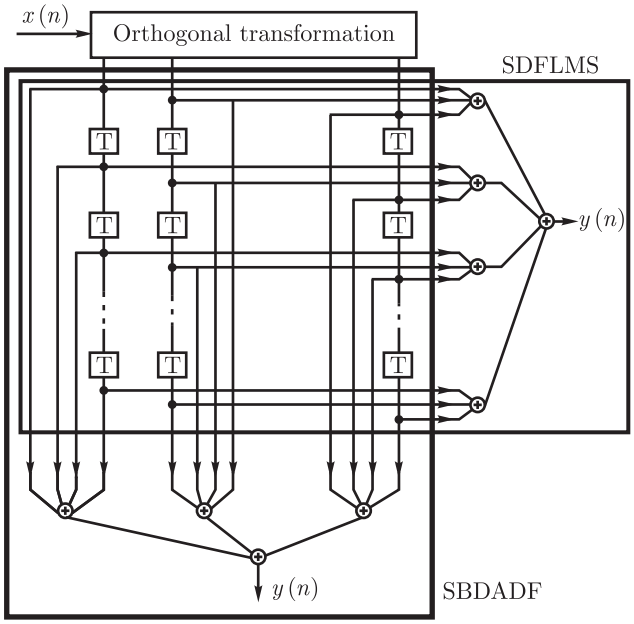


Fig. 13.5. Equivalence between the realization form of SDFLMS [15] and the subband decomposition based ADF [11].

13.2.1. Adaptation algorithm. Consider the output error which, from equations (13.15) and Figs. 13.2 and 13.4, is given by

$$e(n) = d(n) - \left(\sum_{r=0}^{M-1} \mathbf{W}_r^T \mathbf{V}_r(n) \right) * s(n), \tag{13.16}$$

$$e(n) = d(n) - \left(\sum_{r=0}^{M-1} \mathbf{W}_r^T \widehat{\mathbf{V}}_r(n) \right), \tag{13.17}$$

where $\mathbf{W}_r$ is given by equation (13.14), and

$$\widehat{\mathbf{V}}_r(n) = [\widehat{v}_r(n), \widehat{v}_r(n-M), \ldots, \widehat{v}_r(n-(L-1)M)]^T, \tag{13.18}$$

$$\widehat{v}_r(n) = s_r(n) * u_r(n), \tag{13.19}$$

where $u_r(n)$ is the orthogonal transformation of the input signal and $*$ denotes the convolution.

The performance of the proposed ANC structure strongly depends on the choice of the orthogonal transformation, because in the development of adaptive algorithm it is assumed that the transformation components are fully uncorrelated. Several orthogonal transformations that approximately satisfy this requirement could be used, such as the discrete cosine transform (DCT), the discrete Fourier transform (DFT), the discrete sine transform (DST), the Walsh–Hadamard transform, etc. Among them, the DCT appears to be an attractive alternative because it is a real transformation and has better orthogonalizing properties than other orthogonal transformations. Besides, it can be estimated in a recursive form by using a filter bank whose rth output signal is given by [12, 14, 20]

$$u_r(n) = 2\cos\left(\frac{\pi r}{M}\right) u_r(n-1) - u_r(n-2) - \\ - \cos\left(\frac{\pi r}{2M}\right)\{x(n-M-1) - (-1)^r x(n-1) \\ -x(n-M) + (-1)^r x(n)\}. \quad (13.20)$$

To achieve high convergence rates, the coefficients vector $\mathbf{W}_r$, $r = 0, 1, 2, \ldots, M-1$, will be estimated so that the sum of squared errors, $\varepsilon(n)$, given as

$$\varepsilon(n) = \sum_{k=1}^{n}\left(d(k) - \sum_{r=0}^{M-1}\mathbf{W}_r^T\widehat{\mathbf{V}}_r(k)\right)^2, \quad (13.21)$$

$$\varepsilon(n) = \sum_{k=1}^{n}\left(d(k) - \mathbf{W}^T\widehat{\mathbf{V}}(k)\right)^2, \quad (13.22)$$

attains a minimum, where

$$\mathbf{W} = \left[\mathbf{W}_1^T, \mathbf{W}_2^T, \mathbf{W}_3^T, \ldots, \mathbf{W}_{L-1}^T\right]^T, \quad (13.23)$$

$$\widehat{\mathbf{V}}(n) = \left[\mathbf{V}_0^T(n), \widehat{\mathbf{V}}_1^T(n), \widehat{\mathbf{V}}_2^T(n), \ldots, \widehat{\mathbf{V}}_{M-1}^T\right]^T, \quad (13.24)$$

and vectors $\mathbf{V}_r$ and $\mathbf{W}_r(n)$ are given by equations. (13.13) and (13.14), respectively.

Multiplying equation (13.22) on the left by $\widehat{\mathbf{V}}(n)$ and using the orthogonality property of the least square estimation, we obtain

$$\left[\sum_{k=1}^{n}\widehat{\mathbf{V}}(k)\widehat{\mathbf{V}}^T(k)\right]\mathbf{W}(n) = \sum_{k=1}^{n}d(k)\widehat{\mathbf{V}}(k). \quad (13.25)$$

Next, assuming that the DCT coefficients of the input signal are fully mutually uncorrelated [14, 15, 18], we can write equation (13.25) as

$$\left[\sum_{k=1}^{n}\widehat{\mathbf{V}}_r(k)\widehat{\mathbf{V}}_r^T(k)\right]\mathbf{W}_r(n) = \sum_{k=1}^{n}d(k)\widehat{\mathbf{V}}_r(k), \quad (13.26)$$

$$\mathbf{W}_r(n) = \left[\sum_{k=1}^{n}\widehat{\mathbf{V}}_r(k)\widehat{\mathbf{V}}_r^T(k)\right]^{-1}\sum_{k=1}^{n}d(k)\widehat{\mathbf{V}}_r(k), \quad (13.27)$$

where $r = 0, 1, 2, 3, \ldots, M-1$. Equation (13.27) is the solution of the Wiener–Hopf equation, which can be solved recursively by using the Matrix Inversion Lemma as follows:

$$\mathbf{W}_r(n+1) = \mathbf{W}_r(n) + \mu\,\mathbf{K}_r(n)e(n); \quad (13.28)$$

where $e(n)$ is the output error given by equation (13.17); μ is the convergence factor, which controls the stability and convergence rate [16, 21];

$$\mathbf{K}_r(n) = \frac{\mathbf{P}_r(n)\widehat{\mathbf{V}}_r(n)}{\lambda + \widehat{\mathbf{V}}_r^T(n)\mathbf{P}_r(n)\widehat{\mathbf{V}}_r(n)}; \tag{13.29}$$

$$\mathbf{P}_r(n+1) = \frac{1}{\lambda}\left[\mathbf{P}_{\mathrm{r}}(\mathrm{n}) - \mathbf{K}_{\mathrm{r}}(\mathrm{n})\widehat{\mathbf{V}}_{\mathrm{r}}^{\mathrm{T}}(\mathrm{n})\mathbf{P}_{\mathrm{r}}(\mathrm{n})\right]; \tag{13.30}$$

and $\widehat{\mathbf{V}}_r(n)$ is given by equation (13.19). Taking into account [16] that

$$\mathbf{K}_r(n) = \mathbf{P}_r(n)\widehat{\mathbf{V}}_r(n), \tag{13.31}$$

we can write equation (13.29) in the form

$$\mathbf{W}_r(n) = \mathbf{W}_r(n-1) + \mu\,\mathbf{P}_r(n+1)e(n)\widehat{\mathbf{V}}_r(n). \tag{13.32}$$

Equation (13.32), when $\mu < 1$, is the so-called LMS-Newton algorithm, which converges to the optimal solution when $0 < \mu < 1$. A detailed analysis of the LMS-Newton algorithm is given in [21]. The proposed parallel form ANC system is shown in Figs. 13.6 and 13.7.

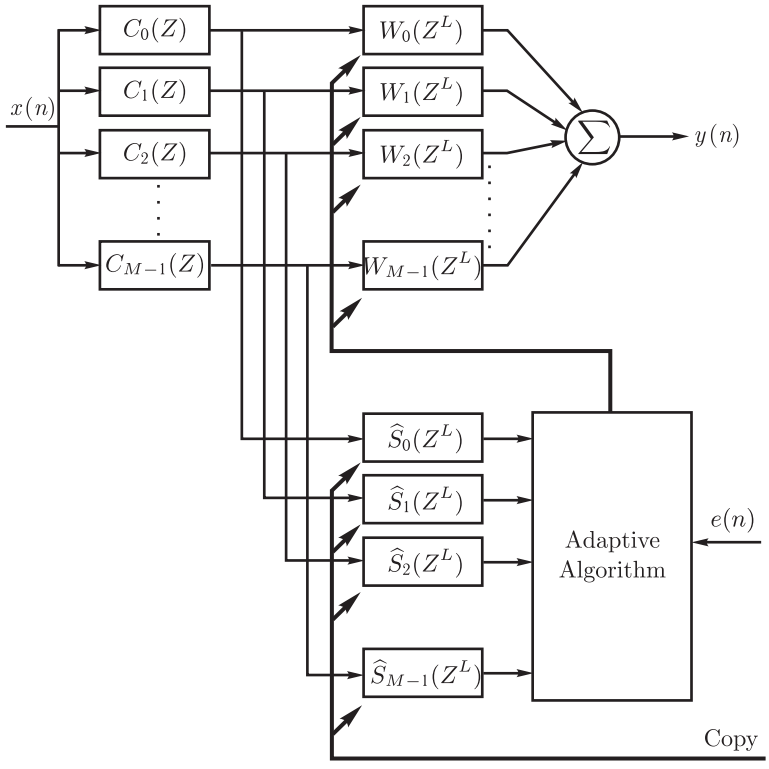


Fig. 13.6. Proposed active noise canceller structure using subband decomposition method.

The proposed adaptive structure can be used in active noise canceller structures using either system identification or predictive configurations. In the first case, the input signal is picked up by the reference microphone, while, when a predictive configuration

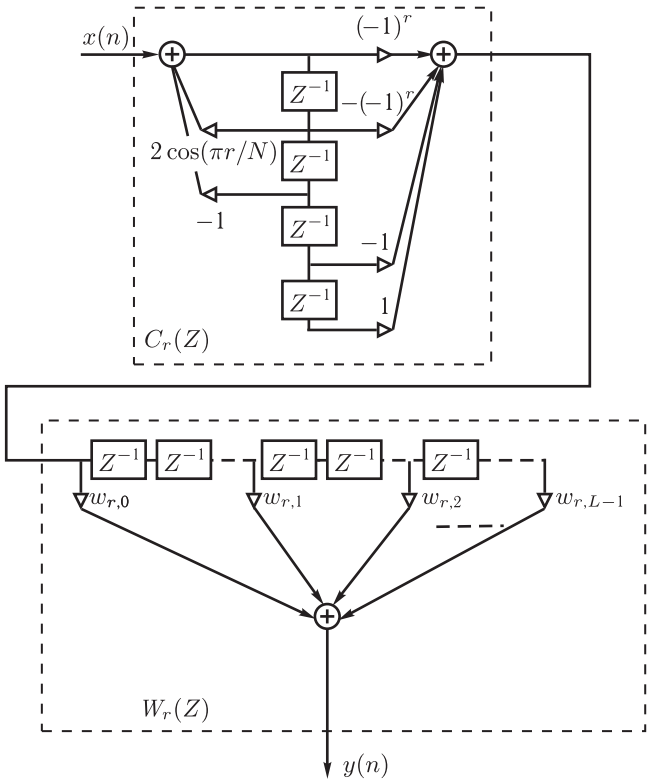


Fig. 13.7. rth stage of proposed ANC structure.

is used, the input signal is estimated from the output error and the adaptive filter output signal.

13.2.2. Secondary path estimation. A widely used secondary path estimation method uses an adaptive filter in parallel with the secondary path feed with an internally generated white noise sequence, which is also added to the noise canceller output signal $y(n)$ to drive the secondary path and generate in this way the reference signal, such that the output error is given by

$$f(n) = s(n) * b(n) + s(n) * y(n) - d(n) - y_e(n), \tag{13.33}$$

where

$$y_e(n) = \sum_{r=0}^{L-1} \widehat{\mathbf{S}}_r^T \mathbf{B}_r(n) \tag{13.34}$$

and $*$ denotes the convolution. Next, assuming that $v(n)$ is uncorrelated with $y(n)$, the term

$$u(n) = s(n) * y(n) - d(n) \tag{13.35}$$

in equation (13.33) denotes the additive noise, where s(n)*b(n) is the reference signal. Then, because the adaptive filter is operating in a system identification configuration, the adaptive filter coefficients vector will converge to the optimum solution even in the

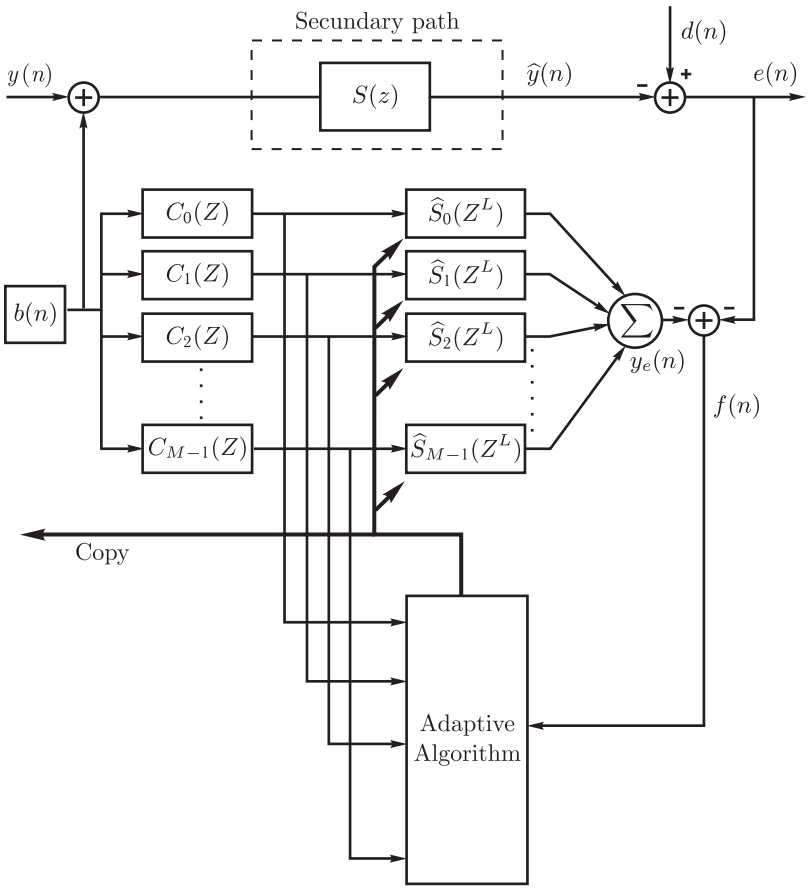


Fig. 13.8. Secondary path estimation using an internally generated white noise sequence $b(n)$.

presence of $u(n)$, when the LMS-Newton algorithm is used [21]:

$$\widehat{\mathbf{S}}_r(n) = \widehat{\mathbf{S}}_r(n-1) + \mu\, \mathbf{G}_r(n) f(n), \tag{13.36}$$

where $f(n)$ is the output error given by equation (13.33),

$$\mathbf{G}_r(n) = \frac{\mathbf{Q}_r(n-1)\mathbf{B}_r(n)}{\lambda + \mathbf{B}_r^T(n)\mathbf{Q}_r(n-1)\mathbf{B}_r(n)}, \tag{13.37}$$

$$\mathbf{Q}_r(n) = \frac{1}{\lambda}\left[\mathbf{Q}_{\mathrm{r}}(\mathrm{n}-1) - \mathbf{G}_{\mathrm{r}}(\mathrm{n})\mathbf{B}_{\mathrm{r}}^{\mathrm{T}}(\mathrm{n})\mathbf{Q}_{\mathrm{r}}(\mathrm{n}-1)\right], \tag{13.38}$$

$$\mathbf{B}_r(n) = \left[b_r(n), b_r(n-L), \ldots, b_r(n-(L-1)M)\right]^T, \tag{13.39}$$

$$b_r(n) = 2\cos\left(\frac{\pi r}{M}\right) b_r(n-1,r) - b_r(n-2,r) + \\ + \left[x(n) - x(n-1) - (-1)^r x(n-M) + x(n-M-1)\right], \tag{13.40}$$

N is the filter order, M is the number of subfilters, and L is the number of subfilters' coefficients. The convergence properties of equation (13.36) are the same as those of equations (13.28) and (13.32).

When an offline estimation is used, $d(n)$ and $y(n)$ are equal to zero and, therefore, the adaptive filter operates in a system identification configuration in near ideal conditions. Then $\mathbf{S}_r$ will converge to the optimal solution of the Wiener–Hopf equation.

13.2.3. Computational complexity. The proposed structure requires $2LM$ multiplication and $5LM$ additions for the DCT estimation, and LM multiplication and $LM + M$ additions to compute the filter output. Next, for adaptation, each stage requires L_s multiplications and L_s additions to estimate the secondary-path output signals, where L_s is the r-th secondary-path stage order. $L^2 + L$ multiplications and $L^2 + L + 1$ additions are required for the Kalman gain estimation. The estimation of the inverse autocorrelation matrix requires $3L^2$ multiplication and $2L^2$ additions. Finally, for updating vectors coefficients L multiplication and L additions are required. Thus, because the proposed structure consists of M stages for filtering and update, it requires $7L^2M + 12LM + (L_s + 2)M$ floating-point operations. This, computational complexity is far lower than the $8(LM)^2 + 8(LM) + 1$ floating-point operations required by the conventional FIR structure. Figure 13.9 shows a comparison of the number of floating point operations required by the proposed and conventional FxLMS-Newton algorithms, respectively. From this figure it is clear that the proposed algorithms require far fewer operations per sample period than the conventional algorithm, especially for a large filter order. In all cases L was fixed to 4.

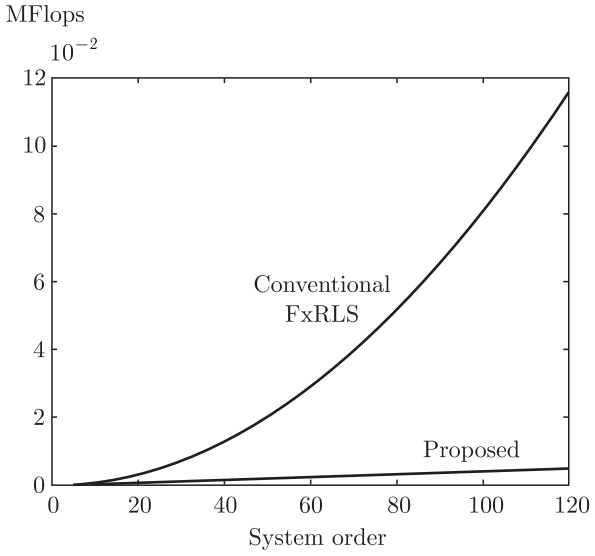


Fig. 13.9. Computational complexity of proposed ANC.

13.3. Computer Simulations

The cancellation performance of the proposed ANC algorithms was evaluated by computer simulations in which the proposed and conventional algorithms with both system identification and predictive configurations were required to cancel the actual airplane, bell, motor, and bike noise signals, whose correlation sequences, estimated from the input data, are shown in Figs. 13.10–13.12. In all cases, the secondary path impulse response $s(n)$ was estimated off-line using a subband decomposition based adaptive filter structure using the LMS-Newton algorithm described in Section 13.2.2.

In all cases both the noise, $P(z)$, and secondary, $S(z)$, paths are shown in Figs. 13.2 and 13.3, which were simulated using FIR filters of order 20 with impulse responses given by

$$p(n) = \exp(-kn)r(n), \quad n = 0, 1, 2, \ldots, N, \tag{13.41}$$

where N is the filter order, n is the time index, k is a constant such that $\exp(-kN) = 0.01$, and $r(n)$ is a uniformly distributed random sequence with a zero mean and a unit variance.

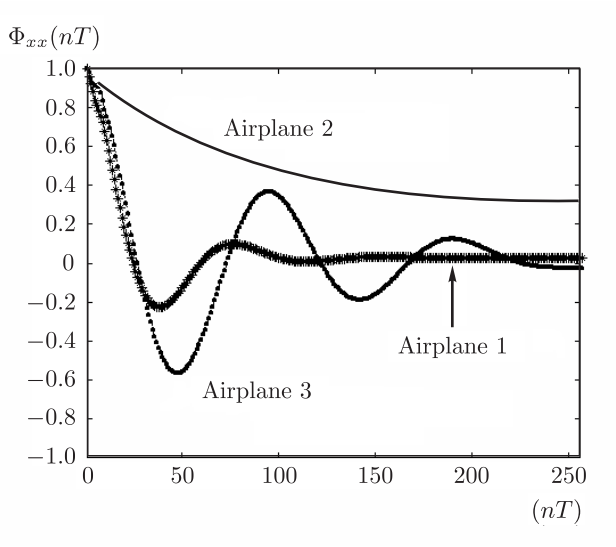


Fig. 13.10. Autocorrelation sequences of airplane signals.

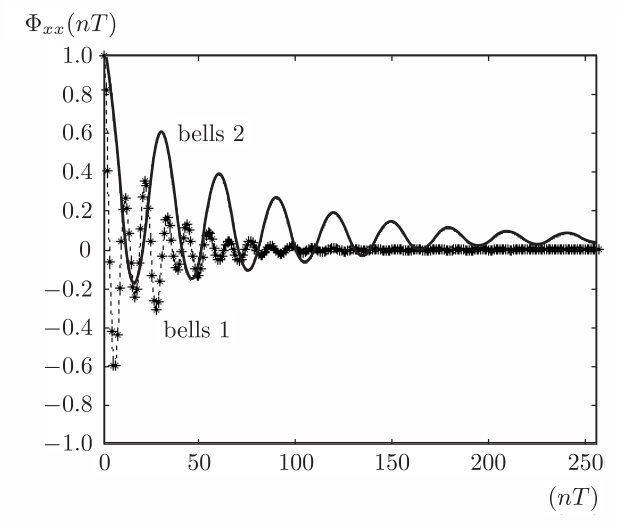


Fig. 13.11. Autocorrelation sequences of bells signals.

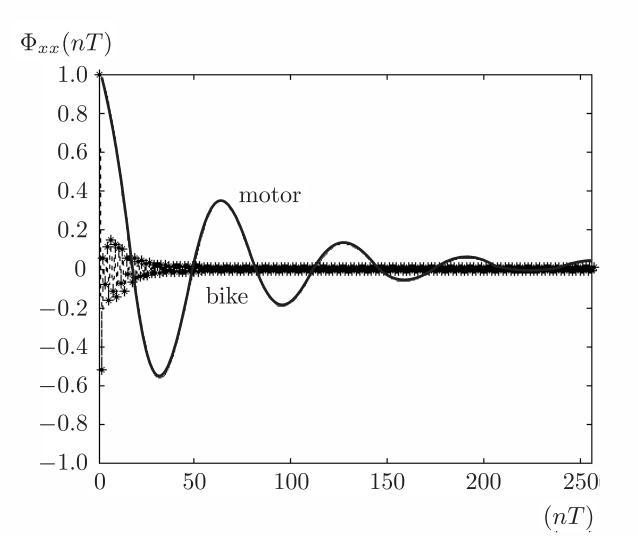


Fig. 13.12. Autocorrelation sequences of a motor and a bike signals.

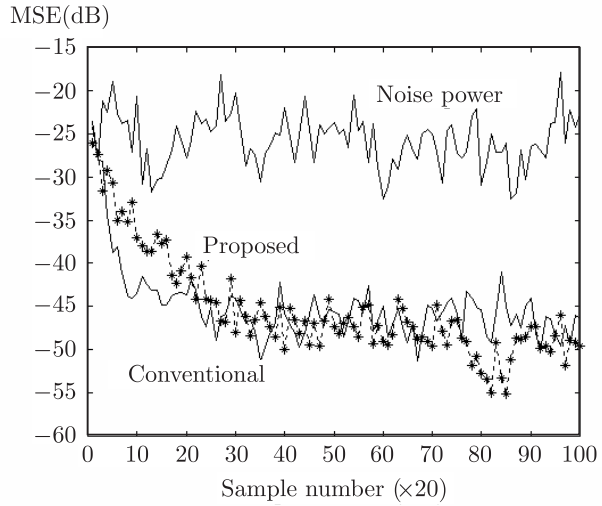


Fig. 13.13. Convergence performance of proposed and conventional algorithm when they are required to cancel an actual airplane noise signal.

13.3.1. ANC algorithm with system identification configuration. Figures 13.13–13.15 show the cancellation performance of the proposed and conventional ANC algorithms, operating with a system identification configuration, when required to cancel three actual noises produced by an airplane, a bell, and a motor, respectively. In all cases, the sparse filters order was equal to 4 and the overall filter order equal to 20. Figures 13.13–13.15 show that the proposed scheme provides quite similar performance to conventional ANC algorithm with much less computational complexity. In all cases, the forgetting factor λ is equal to 0.99 and the convergence factor μ is equal to 0.1.

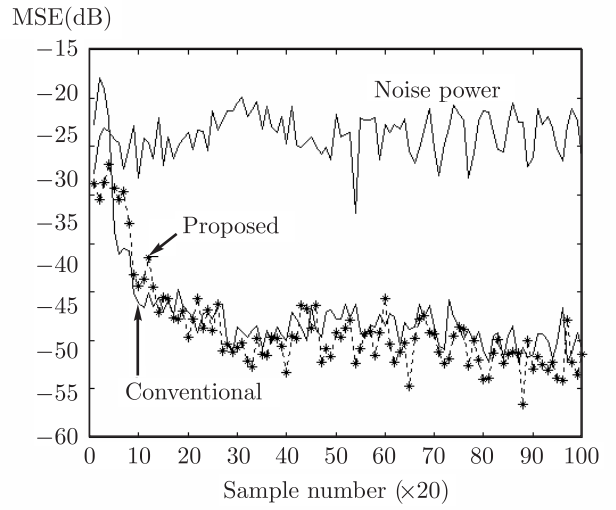


Fig. 13.14. Convergence performance of proposed and conventional algorithms when they are required to cancel an actual bell noise signal.

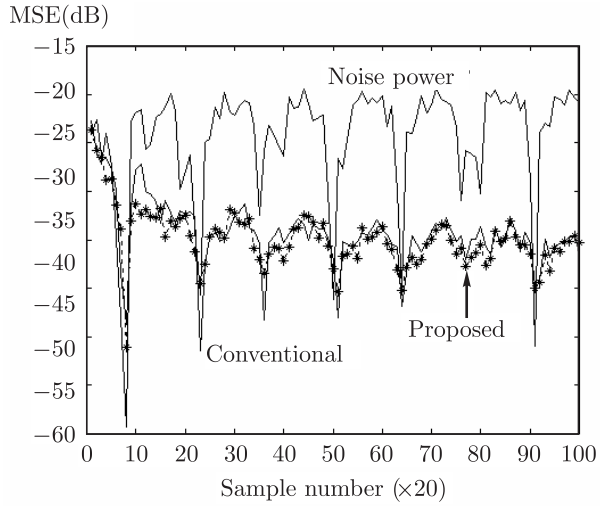


Fig. 13.15. Convergence performance of proposed and conventional algorithm when they are required to cancel an actual motor noise signal.

13.3.2. ANC algorithms with predictive configuration. Figures 13.16–13.18 show the cancellation performance of the proposed ANC with predictive configuration when it is required to cancel actual airplane noise signals whose autocorrelation sequences are shown in Fig. 13.10. These figures show that a fairly good cancellation performance is achieved because, as shown in Fig. 13.10, the airplane signals present a strong correlation between their samples.

Figures 13.19 and 13.20 show the convergence performance of the proposed ANC structure when it is required to cancel two actual bell signals whose correlation

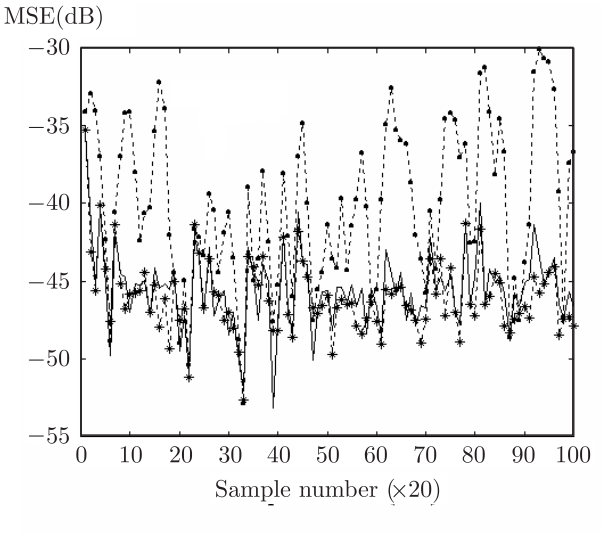


Fig. 13.16. Convergence performance of proposed, __, and conventional, − − ∗ − −, ANC algorithms with a predictive configuration. The noise signal was the airplane noise signal, airplane_1. The time variations of noise power (− − • − −) is shown for comparison.

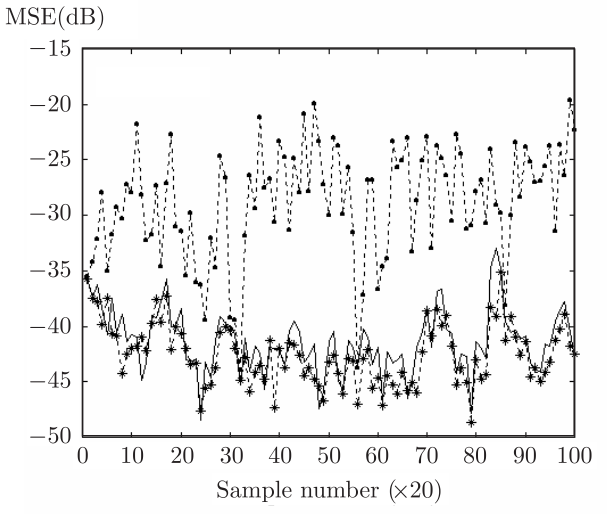


Fig. 13.17. Convergence performance of proposed, _____, and conventional − − ∗ − −algorithms with a predictive configuration. The noise signal was the airplane noise signal, airplane_2. The time variations of noise power, − − • − −, is shown for comparison.

sequences are shown in Fig. 13.11. These figures show that a fairly good cancellation is achieved, because the bell signals present a strong correlation among their samples, enabling an accurate prediction.

Finally, Figs. 13.21 and 13.22 show the performance of the proposed algorithm when it is required to cancel actual motor and bike noise signals, whose autocorrelation

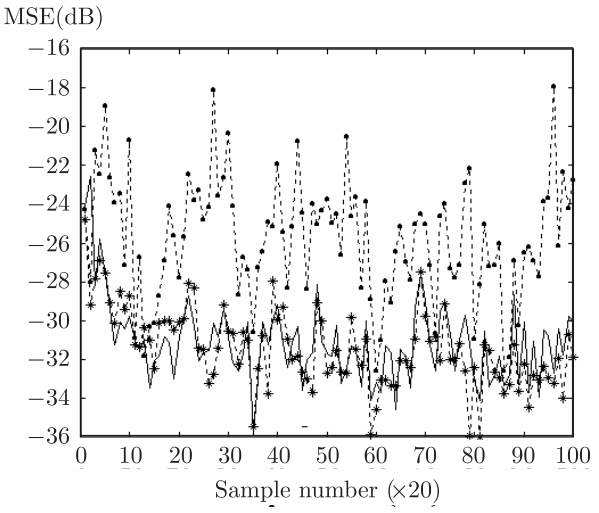


Fig. 13.18. Convergence performance of proposed, _____, and conventional − − ∗ − −ANC algorithms with a predictive configuration. The noise signal was the airplane noise, airplane_3. The time variations of noise power, − − • − −, is shown for comparison.

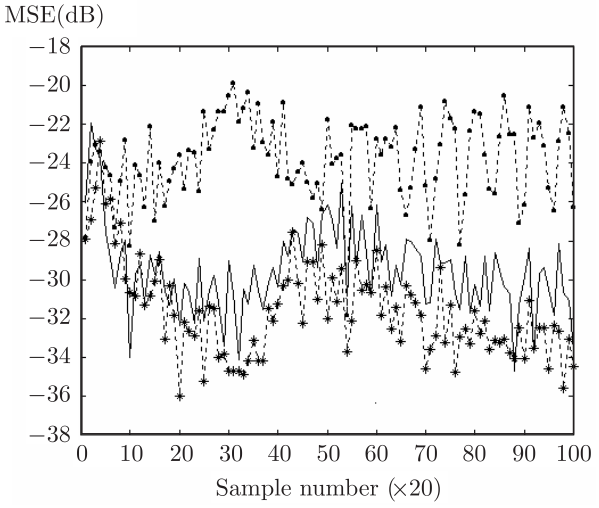


Fig. 13.19. Convergence performance of proposed, _____, and conventional, − − ∗ − −, algorithms with a predictive configuration. The noise signal is a bell noise signal, bell_1. The time variations of noise power, − − • − −, is shown for comparison.

sequences are shown in Fig. 13.12, respectively. These figures show that a fairly good cancellation is achieved in the case of a motor signal, because its samples are strongly mutually correlated. However, in the case of a bike signal, the cancellation achieved is poor, since, in this case, the noise signal presents a week correlation between its samples. In all cases, the convergence performance of the conventional FxLMS-Newton algorithm is also shown for comparison.

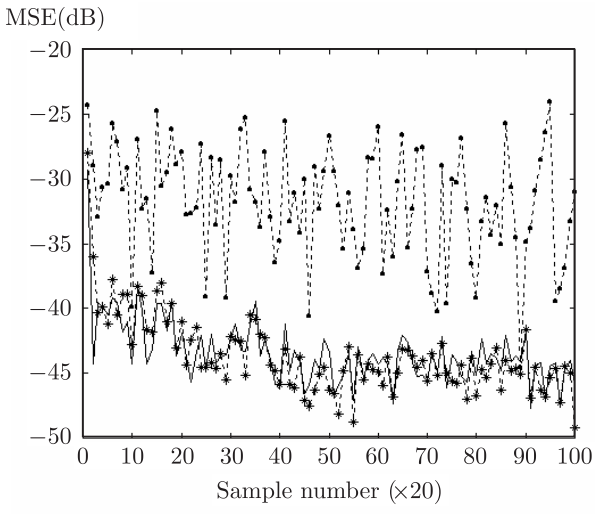


Fig. 13.20. Convergence performance of proposed, _____, and conventional, − − ∗ − −, algorithms with a predictive configuration. The noise signal is a bell noise signal, bell_2. The time variations of noise power, − − • − −, is shown for comparison.

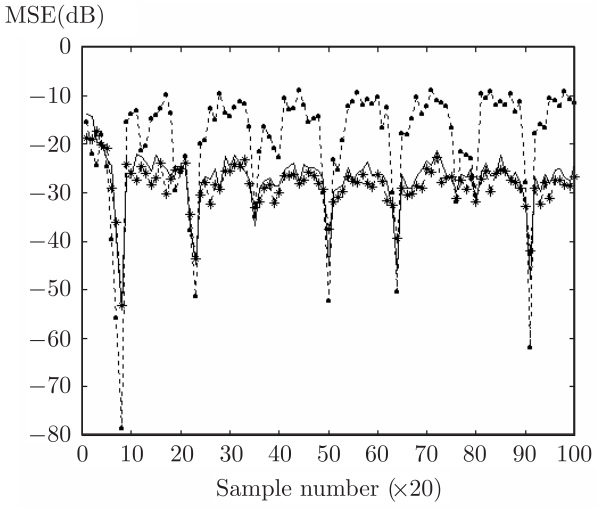


Fig. 13.21. Convergence performance of proposed, _____, and conventional, − − ∗ − −,algorithms with a predictive configuration. The noise signal is an actual motor noise signal. The time variations of noise power, − − • − −, is shown for comparison.

Conclusions

This chapter has proposed active noise cancellation algorithms based on a subband decomposition approach, with system identification and predictive configurations in which the input signals are split into M near orthogonal signal components using the discrete cosine transform. Subsequently, a sparse FIR adaptive filter is inserted in

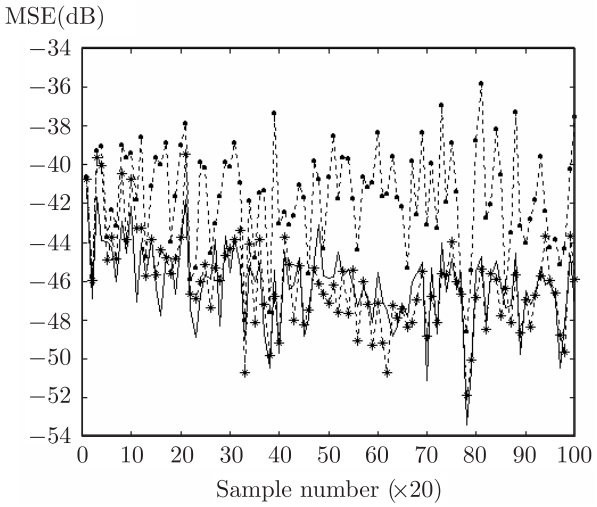


Fig. 13.22. Convergence performance of proposed, _____, and conventional, − − ∗ − −, algorithms with a predictive configuration. The noise signal is an actual bike noise signal. The time variations of noise power,− − • − −, is shown for comparison.

each subband, whose coefficients are independently updated using the FxLMS-Newton algorithm. The proposed algorithms with system identification and predictive configurations were evaluated using different kinds of actual noise signals. In all cases, simulation results show that the proposed approaches allow a significant reduction of the computational of the adaptive FxLMS-Newton algorithms, while keeping convergence performance nearly the same as that of the conventional ones. Simulation results also show that the ANC with system identification configuration can properly handle signals with strong correlation as well as with a week correlation between their samples. On the other hand, using an ANC with predictive configuration, one can achieve a fairly good cancellation level if the samples of noise signals are strongly correlated among them, as it happens with the airplanes, bells, and motor signals, as shown in Figs. 13.17–13.21. However, when the samples of input signal are weakly correlated between them, the cancellation level is smaller, as shown in Fig. 13.22. The reason is that, as the correlation among consecutive samples of noise signal becomes stronger, the ANC system may estimate the noise signal with a higher accuracy, achieving in this way a better cancellation performance.

References

1. *Kuo, S.M. and Morgan, D.* Active noise control systems, John Wiley and Sons, New York, 1996.
2. *Tapia Sánchez, D., Bustamante, R., Pérez Meana, H., and Nakano Miyatake, M.* Single Channel Active Noise Canceller Algorithm Using Discrete Cosine Transform. Journal of Signal Processing, 2005, vol. 9, no. 2 pp. 141–151.
3. *Harris, C.M.* Handbook of Acoustical Measurements and Noise Control, McGraw Hill, New York, 1991.
4. *Beranek, L.L. and Ver, I.L.* Noise and Vibration Control Engineering: Principles and Applications, Wiley, New York, 1992.

5. *Erickson L.J.*, *Allie M.C.*, *Bremigon C.D.*, *and Gilbert J.A.* Active noise control and specifications for noise problems, Proc. of Noise Control, pp. 273–278, 1988.
6. *Nelson*, *P.A. and Elliot*, *S.J.* Active Control of Sound, Academic Press, San Francisco Ca. 1992.
7. *Aoki*, *T. and Morishita*, *T.* Active noise control system in a duct with partial feedback canceller. IEICE Trans. on Fundamentals of Electronics, Communications and Computer Science, 2001, vol. 84, no. 2 pp. 400–405,
8. *Usagawa T. and Shimada*, *Y.* An active noise control headset for crew members of ambulance. IEICE Trans. on Fundamentals of Electronics, Communications and Computer Science, 2001, vol. 84, no. 2 pp. 475–478,
9. *Widrow*, *B. and Stearns*, *S.* Adaptive Signal Processing, Prentice Hall, Englewood Cliffs N,J., 1985.
10. *Muneyasu*, *M.*, *Wakasugi*, *Y.*, *Hisayasu*, *O.*, *Fujii*, *K.*, *and Hinamoto*, *T.* Hybrid active noise control system based on simultaneous equation method. IEICE Trans. on Fundamentals of Electronics, Communications and Computer Science, 2001, vol. 84, no. 2 pp. 479–481.
11. *Petraglia*, *M. and Mitra*, *S.* Adaptive FIR filter structure based on the generalized subband decomposition of FIR filters. IEEE Trans on Circuits and Systems-II. 1993, vol. 40, no. 6, pp. 354–362.
12. *Pérez Meana*, *H.*, *Nakano Miyatake*, *M.*, *and Niño de Rivera*, *L.* Speech and audio signal application, Multirate Systems: Design and application, Ed. G. Jovanovic, Idea Publising Group, 2002, pp. 200–224,
13. *Nakano-Miyatake*, *M. and Perez-Meana*, *H.* Fast orthogonalized FIR adaptive filter structure using a recurrent Hopfield-Like network. Lectures in Computer Science 1606: Foundations and Tools for Neural Modeling, pp. 478–487, Springer Verlag, Berlin, 1999.
14. *Narayan S.*, *Peterson*, *A.*, *and Narasimha*, *J.* Transform domain LMS algorithm. IEEE Trans. Acoustic, Speech, Signal Processing, 1983, vol. 31, no. 6, pp. 609–615.
15. *Perez-Meana*, *H.*, *Nakano-Miyatake*, *M.*, *Martinez*, *A.*, *and Sanchez*, *J.* A Fast Block Type Adaptive Filter Algorithm with Short Processing Delay. IEICE Trans. on Fundamentals of Electronics, Communications and Computer Science, 1996. vol. E79, No. 5, pp. 721–726,
16. *Farhang-Boroujeny B.* Adaptive Filters: Theory and Applications, John Wiley&Sons, New York, 2000.
17. *Zelinski*, *R. and Noll*, *P.* Adaptive transform coding of speech. IEEE Trans. on Acoustic, Speech and Signal Processing, 1977, Vol. ASSP 25, no. 2, 299–309,
18. *Perez-Meana H. and Tsujii S.* A Fast parallel form IIR adaptive filter. IEEE Trans. on Signal Procesing, vol. 39, no. 9, pp. 2118–2122, 1991.
19. *Shynk J.* Adaptive filtering. IEEE Signal Processing Magazine, 1989, No. 4, pp. 4–21.
20. *Tapia*, *D.*, *Perez-Meana*, *H.*, *and Nakano Miyatake*, *M.* Orthogonal methods for filtered X algorithms applied to active noise control using the discrete cosine transform. Zarubezhnaya Radioelektronika, 2003, vol. 2003, no. 6, pp. 59–67.
21. *Stearns*, *S.* Fundamentals of Adaptive Signal Processing. Advanced Topics on Signal Processing, Ed. J. Lim and A. V. Oppenheim, Prentice Hall, Englewood Cliffs, NJ, 1988.

Chapter 14

ADAPTIVE EQUALIZERS

Since the bit rate increased in most digital communication systems, the requirement of better algorithms for inter-symbol interference reduction has also increased. To solve this problem, several efficient equalizer algorithms have been proposed in the last several years; some of them are presented in this chapter.

14.1. Introduction

The development of the communications and computer technology has lead to a widespread use of high-speed wire and wireless data communications. This became possible due to the development of efficient adaptive systems capable to reduce intersymbol interference, enabling, in the case of using the telephone channel, the development of the xDLS communication systems.

In the case of the xDLS systems, generally, the telephone communication channel is nearly stationary and presents low distortion. Then the main problem is the interference introduced in the receiver by its own transmitter or by other transmitters operating in the same wideband communications channel. To reduce the interference in xDLS systems, adaptive equalizers can be used along with cross-talk interference cancellers, which presents a structure similar to that of an echo canceller.

The wireless data communication systems do not present transmitter interference, however the channel distortion is significant. To compensate the severe distortion introduced by the rapid time-varying mobile communication channels, adaptive algorithms with high convergence rates are required to update the DFE coefficients. The LMS adaptive algorithm, widely used in several adaptive filter practical applications, has a very low computational complexity, providing fairly good performance in stationary and slow time-varying communication channels [15]. However, its low convergence rate limits its ability to track rapid time-varying fading multipath channels often found in mobile communications systems [15, 16]. On the other hand, the RLS adaptive algorithm has a much higher convergence rate than that of the LMS algorithm and low sensitivity to the additive noise, although its computational complexity is much higher than that of the LMS algorithm [1, 15, 16]. However, because of its ability to track relatively fast time-varying communication channels, the RLS algorithm with different memory length is often used to update the adaptive DFE coefficients vector [15]; besides, several algorithms have been proposed to reduce the RLS computational complexity, such as the fast Kalman algorithm [15, 17], which reduce the computational complexity of the adaptive algorithm from $O(N^2)$ to $O(N)$. This represents a considerable reduction of the computational complexity. However, this algorithm and another of this type that have been proposed in the last few years may become numerically unstable [1, 15, 17]. Thus, the high computational complexity of the RLS algorithm and the numerical instability of

its low computational complexity modifications still present several problems when used in land mobile communication systems, such as cellular telephone systems.

Wireless radio communications networks need to increase the number of users allowed in the system. As a consequence, modulation and multiple access techniques designed specifically for wireless channels will play an important role in achieving this goal. For mobile communication systems, diversity reception is essential to reduce the effects of fading radio channels. Direct-sequence code division multiple access (DS-CDMA) is a multiplexing technique where several independent users share a common channel by modulating preassigned signature waveforms. The receiver then observes the sum of the transmitted signals over an additive white Gaussian noise (AWGN) channel.

The major limitation on the performance and channel capacity of the DS-CDMA system is the multiple-access interference (MAI) due to simultaneous transmissions. The conventional matched filter (MF) detector cannot suppress MAI effectively, and it suffers from the near–far problem. Since CDMA is not fundamentally MAI limited, multiuser detection (MUD) techniques can substantially improve the performance of a CDMA system. The optimal multiuser detector is, essentially, a maximum-likelihood (ML) sequence detector. However, because it has a prohibitive complexity, many other multiuser detectors with relatively low complexity, such as the decision feedback detector, parallel interference canceller, and linear multiuser detectors, have been developed. All of them provide suboptimal solutions, such as the linear decorrelator, which removes all cross-correlations between active users, eliminating in such way the MAI at the price of enhancing the additive noise [2].

Recently, blind adaptive multi-user detection has received special attention and several blind adaptive detectors have been proposed [1–3]. The main motivation for employing a blind detector is to avoid the necessity to have a training sequence, which is commonly required in most adaptive multi-user detectors proposed previously. Blind detection avoids the requirements for a reference signal, and, under appropriate initial conditions, its performance is not considerably degraded as compared to detectors requiring a training sequence. In this sense, the constant modulus algorithm (CMA) has been widely applied to cancel intersymbol interference (ISI) for digital transmission through band-limited channels. So, based on the combined channel and equalizer parameter space, a finite-length tap filter with CMA tap updates will be able to converge closely to the global minimum.

This chapter presents interference cancellation systems to be used in wire as well as data communication systems using digital, analog, fuzzy, and blind detection approaches.

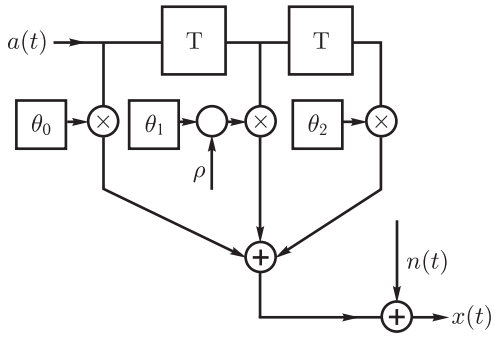


Fig. 14.1. A three-ray Rayleigh fading model of a mobile communication channel.

14.2. Channel Model for Land Mobile Communication

A realistic evaluation of any equalizer method strongly depends on the model of communication channel used [39, 41–43]. Therefore, considerable research has been carried out to properly model mobile communication channels. As a result of this intensive research, several models have been reported in literature. Among them, one of the most widely used is the Three-Ray Rayleigh Fading Model, given as follows:

$$h(t) = \theta_0 \exp(j\varphi_0)\delta(t) + (\theta_1 + \rho)\delta(t - T) + \theta_2 \exp(j\varphi_2)\delta(t - 2T), \tag{14.1}$$

where θ_k $(k = 0, 1, 2)$ are independent and Rayleigh-distributed random numbers, φ_k $(k = 0, 2)$ are independent and uniformly distributed random numbers, and ρ is a deterministic component.

Experiments carried out to validate the above model show that, in an urban area, the received signal usually consists of multipath components which can be thought of as being independently traveling plane waves whose phases, amplitudes, incoming angles, and time delays are random variables [40]. Thus, the mobile communication channel can be assumed as a random process that is the result of two overlapping stationary random processes. One of them, termed the shadowing random process, is related to the large-scale fluctuations, e.g., in urban area, to the density and average height of buildings or the width of streets. It can be assumed to be stationary over several hundreds of meters [40]. The second is the short-term random process that is mainly related to the motion of the mobile station and is responsible for the fluctuations of the propagation channel within fractions of wavelengths. The short-term random process can be assumed to have statistics of the Rayleigh type and be stationary over 4–5 m in the 900-MHz frequency band [40]. In a suburban or rural area mobile communication channels can be simulated by adding a deterministic component to the impulse response of the mobile communication channel urban area described above [39, 41–43]. Such a deterministic component stands for the main path characterized by a power ρ relative to the random component [40]. For low values of ρ, i.e., near 0 dB, random contribution prevails [39, 41–43], which results in the Rayleigh distribution for the channel. However, even a moderate increase of ρ ($5 \sim 10$ dB) causes noticeable changes in both the statistic and dynamic properties of the communications channel transfer function [35]. On the basis of these qualitative interpretations of some experimental results, the following classification can be drawn [39, 41–43]:

(a) Urban center with building density $> 30\%$,

$$\rho \ll 0 \text{ (dB). Only multipath component.}$$

(b) Urban area with a building density of 20%–30%

$$0 < \rho < 4 \text{ dB.}$$

(c) Urban area with a building density of 10%–20%

$$4 < \rho < 6 \text{ dB.}$$

(d) Suburban area

$$6 < \rho < 10 \text{ dB.}$$

(e) Open rural area

$$\rho > 10 \text{ dB.}$$

Figures 14.2–14.4 show the frequency responce of some typical commnicacion channels. In Fig. 14.2 the frequency response of a high-qualtiy telephone channel is shown. Figure 14.3 shows the frequency response of a three rays communication channel

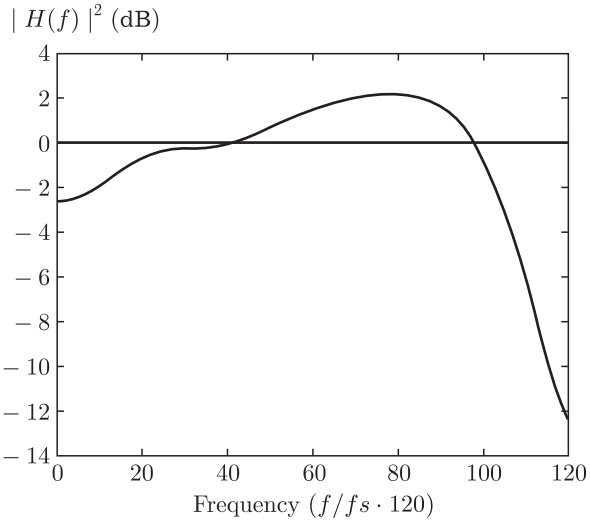


Fig. 14.2. Frequency response of a high-quality telephone channel.

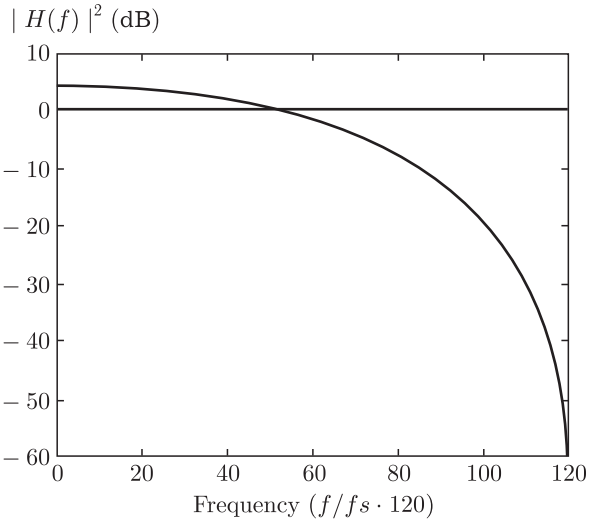


Fig. 14.3. Frequency response of a communication channel often found in mobile communications, with large spectral distortion.

(Fig. 14.1) and, finally, in Fig. 14.4 the frequency response of a five rays communicacion channels is shown.

14.3. Interference Cancellation in Wire Data Communication Systems

Figure 14.5 shows a block diagram of a typical data communications channel operating in a wire-based communications system. As shown in this figure, the connection of the transmitter and receiver in the local station, the 2–to–4 wires conversion, is carried

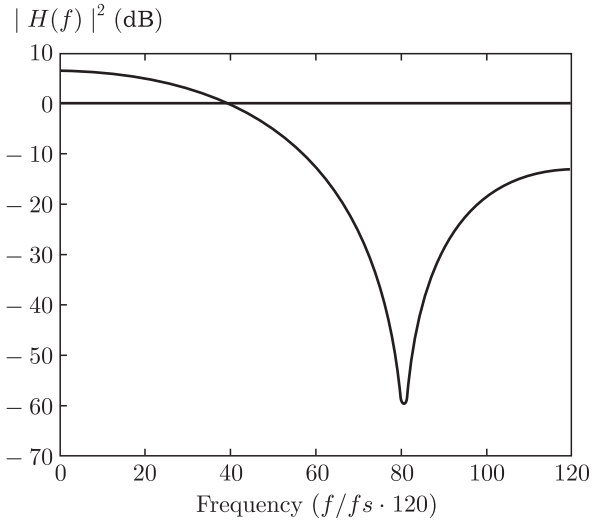


Fig. 14.4. Frequency response of a communication channel often found in mobile communications, with large spectral nulls.

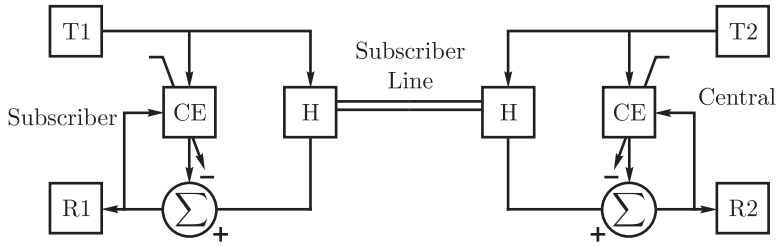


Fig. 14.5. Adaptive Echo Cancellation Model, where $T1, R1$ are the Transmitter and Receiver at the subscriber side and $T2, R2$ are the Transmitter and Receiver at the service provider side. EC is an Adaptive Echo Canceller and H is the Hybrid circuit.

out using a hybrid circuit. However, because the hybrid circuit is not perfectly balanced, a significant amount of transmitted signal arrives to the receiver, producing a significant distortion, which must be reduced along with the intersymbol interference due to the non-ideal communication channel characteristics. In order to solve this problem, three different structures can be used: the intersymbol interference predicted decision feedback equalizer, ISI-DFE, the noise predicted decision feedback equalizer, NP-DFE, and the hybrid decision feedback equalizer, H-DFE. In all cases the canceller and the equalizer can be adapted independently, as described in Section 14.2.1, or jointly, as shown in Section 14.2.2.

14.3.1. Independently Updated Canceller and Equalizer Structures. To reduce the interference introduced by the reflected signal from the transmitted to the receiver or even speech signal if the data communication system share the channel with a telephone one, three different equalizer structures have been proposed: the ISI-DFE, the NP-DFE, and the H-DFE, which are shown in Figs. 14.6–14.8. The first approach is the ISI-DFE, which is commonly known as DFE (Fig. 14.6). Here, the detected data are inserted into the feedback section to improve the DFE performance. In situations when a colored noise is the main distortion source, the NP-DFE, shown in Fig. 14.7, is a suitable

choice. Here, a linear predictor, intended to reduce the colored noise, is inserted after a feedforward equalizer. This structure performs fairly well if the interference noise is strongly correlated. Finally, the H-DFE approach, shown in Fig. 14.8, combines the desirable properties of both structures. In all the additive, interference is reduced by placing the canceller before the equalization process, adapting independently both of them.

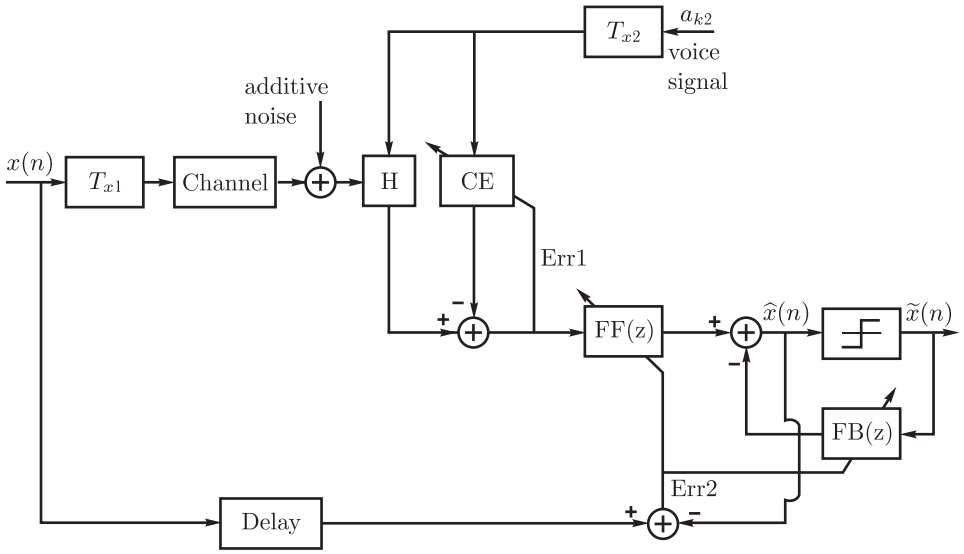


Fig. 14.6. ISI-DFE with adaptive echo cancellation approach.

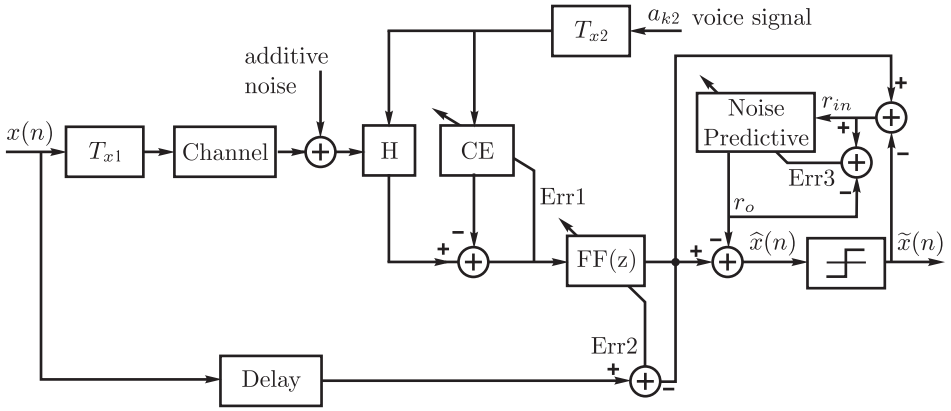


Fig. 14.7. NP-DFE with adaptive echo cancellation approach.

14.3.2. Jointly Updated Canceller and Equalizer Structures. The second approach consists in jointly updating the canceller and equalizer using a common error, as shown in Figs. 14.9–14.11.

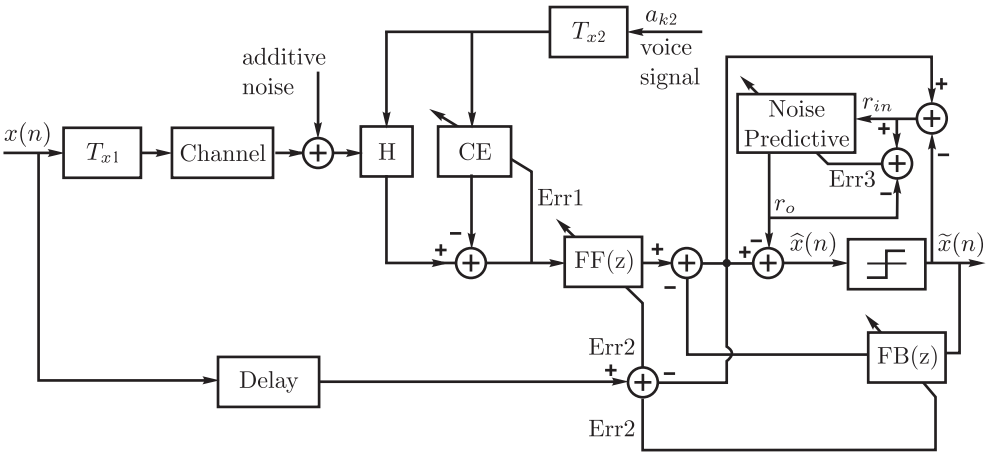


Fig. 14.8. H-DFE with Adaptive Echo Cancellation approach.

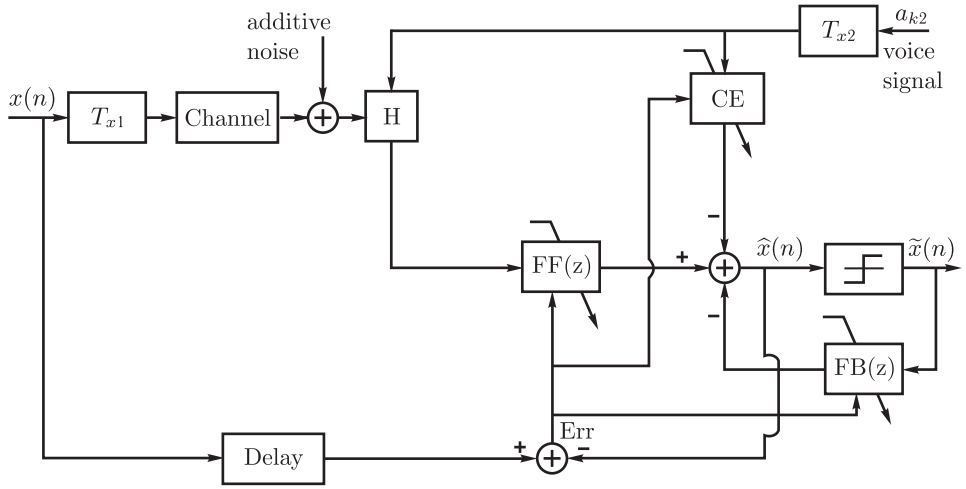


Fig. 14.9. ISI-DFE with adaptive echo Cancellation model using a common error.

Figures 14.12 and 14.13 show the performance of equalizers updated using separated and common errors, respectively when the used channel presents severe intersymbol interference.

14.4. Analog Equalizer Structure

Recently, neural-network based adaptive equalizers have been proposed, which have the ability to track fast variations on the communication channels impulse response [23–25]. However, although they are reported to perform fairly well in many practical situations, their computational complexity in some cases is much higher than that of conventional DFE using the RLS algorithm [23–25]. This may limit their use in some practical applications. On the other hand, the interest in adaptive analog systems has grown in last few years, because they have the potential to handle much higher

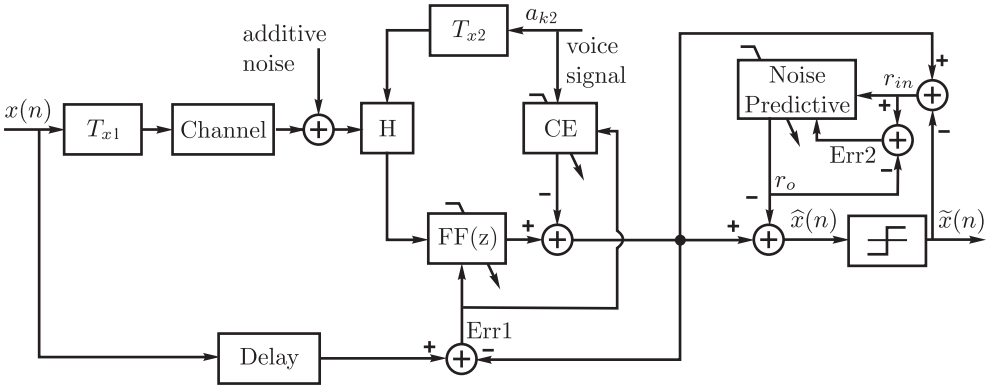


Fig. 14.10. NP-DFE with adaptive echo cancellation model using a common error.

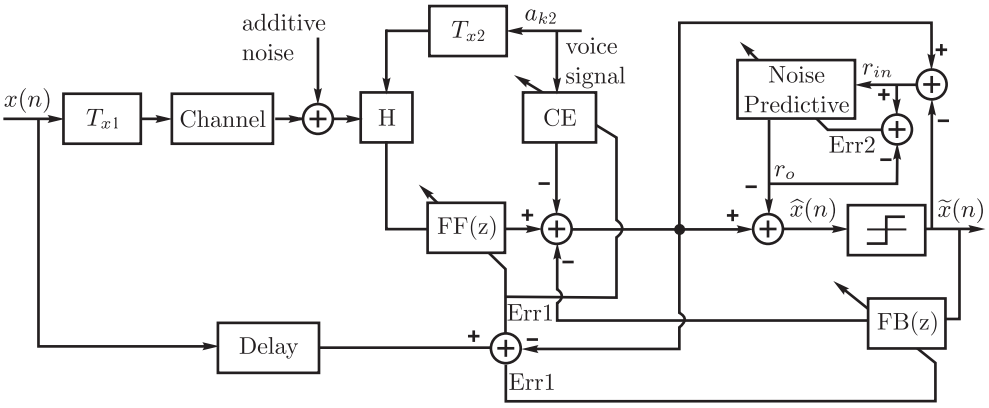


Fig. 14.11. H-DFE with adaptive echo cancellation model using a common error.

frequencies with a smaller implementation size and lower power requirements than their digital counterparts [18, 19, 26, 27].

This section describes an analog sampled data recursive least square (RLS) adaptive DFE structure that extends previous proposed structure [19] for handling complex valued input signals in which the DFE coefficients vector is updated by using a continuous time modified Hopfield Network [21, 27–31]. In the equalizer structure M time delays are inserted between consecutive coefficients to provide faster convergence rates and lower misadjustment, while keeping the same number of coefficients. This fact increases the convergence rate and reduces the misadjustment of adaptive algorithm by the factor M. Thus, the equalizer structure has, potentially, a much smaller implementation size, lower power requirement, much higher convergence rates, and better tracking ability than its digital counterparts. Computer simulations are given to show that the analog structure has very similar performance than the conventional DFE with RLS adaptation algorithm in the stationary or slowly time varying channel. However, it performs much better than the conventional DFE in rapidly time varying communication channels.

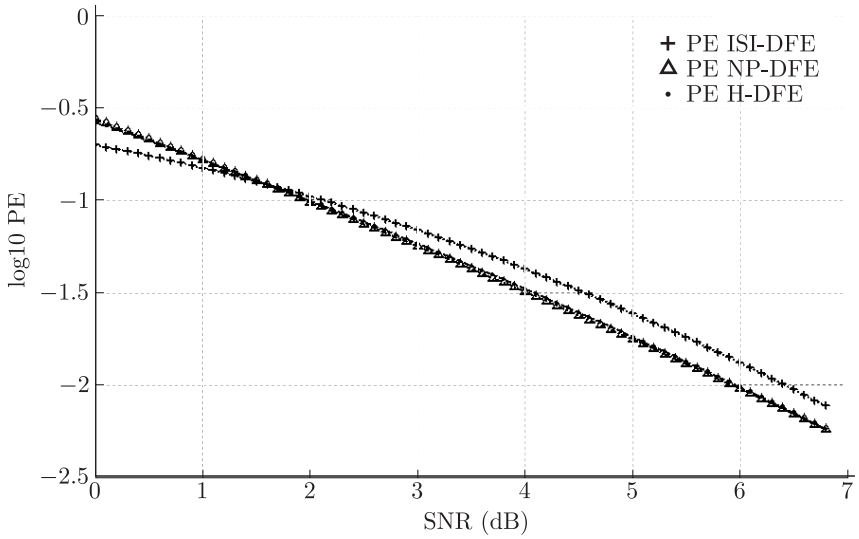


Fig. 14.12. Bit Error Rate (BER) Performance with the canceller and equalizer updated using different errors.

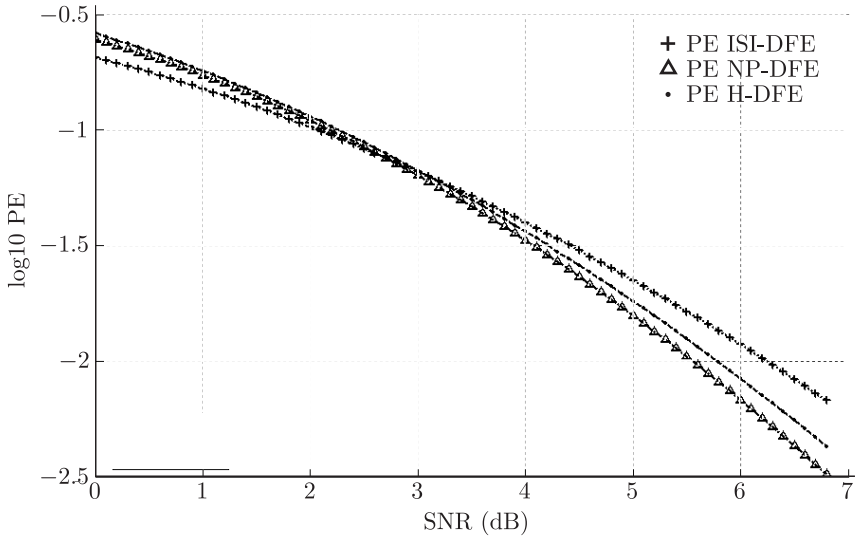


Fig. 14.13. Bit Error Rate (BER) Performance with the canceller and equalizer updated using a common error.

Consider the continuous time decision feedback equalizer (DFE) structure shown in Fig. 14 (a), with the output signal $y(t)$ given by

$$y(t) = \sum_{k=0}^{N-1} w_k x_k(t), \tag{14.2}$$

where $x_k(t)$ is given by

$$x_k(t) = c_k \left(x_r(t - nMT) + x_i(t - nMT) \right). \tag{14.3}$$

In (14.3), c_k is the attenuation introduced by the k-th delay stage and w_k is the k-th expansion coefficient, which is estimated so that the output error energy attains a minimum, where

$$e(t) = d(t) - y(t) \tag{14.4}$$

is the output error and $d(t)$ is the reference signal, which is a known symbol during the training period and the detected symbols during the operation period. From equations (14.2) and (14.4) it follows that the output error energy is given by

$$\int_0^t \left| d(\tau) - \sum_{k=0}^{N-1} w_k x_k(\tau) \right|^2 d\tau. \tag{14.5}$$

Next, using the orthogonality principle in least-square estimation, after some modifications, we obtain the optimal coefficients vector in the form [1]

$$\mathbf{W} = \Phi_{xx}^{-1} \Phi_{xd}, \tag{14.6}$$

where $\mathbf{W}$ is the coefficients vector whose k-th element is w_k, $\mathbf{\Phi}_{xd}$ is the correlation vector between the all-pass sections output signals and the reference whose k-th element is given by

$$\varphi_{xd}(k) = \int_0^t d(\tau) x_k^*(t) d\tau, \tag{14.7}$$

where $*$ denotes the complex conjugation, and $\mathbf{\Phi} g_x$ is the correlation matrix between the all-pass sections output signals whose $(j,k)-$th element is given by:

$$\varphi_{xx}(j,k) = \int_0^t x_j(\tau) x_k^*(t) d\tau. \tag{14.8}$$

In order to derive an adaptive algorithm for on-line estimation of the coefficients vector $\mathbf{W}$, consider a continuous time Hopfield neural network shown in Fig. 14.14(b), whose output signal is given by [21, 27, 6, 28, 30]

$$\begin{gathered} \frac{d}{dt} w_k(t) = \frac{1}{RC} w_k(t) + \sum_{m=0}^{N-1} p_{m,k} w_m(t) + b_k, \\ k = 0, 1, 2, \ldots, N-1, \end{gathered} \tag{14.9}$$

where $w_k(t)$ is the k-th node complex-valued output signal, R and C are real positive constants, $p_{m,k}$ is the connection weight of the path going from the m-th node to the k-th one and b_k is a real constant. Taking the Laplace transform of (87), we obtain

$$s\mathbf{W}(s) + \frac{1}{RC}\mathbf{W}(s) + \frac{1}{C}\mathbf{P}W(s) = \frac{1}{C}\mathbf{B}. \tag{14.10}$$

Next, using the Final Value Theorem, we may conclude that, in a sufficiently long time interval, the output vector of the modified Hopfield Network takes the form [18, 27, 28]

$$\mathbf{W}(\infty) = \left[\mathbf{I} - \mathrm{R}\mathbf{P} \right]^{-1} R\mathbf{B}. \tag{14.11}$$

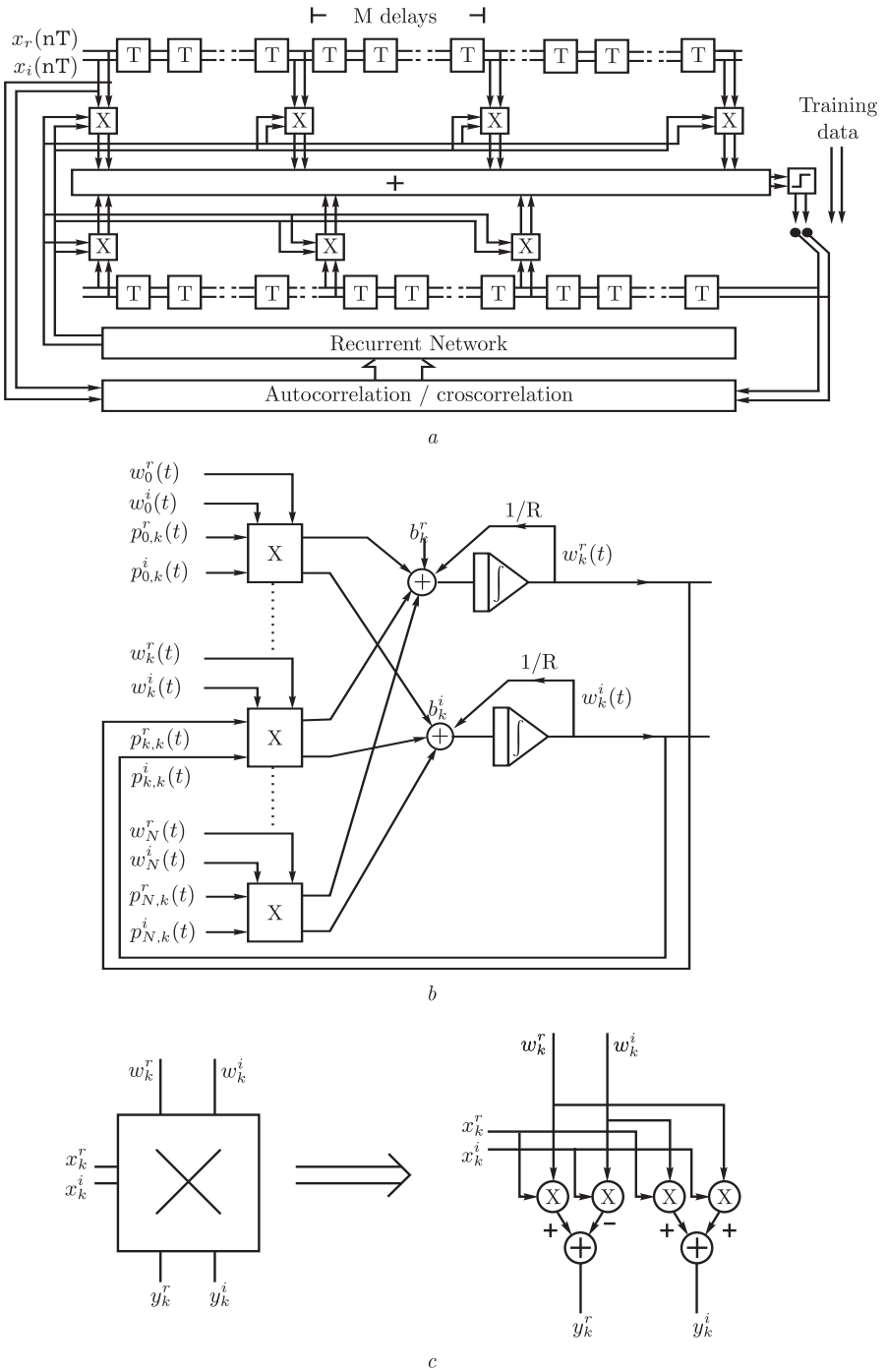


Fig. 14.14. Analog sampled data decision feedback equalizer based on a modified Hopfield network, (b) k-th node of a modified complex valued Hopfield network used for updating the proposed DFE coefficients vector. (c) Implementation of a complex multiplier.

Consider now the adaptive filter optimal coefficient vector, which is given by equation (14.11), and assume that

$$\mathbf{P} = \frac{\mathbf{I} - \Phi_{xx}}{R} \tag{14.12}$$

and

$$\mathbf{B} = \frac{\Phi_{xd}}{R}. \tag{14.13}$$

As follows from equations (14.10)–(14.13), after convergence, the Hopfield Neural Network provides the optimal solution to the Wiener–Hopf equation and then can be used for estimating the optimal coefficients vector of an analog time adaptive filter structure [19, 27, 28].

The adaptive algorithm is a recursive implementation of equation (14.6). Then, the same convergence characteristics as for the standard RLS algorithm [1], when both operate under the same conditions, can be expected. Hence, the misadjustment produced by both algorithms due to the weight vector noise is approximately given by

$$M_a = (1 - \gamma)N/(1 + \gamma), \tag{14.14}$$

where γ is the forgetting factor and $N$is the number of equalizer coefficients. On the other hand, the variation of the power error with time is given by [1]

$$\mathbf{K}(n) = \sigma^2/(n\lambda_{\min}). \tag{14.15}$$

Here, assuming that the systems memory in seconds remain constant if the sampling period is reduced by M, equations (14.14) and (14.15) take the form

$$M_a = \frac{(1 - \gamma)\, N}{2M + \gamma + 1} \tag{14.16}$$

and

$$\mathbf{K}(n) = \sigma^2/(nM\lambda_{\min}). \tag{14.17}$$

Hence, the analog DFE structure provides lower misadjustment with better tracking ability than the standard RLS adaptive algorithm.

14.4.1. Computer Simulation. In this section, we apply the modeling ideas described in the previous section to a simulated mobile radio channel with three propagation paths ($L = 3$), where the time behavior of the fading channel is characterized by half the Doppler spread. That is, the variation rate of the multipath channel is given by [43]

$$f_D = f_c \frac{V}{C}, \tag{14.18}$$

where f_c is the carrier frequency, V is the mobile speed and C is the speed of light. The simulation results presented in this section are not intended to be a complete computer simulation study of the proposed DFE structure, but offer an illustration of its applicability and performance. Here, each time-varying communication channel is a linear combination of three paths (one direct path and two reflectors) as shown in Fig. 14.1, that is [19, 40, 43]

$$h(k, t) = \theta_r f_r(k), \quad r = 0, 1,2, \tag{14.19}$$

where

$$f_1(k) = 1.0, \tag{14.20}$$

$$f_0(k) = \exp\left(\frac{j2\pi k}{M_1}\right), \tag{14.21}$$

and

$$f_2(k) = \exp\left(\frac{j2\pi k}{M_2}\right), \tag{14.22}$$

where, for a rapid time-varying channel, M_1 and M_2 are assumed to be 120 and 200, respectively, and $\theta_0 = 0.5$, $\theta_1 = 1$, and $\theta_2 = 0.5$. These numbers are rather realistic for a carrier frequency of 900 MHz bit rate of 20 Kb/s and vehicle speed of 100 km/h. The input was a 4-QAM symbol series filtered through the channel and corrupted with white Gaussian noise (Additive white Gaussian noise AWGN). No error-correcting code was used. The burst structure of the transmitted signal consists of 2 training data and 6 information data. To evaluate the performance of the proposed DFE structure shown in Fig. 14.14 by computer simulation, the sampling rate was assumed to be 10 times faster than the symbol rate. Thus, the number of delay sections inserted between consecutive DFE coefficients was 10. To simulate the non-ideal characteristics of the delay line. a loss of about 0.01 dB was added between consecutive filter taps. The modified Hopfield network was simulated by solving equation (14.9) as follows:

$$w_k(nT) = \exp\left(-\left(\frac{1}{RC} + p_{kk}\right)nT\right) \times \left[\sum_{m=0, m\neq k}^{N-1} p_{km} w_m(nT) + b_k\right], \tag{14.23}$$

where $*$ denotes the convolution and T, the symbol rate, is assumed equal to 1.0. Finally, the integrators required to estimate the autocorrelation and cross correlation functions were replaced by low-pass filters with appropriate cutoff frequencies. The additive noise was a white noise sequence.

Three different cases were considered for evaluation by computer simulation, namely: stationary communication channels, slow time-varying communication channels, and rapid time-varying communication channels.

Case 1: Stationary Communication Channel.

The performance of the proposed structure was evaluated using three different stationary channels whose power spectrum densities are shown in Figs.14.2–14.4, respectively. The equivalent communication channel shown in Fig. 14.2 (channel 1) is a high-quality typical telephone channel, which is relatively easily compensated by the adaptive DFE [1, 22]. In contrast, the equivalent communication channels 2 (Fig. 14.3) and 3 (Fig. 14.4) have deep spectral nulls, which cause serious information distortion. The spectral characteristics of the equivalent channels shown in Figs. 14.3 and 14.4, often found in multi-path fading mobile communication channels, cause serious distortion of the transmitted signals [1]. Figure 14.15 shows that the bit error rate (BER) of the proposed DFE structure and the conventional DFE structure using the RLS algorithm [23], when required to equalize the three communication channels mentioned above. The forgetting factor [1] of the RLS algorithm is equal to 0.99 and the modified Hopfield Network is considered to have converged when ε_i <0.001 ($i = 0, 1, \ldots N$), where ε_i is the difference between two consecutive samples of the i-th node Hopfield ANN output signal.

Case 2: Slowly Time Varying Communication Channel.

We evaluated the DFE structure using two slow time-varying communication channels. In the first one, we used the «snapshot» method [46], in which we assumed that the channel remains constant over 100 data symbols. Here, the impulse response of an equivalent communication channel is given by equations (14.18)–(14.21) with $M_1 = 120$ and $M_2 = 200$ and k given by the integer part of $S_r/100$, where S_r is the symbol number. The simulation result is shown in Fig. 14.16, where the forgetting factor used for the RLS algorithm is equal to 0.99. Figure 14.17 shows the performance of the proposed DFE and conventional DFE structure using the RLS algorithm when both are required

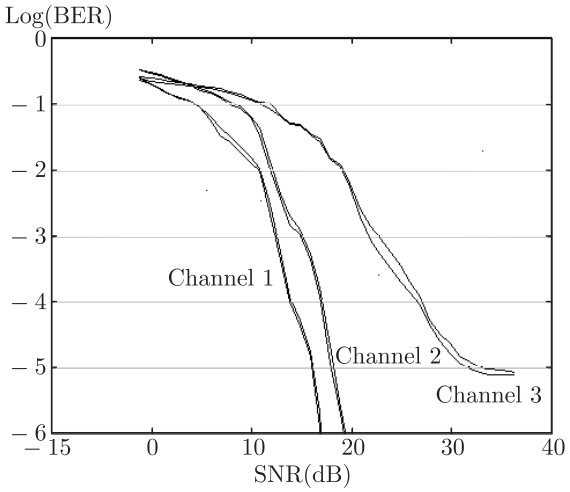


Fig. 14.15. Bit error rate obtained by using the proposed and conventional DFE structures when both are required to equalize the three communication channels described above.

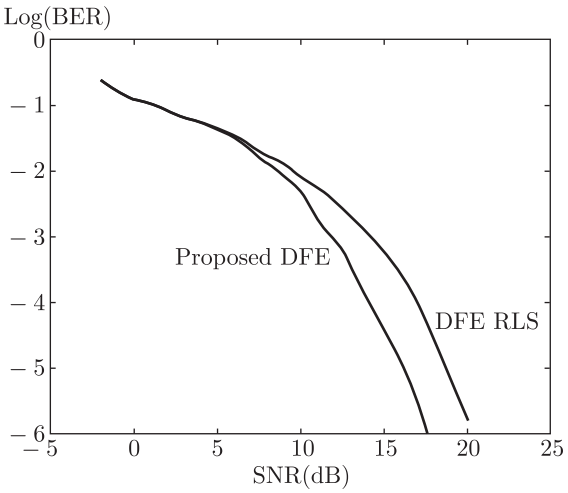


Fig. 14.16. Bit error rate obtained by using the proposed and conventional DFE structures when both are required to equalize a slowly time-varying communication channel. The snapshot method was used.

to equalize a continuously slow time-varying communication channel, whose equivalent channel is given by the equations (14.18)–(14.21) with $M_1 = 1200$ and $M_2 = 2000$, and k is the symbol number. A forgetting factor equal to 0.99 was used in the RLS algorithm. Figure 14.18 shows the trace of the theoretical and estimated coefficients when using the proposed structure in a slow time-varying environment, and Fig. 14.19 shows the coefficients trace when the conventional RLS algorithm is used. These figures show that the analog DFE structure outperforms the conventional DFE with RLS algorithm when the communication channel varies slowly in time.

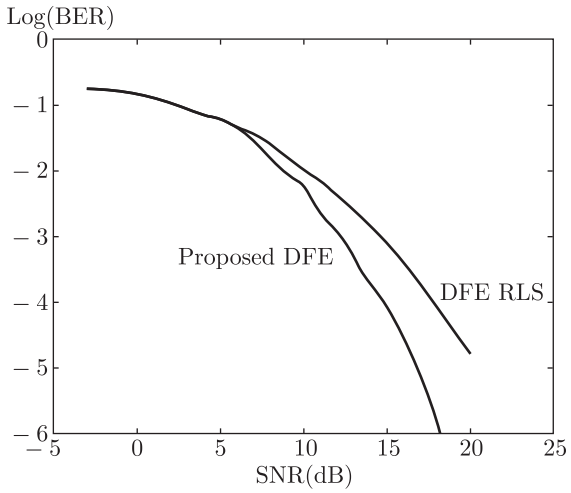


Fig. 14.17. Bit error rate obtained by using the proposed and conventional DFE structures when both are required to equalize slowly time-varying communication channels.

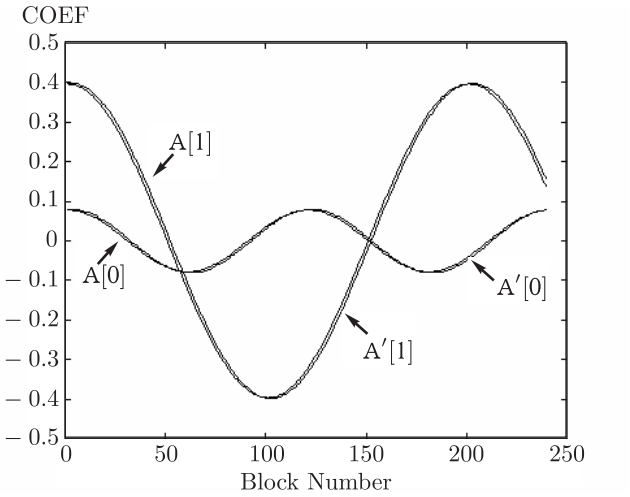


Fig. 14.18. Trace of the first A'[0] and second A'[1] coefficients of proposed DFE structure when it is required to equalize a slowly time varying communication channel. The theoretical values A[0] and A[1] are also shown.

Case 3: Rapidly Time Varying Channel.

We evaluated the convergence performance of analog DFE structure and compared it with the performance of a conventional DFE structure when both are required to equalize rapidly time-varying communication channels, often found in mobile communication systems. The channel impulse response varies according to equations (14.18)–(14.21) in the symbol rate. It is assumed that $\rho = 3$ dB, which can be considered as a standard situation in several large cities, such as Mexico City. The sampling rate of the conventional DFE structure input signal was assumed to be equal to the symbol rate, while in the proposed DFE structure the sampling rate was assumed to be 10 times faster than

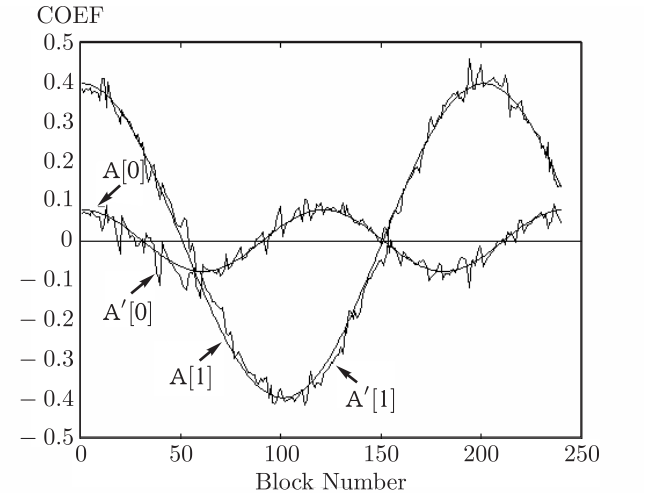


Fig. 14.19. Trace of the first A'[0] and second A'[1] coefficients of conventional DFE structure when it is required to equalize a slowly time-varying communication channel. A[0] and A[1] are the theoretical values.

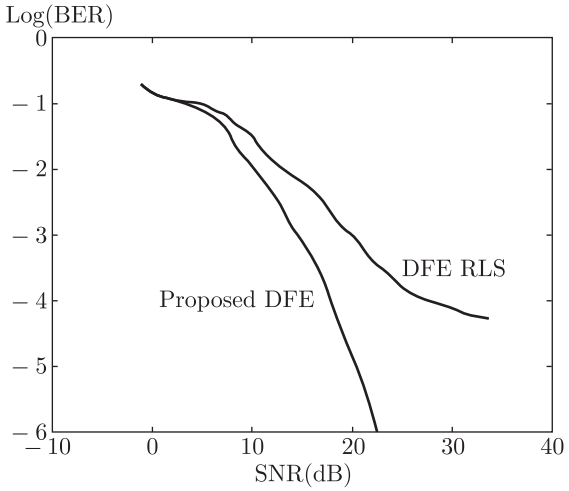


Fig. 14.20. Bit error rate obtained by using the proposed and conventional DFE structures when both are required to equalize a rapidly time-varying communication channel.

the symbol rate. In both cases, the communication channel varies with each sampling period of a different variation rate, such that at the end of each symbol period both reach the same value. Figure 14.20 shows the bit error rate (BER) of the analog DFE structure along with the BER provided by the conventional DFE structure with the RLS algorithm. The forgetting factor of the RLS algorithm is equal to 0.9, which is the best value according to the simulations shown in Fig. 14.21. Figures 14.22 and 14.23 show the coefficient traces of theoretical and estimated coefficients obtained using the analog and conventional algorithms operating in a rapid time-varying environment. Figures 14.20, 14.22, and 14.23 show that the analog structure outperforms, significantly, the performance of conventional discrete time DFE structure with RLS adaptation algorithm.

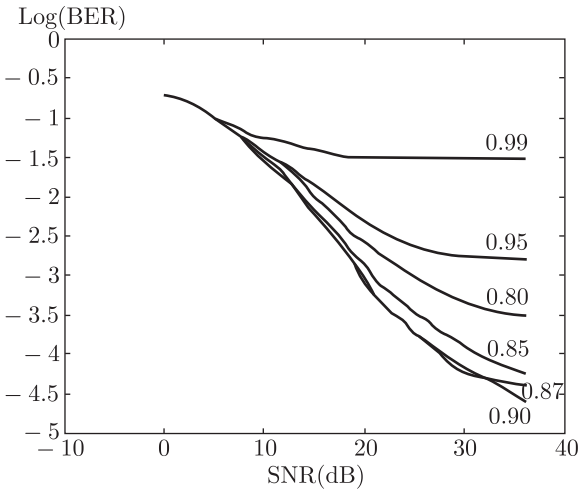


Fig. 14.21. Bit error rate obtained by using the conventional DFE structures with different forgetting factors when it is required to equalize a rapidly time-varying communication channel.

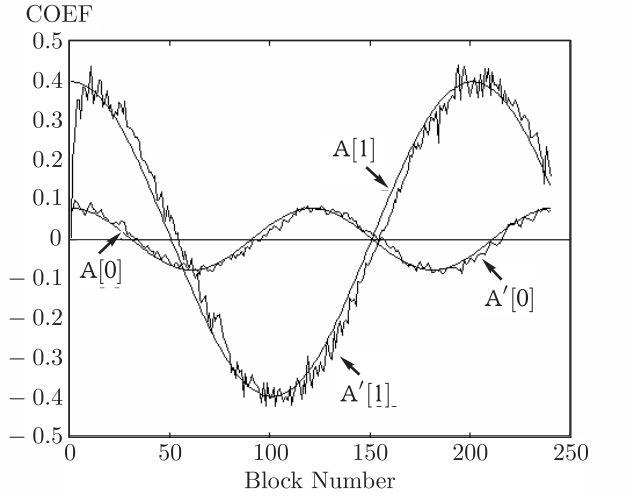


Fig. 14.22. Trace of the first A'[0] and second A'[1] coefficients of proposed DFE structure when it is required to equalize a rapidly time-varying communication channel. The theoretical values A[0] and A[1] are also shown.

14.5. Fuzzy Equalizer Structure

The equalizer structures described in Sections 14.3 and 14.4 assume that the communication channels are linear and then intended to compensate those linear distortions. However, although these structures are very efficient to handle linear distortion, their performance degrades when the distortions are due to nonlinearities of the communications channels. A suitable choice to handle these kinds of distortions is the standard additive model (SAM) fuzzy logic based equalizers. A standard additive model (SAM) system F is a set of rules of the form «If $X = A_j$ then $Y = B_j$», that maps

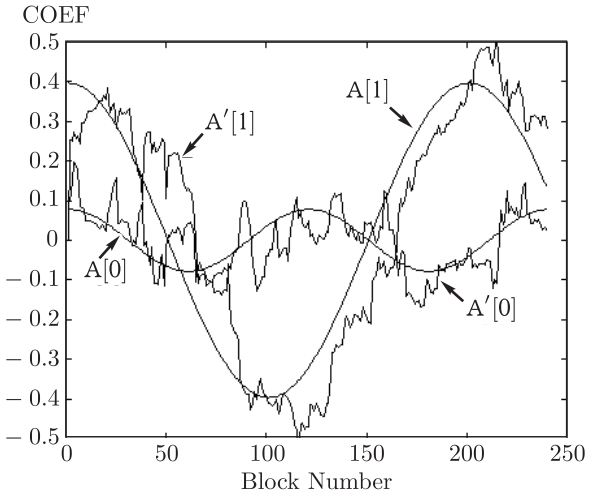


Fig. 14.23. Trace of the first A'[0] and second A'[1] coefficients of conventional DFE structure when it is required to equalize a rapidly time-varying communication channel. The theoretical values A[0] and A[1] are also shown.

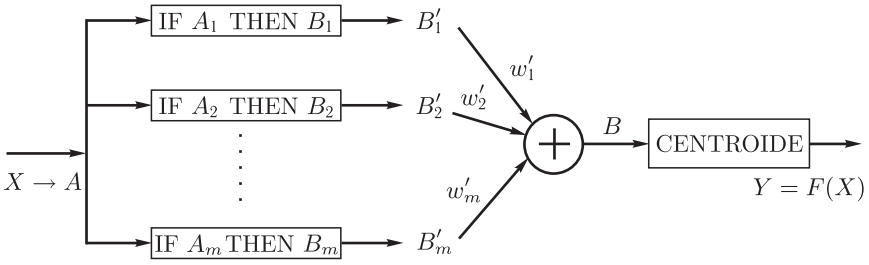


Fig. 14.24. SAM Model.

the inputs to outputs, such that the fuzzy system F can approach any arbitrary function f by covering its graph with rule patches and averaging the patches that overlap among them [49], as shown in Fig. 14.24, whose output is defined as

$$F(x) = \frac{\sum_{j=1}^{m} w_j a_j(x) V_j c_j}{\sum_{j=1}^{m} w_j a_j(x) V_j}, \tag{14.24}$$

where w_j is the adaptive weight, a_j is the if-part of the «fuzzy» or multivalued set A_j, $V_j$is the volume or the area of then-part set and C_j is the centroid of then-part set. All of these parameters can be adapted using a gradient algorithm as follows [49]:

$$w_j(n+1) = w_j(n) + \mu \frac{p_j(x)}{w_j(n)} \left(d(n) - F(x)\right)\left(C_j - F(x)\right), \tag{14.25}$$

$$V_j(n+1) = V_j(n) + \mu \frac{p_j(x)}{V_j(n)} \left(d(n) - F(x)\right)\left(C_j - F(x)\right), \tag{14.26}$$

$$C_j(n+1) = C_j(n) + \mu \left(d(n) - F(x)\right) p_j(x). \tag{14.27}$$

14.5.1. SAM based Equalizer Structure. The fuzzy equalizer algorithm is a modification of the SAM model applied to the channel equalization. From the SAM theory, if the modes or «peaks» of the then-parts sets are equal to the centroids of the then-part sets and if then-part sets have all the same areas or volumes and the same weights, then the SAM fuzzy system reduces to the center of gravity of the COG fuzzy model [49].

$$F(x) = \frac{\sum_{j=1}^{m} a_j(x) P_j}{\sum_{j=1}^{m} a_j(x)}. \tag{14.28}$$

The radial basis function networks or RBFs neuronal network theory is a special case of the SAM. This model is given for the COG model and the radial basis function (RBF) model of neuronal networks, under the following considerations:

$$\begin{gathered} y = z, \\ a_j(x) = \prod_{i=1}^{n} a_i^j(x_i) = \prod_{i=1}^{n} \mu_{A_i^j}(x_i), \\ V_j = 1, \\ C_j = z^j, \end{gathered}$$

it follows that

$$F(x) = \frac{\sum_{j=1}^{m} z^j \left(\prod_{i=1}^{n} \mu_{A_i^j}(x_i) \right)}{\sum_{j=1}^{m} \prod_{i=1}^{n} \mu_{A_i^j}(x_i)}, \tag{14.29}$$

where z^j is the point in R where the membership function achieves its maximum value and μ_A^j is a Gaussian membership function

$$\mu_{A_i^j}(x_i) = \exp\left[-\frac{1}{2} \left(\frac{x_i - x_i^j}{\sigma_i^j} \right)^2 \right], \tag{14.30}$$

where x_i^j is the mean of the membership function, σ_i^j is its variance. Here, the parameters of the system can be adapted using a gradient algorithm [49]:

$$z^j(n+1) = z^j(n) + \alpha \left(d(n) - F(x) \right) p^j(x), \tag{14.31}$$

$$x_i^j(n+1) = x_i^j(n) + \alpha \left(d(n) - F(x) \right) \left(z^j - F(x) \right) p_j(x) \left(\frac{x_i - x_i^j}{{\sigma_i^j}^2} \right), \tag{14.32}$$

$$\sigma_i^j(n+1) = \sigma_i^j(n) + \alpha \left(d(n) - F(x) \right) \left(z^j - F(x) \right) p_j(x) \left(\frac{\left(x_i - x_i^j \right)^2}{{\sigma_i^j}^3} \right). \tag{14.33}$$

14.5.2. Performance Evaluation. When the transmitted signal passes through of a nonlinear channel, the received signal is a nonlinear function from the last values of the transmitted symbols. The generated nonlinear distortion varies with time and place. A comparison is shown between the performance of the fuzzy algorithm and the traversal equalizer with the LMS and RLS algorithms, using a nonlinear channel [51], given by

$$x(n) = z(n) + 0.5z(n-1) - 0.9[z(n) + 0.5z(n-1)]^3. \tag{14.34}$$

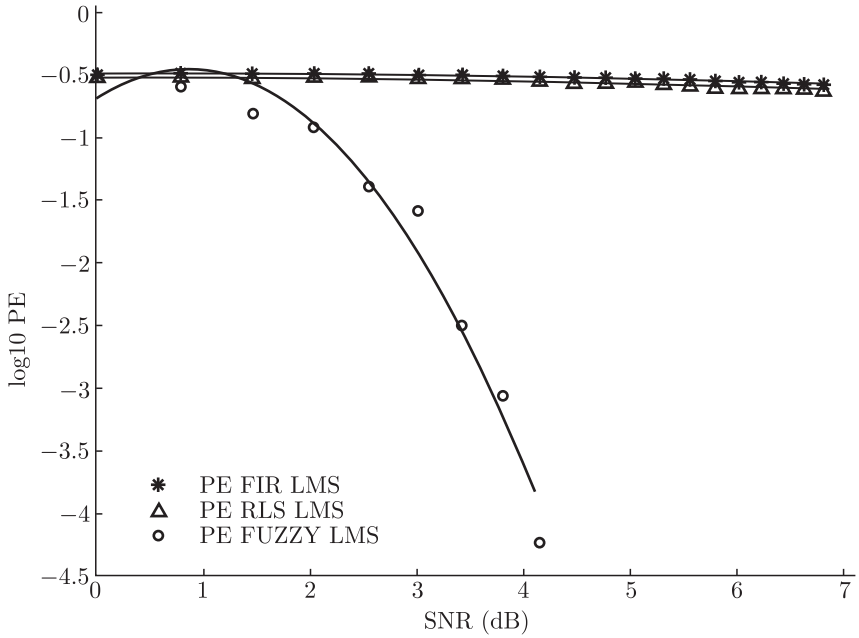


Fig. 14.25. Bit error rate obtained using the fuzzy and transversal equalizers.

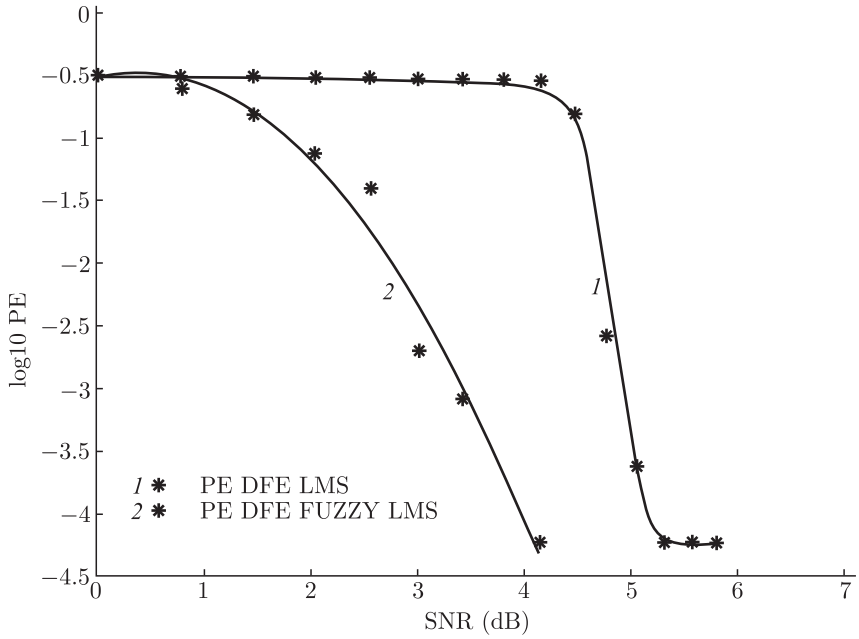


Fig. 14.26. Bit error rate obtained using the fuzzy and conventional DFE equalizers.

For the fuzzy equalizer, 25 rules were used with a sigma equal to 0.2 for all them and $\alpha = 0.01$. The transversal equalizer with the LMS and RLS adaptation algorithms have the same numbers of delays and the same convergence factors, $\alpha = 0.01$. The obtained bit error rate (BER) is shown in Fig. 14.26. From these results we can see that the fuzzy equalizer performs better than the transversal equalizer using either the RLS or LMS algorithms. The total number of samples used to obtain the results mentioned above is 20000, with 3000 samples only considered for training.

14.6. Blind Equalizer Structure

This section describes a blind detector to operate in an asynchronous DS CDMA spatial temporal array communication system subject to a frequency selective channel, which is based on the work by Sadler and Manikas [50].

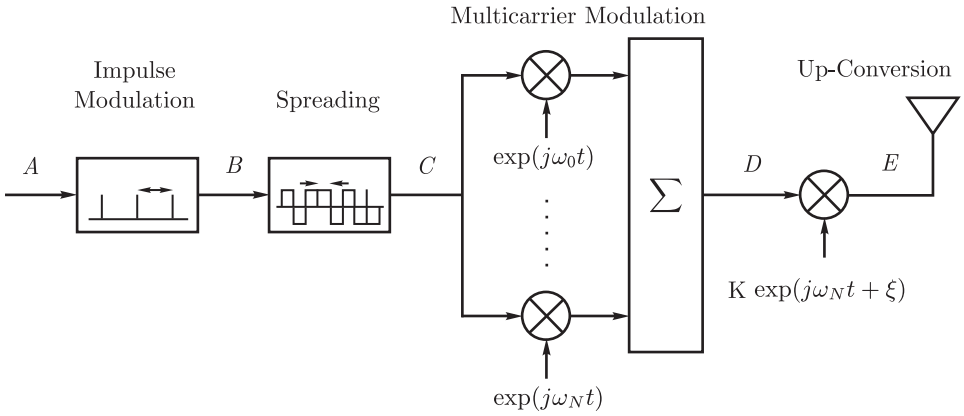


Fig. 14.27. Block diagram of DS CDMA Transmitter system.

14.6.1. Transmitter. Consider the block diagram of a particular mobile transmitter shown in Fig. 14.27. At point A, the ith user produces a sequence of complex channel symbols according to the $M-$ary modulation scheme to be employed, where M is the number of points in the signal constellation. The channel symbols are denoted by $b_i(n)$, which have a rate of $r_{cs} = r_b = \log_2(M)$ symbols per second. In this paper, quaternary phase shift keying (QPSK) is considered, where each symbol being transmitted has unit energy. The channel symbol sequence is then transformed into an impulse train at point B given by

$$b_i(t) = \sum_{n=-\infty}^{\infty} b_i(n)\delta(t - nT_{cs}), \quad nT_{cs} \leqslant t < (n+1)T_{cs}, \tag{14.35}$$

where $T_{cs} = 1/r_{cs}$ is the channel symbol period and $\delta(t)$ is the delta-function. Convolving this signal with one period of a pseudo-noise (PN) signal, $c_{PN.i}$, spreads the signal over a wider bandwidth, producing a baseband DS-CDMA signal at point C given by

$$m_i(t) = \sum_{n=-\infty}^{\infty} b_i(n)c_{PN,i}(t - nT_{cs}). \tag{14.36}$$

where $c_{PN.i}$ is a single period of the PN-signal for the ith user modeled by

$$C_{PN,i}(t) = \sum_{m=0}^{N_c-1} \alpha_i(m) p_c(t - mT_c), \quad mT_{cs} \leqslant t \leqslant (m+1)T_{cs}, \tag{14.37}$$

$\alpha_i(m)$ is the ith user's PN-sequence of length N_c, and $p_c(t)$ is a rectangular chip pulse waveform of duration T_c. Note that a short code system is being used, so the number of chips per symbol is equal to the length of the PN-sequence. The DS-CDMA signal now modulates N subcarriers, which are summed to produce the signal at point D. It is then up converted to the carrier frequency to produce the transmitted radio frequency signal at point E:

$$y_i(t) = \sum_{k=0}^{N-1} \sqrt{P_i} \exp\left(j\left(2\pi F_c t + \varsigma_i\right)\right) \exp\left(j2\pi F_k t\right) m_i(t), \tag{14.38}$$

in which P_i is the transmitted power, F_c is the carrier frequency, and ς_i is a random phase offset relative to the base station receiver.

14.6.2. Channel Model. The radio channel is assumed to be fading and multipath dispersive so that the array complex baseband channel impulse response for the kth subcarrier, jth path of the ith user is given by

$$\mathbf{c}_{ijk}(t) = \beta_{ijk} \mathbf{S}_{ijk} \delta\left(t - \tau_{ij}\right), \tag{14.39}$$

where β_{ijk} is the complex path coefficient that encompasses random phase, shifts, and fading effects. The value τ_{ij} is the path delay, which will be the same for all subcarriers for a particular path. The vector $\mathbf{S}_{ijk}$ is the array manifold vector at a frequency of $F_c + F_k$ for a specific path. In general, the parameters β_{ijk}, τ_{ij}, and $\mathbf{S}_{ijk}$ can be assumed to be independent of time for symbols transmitted during the channel coherence time. For this case, the channel is assumed to be quasi-stationary.

14.6.3. Array Receiver Front Model. At the base station the superimposed radio signals for all users, paths, and subcarriers are received through an antenna array. We consider M mobile stations with K_i paths for the ith user. After the carriers are removed, the $N \times 1$ complex received signal vector at point F of Fig. 14.28 is given by

$$\underline{x}(t) = \sum_{i=1}^{M} \sum_{j=0}^{K_i} \sum_{k=0}^{N_{sc}-1} \beta_{ijk} \underline{S}_{ijk} \times \exp\left(j2\pi F_k \left(t - \tau_{ij}\right)\right) m_i\left(t - \tau_{ij}\right) + \underline{n}(t). \tag{14.40}$$

Once this signal is discretized through a bank of samplers operating at a rate of $1/T_s$, where $T_s = T_c/qN_{sc}$ and $q \in N$ is the oversampling factor, the samples are passed through N-tapped delay lines of length $2L$. A long vector $\mathbf{X}_i(n)$ is formed at point G by concatenating the contents of the tapped delay lines of all antennas and reading the entries by every symbol period:

$$\mathbf{X}_i(n) = \left[\mathbf{X}_1(\mathrm{n})\ ,\ \mathbf{X}_2(\mathrm{n})\ \ldots\ \mathbf{X}_{\mathrm{N}}(\mathrm{n})\right]_{\cdot}^{T} \tag{14.41}$$

This multi-user space–time received signal vector contains the signals associated with the nth instant symbol. Furthermore, contributions from the previous and next data symbols are present due to the lack of synchronization. This vector considers the contributions from all subcarriers, and its derivation is explained in [50].

14.6.4. Blind Multiuser Detector. A blind adaptive multiuser detector can be implemented by introducing a bank of equalizers followed by quantizers. Proper equalizer design usually requires the knowledge (or estimation) of the channel characteristics.

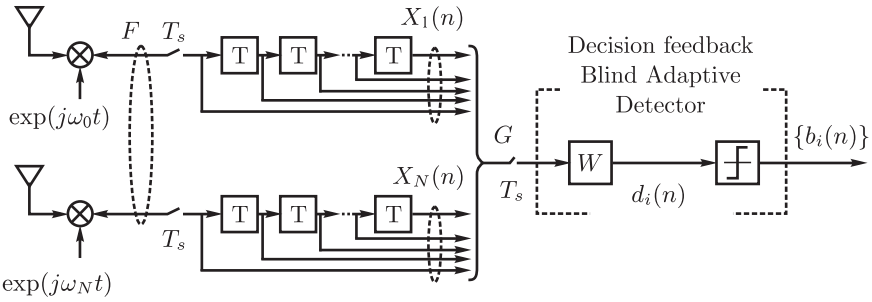


Fig. 14.28. Block diagram of blind detection structure.

Some adaptive methods, such as the least means square (LMS), require a bandwidth consuming training sequence (see [63]). Blind equalization [64], on the other hand, allows adaptation to the ISI reducing equalizer settings without the need for such training sequences or channel estimates. In this sense, the equalizer output for the ith user can be expressed as

$$d_i(n) = \sum_{l=1}^{i} w_{il} x_l(n) = \mathbf{W}_i^T(n)\mathbf{X}(n), \quad i = 1, 2, \dots K, \tag{14.42}$$

where $\mathbf{W}_i(n)$ and $\mathbf{X}(n)$ represent the equalizer coefficients and input signal vectors, respectively. The CMA [56] is a popular blind adaptation method, which penalizes the deviation of the modulus of equalizer output from some given constant. Assuming an antipodal binary value ± 1, the CMA cost function may be written as

$$J\left(d_i(n)\right) = \frac{1}{4} E\left[\left(d_i^2(n) - 1\right)^2\right], \quad i = 1, 2, \dots K. \tag{14.43}$$

The coefficients vectors $\mathbf{W}_i(n)$ are updated using the gradient of J with respect to the coefficients vector $\mathbf{W}_i$. Thus, under the assumption that $\mathbf{W}_i(n+1)$ at the nth instant is known, $\mathbf{W}_i(n+1)$ can be updated recursively as follows:

$$\mathbf{W}_i(n+1) = \mathbf{W}_i(n) - \mu \frac{\partial J\left(d_i(n)\right)}{\partial \mathbf{W}_i(n)}, \quad i = 1, 2, \dots K, \tag{14.44}$$

where μg is the step size, which controls the stability and convergence rate. Taking the derivative of $J\left(d_i(n)\right)$ with respect to $\mathbf{W}_i$ and dropping the expectation operation, we obtain

$$\mathbf{W}_i(n+1) = \mathbf{W}_i(n) - \mu \mathbf{X}(n) d_i(n) \left(d_i^2(n) - 1\right), \quad i = 1, 2, \dots K. \tag{14.45}$$

As follows from (14.44), only the signal $\mathbf{X}(n)$ contains the desired information $b_i(n)$ of ith user. Thus, the minimization of the cost function (14.43) naturally results in an optimal solution for the ith user only if the main tap coefficient w_{ii} is not equal to zero. After convergence of the blind equalizer coefficients vector, the decision for the ith user at the nth time instant can be made by taking the sign of $d_i(n)$:

$$\widehat{b}_i(n) = \operatorname{sgn}\left(d_i(n)\right). \tag{14.46}$$

The decision feedback introduces a nonlinear process, which has the potential of improving performance beyond the constraints imposed by linear detectors. However, such detectors are susceptible to error propagation in case of erroneous decisions. It is therefore important to detect users according to the received amplitude. We can extent

a blind CMA detector to include decision feedback modifying the above algorithm as follows.

Based on the noise-whitened statistics, we can assume that user 1 can be detected directly using equation (14.44) as follows:

$$\widehat{b}_1(n) = \text{sgn}\,(d_1(n))\,. \tag{14.47}$$

For user 2, since the decision for user 1 has been done, the equalization for user 2 can be realized by feeding back the decision $\widehat{b}_1$ as follows:

$$d_2(n) = \mathbf{W}_{2,i}\mathbf{X}_i(n) + w_{2,1}\widehat{b}_1(n). \tag{14.48}$$

The decision for user 2 can be obtained as

$$\widehat{b}_2(n) = \text{sgn}\,(d_i(n)) = \text{sgn}\,(d_2(n))\,. \tag{14.49}$$

Similarly, for the ith user, the equalizer output can be expressed as

$$d_i(n) = \mathbf{W}_{i,i}\mathbf{X}_i(n) + w_{i,i-1}\widehat{b}_{i-1}(n) + w_{i,i-1}\widehat{b}_{i-1}(n) + \cdots + w_{i,1}\widehat{b}_1(n), \tag{14.50}$$

$$d_i(n) = \mathbf{W}_i^T(n)\widehat{\mathbf{X}}_i(n), \quad i = 2,\, 3, \ldots, M, \tag{14.51}$$

where

$$\widehat{\mathbf{X}}_i(n) = \left[\mathbf{X}(\mathrm{n}),\ \widehat{\mathrm{b}}_{\mathrm{i}-1}(\mathrm{n})\ \widehat{\mathrm{b}}_{\mathrm{i}-\mathrm{i}}(\mathrm{n})\ \ldots\ \widehat{\mathrm{b}}_1(\mathrm{n})\right]^T \tag{14.52}$$

and

$$\mathbf{W}_i(n) = [\mathbf{W}_{\mathrm{i,i}}, \mathrm{w}_{\mathrm{i,i}-1}, \mathrm{w}_{\mathrm{i,i}-2,}\ldots, \mathrm{w}_{\mathrm{i,1}}]^T \tag{14.53}$$

are the input and coefficients equalizer vectors at time instant n. Thus, from equation (14.50) the decision for the ith user can be made by

$$\widehat{b}_i(n) = \text{sgn}\,(d_i(n))\,, \quad i = 1,\ 2,\ \ldots,\ M. \tag{14.54}$$

Finally, substituting (14.50) into (14.10)–(14.12), we obtain the decision feedback blind equalization algorithm whose coefficients vector is updated as follows:

$$\mathbf{W}_i(n+1) = \mathbf{W}_i(n) - \mu\widehat{\mathbf{X}}_i(n)d_i(n)\left(d_i^2(n) - 1\right). \tag{14.55}$$

Combining (14.45)–(14.50), we obtain the decision-feedback blind adaptive multiuser detector. The principal problem with this method is the slow convergence rate derived from its gradient-based structure. Moreover, in practical implementations, the order for equalizer filters can be extremely long. This leads serious degradation in the performance, which can make difficult its implementation under real conditions. Although (14.55) seems easy to implement, it have limitations that have motivated us to investigate a method to improve its convergence rate. Since this is a gradient based algorithm, one limitation concerns the sensitivity of convergence to the statistics of the input signal. An examination of this limitation in the nonblind case and the methods that have been devised to deal with it provide background material for the adaptive algorithm developed in next section. One alternative to solve this problem is to use an orthogonalized adaptive filtering approach to improve the performance of blind multiuser detector. The next section provides the derivation of the proposed orthogonalized blind detector.

14.6.5. Parallel Realization Form. Consider the output signal $y(n)$ of an Nth-order transversal filter, which is given by

$$d_i(n) = \mathbf{X}_F^T(n)\mathbf{H}_i, \tag{14.56}$$

where

$$\mathbf{X}_F(n) = \left[\mathbf{X}^{\mathrm{T}}(\mathrm{n}), \mathbf{X}^{\mathrm{T}}(\mathrm{n-M}), \mathbf{X}^{\mathrm{T}}(\mathrm{n-2M}), \dots\right.,$$
$$\left.\dots, \mathbf{X}^T(n-(L-2)M), \mathbf{X}^T(n-(L-1)M)\right]^T, \quad (14.57)$$

$$\mathbf{X}(n-kM) = [x(n-kM), x(n-kM-1), \dots, x(n-(k+1)M+1)]^T \quad (14.58)$$

is the input vector,

$$\mathbf{H}_i = \left[\mathbf{H}_0^{\mathrm{T}}, \mathbf{H}_1^{\mathrm{T}}, \mathbf{H}_2^{\mathrm{T}}, \dots, \mathbf{H}_{\mathrm{L-1}}^{\mathrm{T}}\right]^T \quad (14.59)$$

is the overall equalizer coefficients vector, and

$$\mathbf{H}_k = \left[h_{kM}, h_{kM+1}, h_{kM+2}, \dots, h_{(k+1)M-1}\right]^T \quad (14.60)$$

is the kth sparse filter coefficients vector. Next, substituting equations (14.57) and (14.59) into equation (14.56), we obtain

$$d_i(n) = \sum_{k=0}^{L-1} \mathbf{X}^T(n-kL)\mathbf{H}_k. \quad (14.61)$$

Next, define

$$\mathbf{H}_k = \mathbf{C}^T\mathbf{A}_k, \quad (14.62)$$

where $\mathbf{C}$ denotes the discrete cosine transform (DCT). Substituting equation (14.62) into equation (14.61), we obtain

$$d_i(n) = \sum_{k=0}^{L-1} \left(\mathbf{C}\mathrm{X}(\mathrm{n-kM})\right)^T \mathbf{A}_k = \sum_{k=0}^{L-1} \mathbf{U}^T(n-kM)\mathbf{A}_k, \quad (14.63)$$

where $\mathbf{U}^T(n-kM) = (\mathbf{C}\mathbf{X}(n-kM))^T$ is the DCT of the input vector $\mathbf{X}(n)$,

$$\mathbf{U}(n-kM) = [u_0(n-kM), u_1(n-kM), \dots,$$
$$\dots, u_3(n-kM), \dots, u_{M-1}(n-kM)]^T, \quad (14.64)$$

$$\mathbf{A}_k = \left[a_{k,1}, a_{k,2,}, \dots, a_{k,(M-1)}\right]^T. \quad (14.65)$$

Then, from equations (14.63) and (14.65) we obtain

$$d_i(n) = \sum_{k=0}^{L-1}\sum_{r=0}^{M-1} a_{k,r}u_r(n-kM). \quad (14.66)$$

In order to improve the system performance, firstly interchange the summation order as follows:

$$d_i(n) = \sum_{r=0}^{M-1}\sum_{k=0}^{L-1} a_{k,r}u_r(n-kM) \quad (14.67)$$

and define

$$\mathbf{V}_r(n) = [u_r(n), u_r(n-M), u_r(n-2M), \dots,$$
$$\dots, u_r(n-(L-2)M), u_r(n-(L-1)M)]^T, \quad (14.68)$$

$$\mathbf{W}_r = \left[a_{0,r}, a_{1,r}, a_{2,r}, \dots, a_{(L-1),r}\right]^T, \quad (14.69)$$

so that equation (14.67) takes the form

$$d_i(n) = \sum_{r=0}^{M-1} \mathbf{W}_r^T \mathbf{V}_r(n). \tag{14.70}$$

Expression (14.70) defines the output signal of the subband decomposition based filter structure proposed in [64], which also has perfect reconstruction properties without regarding the statistics of the input signal or the adaptive filter order.

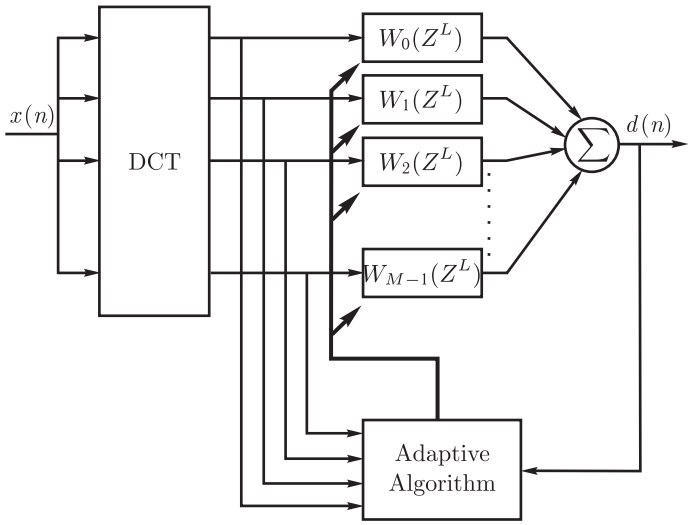


Fig. 14.29. Subband decomposition based blind detection structure.

14.6.6. Adaptation algorithm. The blind equalizer coefficients vector is updated so that the cost function given by equation (14.43) attains a minimum. Substituting equation (14.70) into (14.55) and dropping the expectation operator, we obtain

$$\mathbf{W}_{r,i}(n+1) = \mathbf{W}_{r,i}(n) - \mu \mathbf{V}_{r,i}(n) d_i(n) \left(d_i^2(n) - 1\right), \tag{14.71}$$

where $d_i(n)$ is the detected symbol,

$$\mathbf{V}_{r,i}(n) = [u_{r,i}(n), u_{r,i}(n-M), u_{r,i}(n-2M), \ldots, \\ \ldots, u_{r,i}(n-(L-2)M), u_{r,i}(n-(L-1)M)]^T \tag{14.72}$$

is the rth filter input vector corresponding to the ith user, and $u_{r,i}(n)$ is the rth DCT component of the input vector of the ith user which, from equation (14.51) is given by

$$\widehat{\mathbf{X}}_i(n) = \left[\mathbf{X}(n)\, \widehat{b}_{i-1}(n)\, \widehat{b}_{i-i}(n)\ \ldots\ \widehat{b}_1(n)\right], \tag{14.73}$$

where $\widehat{b}_i(n)$ is the ith user detected symbol, given by equation (14.51). Figure 14.28 shows the block diagram of the proposed blind detector corresponding to the ith user.

14.6.7. Simulation results. This section provides some computer simulation results obtained to evaluate the actual convergence and detection performance of the parallel blind detection algorithm described above. In order to illustrate the performance of the proposed method and compare it with the convergence performance of the

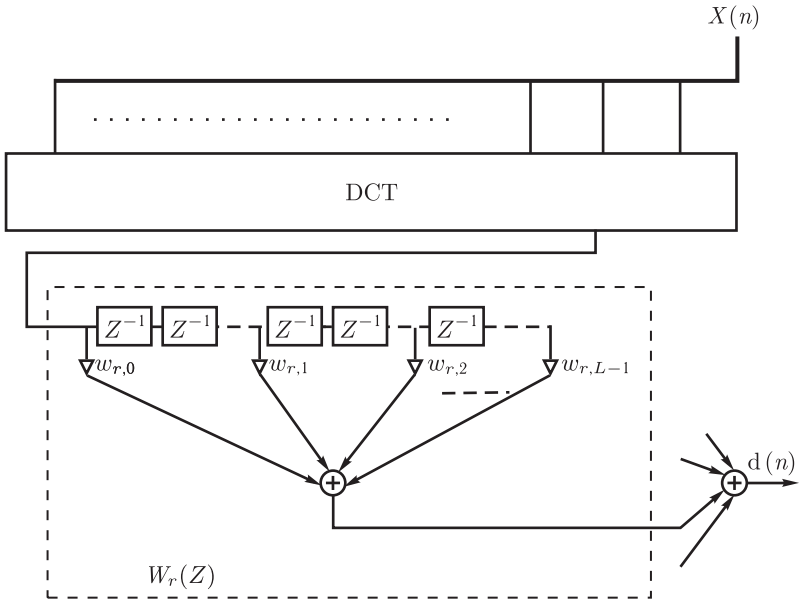


Fig. 14.30. The rth stage of blind subband decomposition in the proposed blind detector.

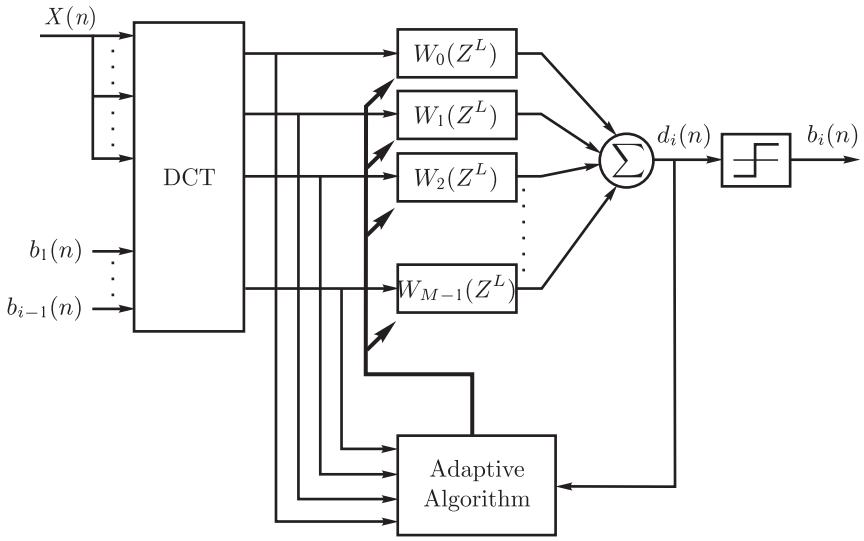


Fig. 14.31. Blind DFE based on subband decomposition approach.

conventional algorithm, we consider two different scenarios. First, the conventional blind adaptive multiuser detector (BMUD) is evaluated using an asynchronous DS-CDMA system; and next the subband decomposition based scheme (DCT-MUD) is evaluated under the same conditions to compare the performance of both cases. Two important evaluation parameters of both detectors are calculated in our analysis: the convergence performance or the mean-square error (MSE) and the bit error rate (BER). The results for these parameters are obtained for 5, 10, and 15 users in the DS-CDMA system, which

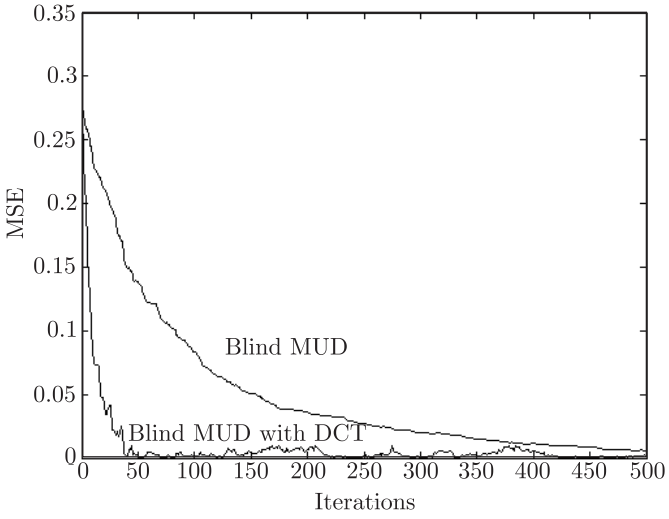


Fig. 14.32. Convergence performances of the BMUD and DCT-MUD for the 5^{th} user. The $SNR = 10$ dB, step size is 0.001, and all user energies are identical.

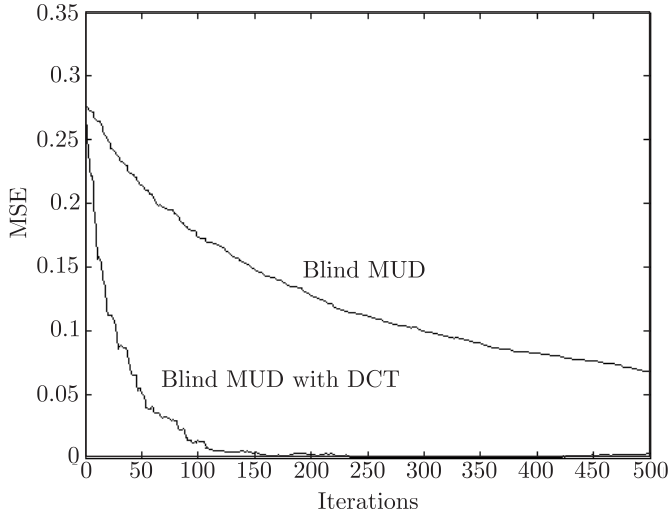


Fig. 14.33. Convergence performance of BMUD and DCT-MUD for the 10^{th} user. The $SNR = = 10$ dB, step size is 0.001, and all user energies are identical.

are shown in Figs 14.32–14.37. The convergence performance of the BMUD and the DCT-MUD for the fifth-user case, assuming that the energies of all users are identical, is shown in Fig. 14.32. In this case, the signal-to-noise ratio (SNR) is 10 dB and the step size of the blind equalizers is $\mu = 0.01$. The same evaluation results are shown for the 10th user and 15th user in Figs. 14.33 and 14.34, respectively. From these figures, we can see that the DCT-MUD achieves faster convergence performance in comparison to the BMUD. This reason is that the DCT-MUD carries out the adaptation process

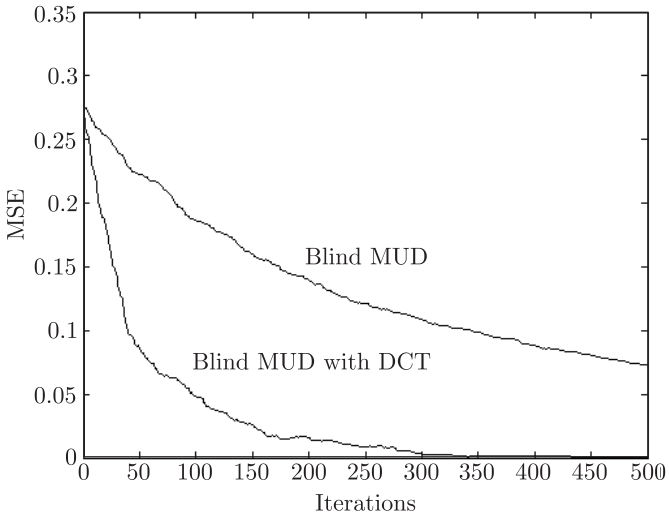


Fig. 14.34. Convergence performance of the BMUD and DCT-MUD for 15^{th} user. The $SNR =$ $= 10$ dB, step size is 0.001. and all user energies are identical.

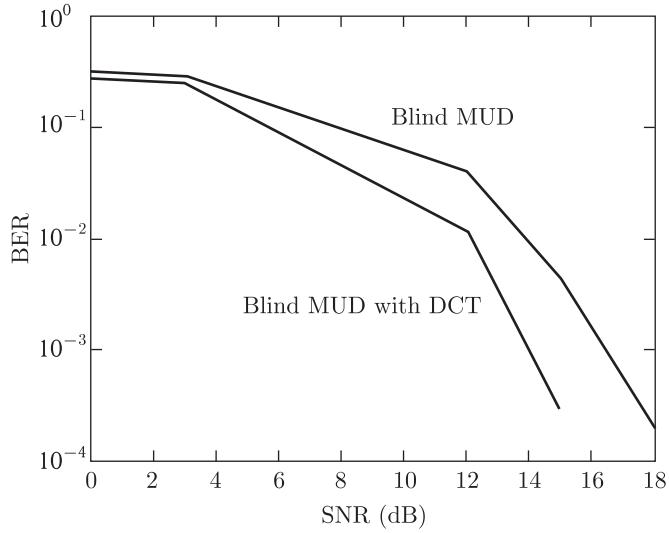


Fig. 14.35. Bit error performance of the BMUD and DCT-MUD for the 5th user.

over signal with improved characteristics derived from the use of a DCT based subband decomposition. The subband decomposition performs two important operations: firstly, it approximately decorrelate the input signal and, secondly, decompose the full-band equalizer into a bank of sparse equalizers with a reduced number of taps, allowing an easy parallel realization form. These two facts make it possible to increase the convergence rate of the gradient-search-based adaptive algorithm used to update the blind detector coefficients vector.

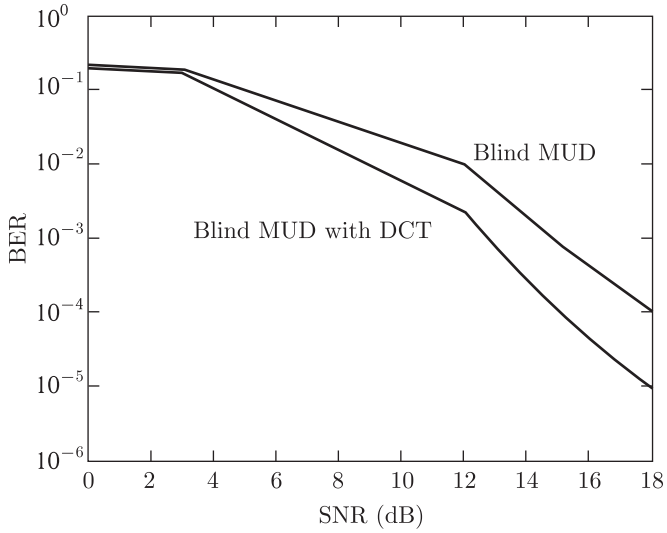


Fig. 14.36. Bit error performance of the BMUD and DCT-MUD for the 10th user.

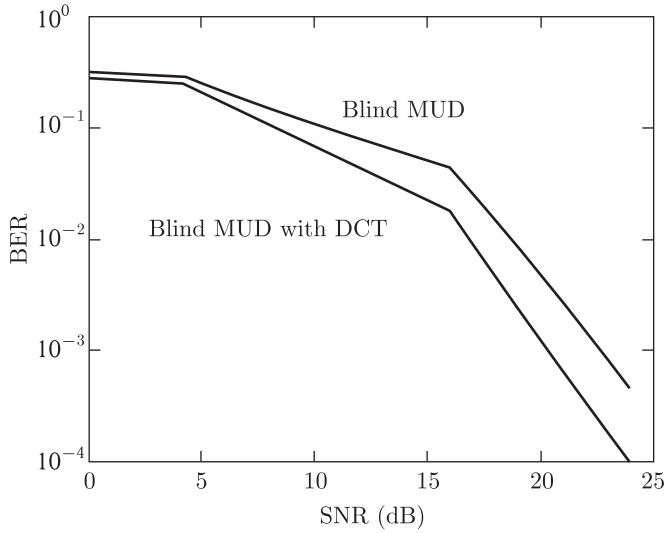


Fig. 14.37. Bit error performance of the BMUD and DCT-MUD for the 15th user.

Figures 14.35–14.37 show the bit error rate performance of DCT-MUD and the BMUD when they are required to detect the 5th, 10th, and 15th users, respectively, operating in a DS-CDMA system. In all cases, all users have identical energies. For comparison, both the BMUD and the DCT-MUD were simulated under the same conditions. The simulation results have shown that, although the performance for both detectors must be the same for the case of user 1, according to (14.47), as the user number increases, the presence of multiple access interference (MAI) is also more severe. This fact introduces a larger error rate which degrades the equalizer performance. However, in all cases the proposed equalizer provides a better performance than the conventional one. Simulation results have shown that the performance of blind

detection method is similar to that of the conventional algorithm with respect to the BER. However, the algorithm with subband decomposition provides better convergence rates, offering the same estimation error as compared with the conventional detector.

Conclusions

This chapter presents several approaches to reduce the intersymbol interference in data communication systems. Firstly, the problem of simultaneous reduction of the intersymbol and transmitter signal interferences is analyzed. Here, three different approaches are analyzed: the ISI-DFE, the NP-DFE, and the H-DFE. The computer simulation has shown that both the NP-DFE and H-DFE structures provide a little better performance than the conventional ISI-DFE under distortion conditions of ISI and for a colored signal such as a voice signal at the receiver generated for its own transmitter, due to fact that the NP-DFE structure does not propagate an erroneously detected symbol, as it happens with the ISI-DFE structure, while the ISI-DFE performs better when the additive noise is white. The H-DFE is a structure that combines the characteristics of the DFE-ISI and DFE-NP (ISI reduction and noise reduction). According to the results, it is better to adapt simultaneously the echo canceller and the equalizer.

A Hopfield ANN-based continuous time decision feedback equalizer structure is presented for equalization of time varying land mobile communications channels, in which the DFE output signal is computed in a analog discrete time way, and a continuous time Hopfield neural network is used to update the DFE structure coefficients vector. Thus, the DFE output and the coefficients vector update can be computed in less time and with less power than it is required by its digital counterparts. The performance of the proposed DFE structure was evaluated and compared with that of the conventional DFE with a RLS adaptation algorithm by computer simulations assuming three different cases: time invariant (stationary) communication channels, slow time-varying communication channels, and rapid time-varying communication channel. In all the cases, a 4-QAM modulation scheme was assumed. These results show that both the proposed and the conventional DFE perform quite similarly when it is required to equalize stationary communication channels. However, the proposed DFE structure outperforms the conventional DFE structure when it is required to equalize time-varying communication channels. The reason is that the proposed scheme is updated more frequently. Thus, potentially, it has a much faster convergence rate and a greater ability to track rapidly changing channels as well as a smaller size and lower power consumption.

A fuzzy equalizer is also presented. Computer simulations show that the neurofuzzy algorithm provides a much better performance than conventional transversal equalizers with LMS and RLS equalizers applied in nonlinear channels equalization problems. The neurofuzzy DFE algorithm shows also an improvement with respect to its equivalent DFE with RLS adaptation algorithm (nonlinear technique of equalization). In both cases, it is achieved by increasing the number of membership functions, although it will also increase its computational complexity. It may be their main disadvantage when limited computational power is available.

This chapter presented a blind decision-feedback structure based on a subband decomposition, in which, firstly, the DCT is used to approximately orthogonalize the input vector, which is subsequently inserted in a bank of parallel sparse FIR whose coefficients are updated using the CMA algorithm. This equalizer structure is based on a previously proposed blind multicarrier CDMA receiver, whose bit error rate is reduced introducing the detected bits of previous users. This approach does not require

a previous knowledge about the system. Simulation results show that the orthogonalized blind structure has better convergence performance than the conventional blind equalizer. The performance of both equalizers is improved with the introduction of a feedback stage; however, the computational complexity becomes higher when the number of users increases, because, in this situation, the size of the input and coefficients vector also increases in both the parallel and conventional equalizers. Simulation results show that the parallel equalizer provides higher convergence rates and better performance than conventional method.

References

1. *Proakis, J.G.*, Digital Communications. «Digital Signaling over a Channel with Intersymbol Interference and Additive Gaussian Noise», Chapter. 6, McGraw-Hill 1985.
2. *Sánchez-García, J.C., Nakano-Miyatake, M., and Pérez-Meana, H.M.* A Modified Hopfield Network Based Analog Decisión Feeeback Ecualizer for Land Mobile Communications. Journal of Signal Processing, 1999, vol. 3, no 5, pp. 347–356.
3. *Bastián, J.A.* Algoritmo Neurodifuso para Ecualización de Canales. chapter. 1,4 Tesis de Maestria de la Sección de Estudios de Posgrado e Investigación de ESIME del IPN, 2001.
4. *Cowan, C.F.N. and Grant, P.M.*, Adaptive Filters, Chapters 1, 3, 8, Prentice Hall 1985.
5. *Giordano, A.A. and Hsu, F.M.*, Least Square Estimation with Applications to Digital Signal Processing, Chapter 6, Wiley Interscience 1985.
6. *Gi-Hong, I. and Kyu-Min, K.*, Performance of a Hybrid Decision Feedback Equalizer Structure for CAP-Based DSL Systems. IEEE Transactions On Signal Processing. Julio 2001, vol. 49, no. 8.
7. *Haykin, S.* Adaptive Filter Theory, Prentice Hall, Englewood Cliffs, NJ, 1991.
8. *Honing, M. and Messerschmitt, D.* Adaptive filters: Structures, algorithms and applications, Kluwer Academic Publishers, Boston, MASS, 1984.
9. *Perez, H. and Amano F.* Acoustic echo cancellation using multirate techniques. IEICE Trans. on Fundamentals of Electronics, Communications and Computer Science. Nov. 1991, vol. E74, no. 11, pp. 3559–3568.
10. *Shynk J.* Frequency Domain and Multirate Adaptive Filtering. IEEE Signal Processing Magazine, Jan. 1992, vol. 9, no. 1, pp. 1437.
11. *Chao, J., Perez, H., and Tsujii, H.* A jumping Algorithm for Adaptive Filtering. Trans. of The IEICE, July 1990, vol. J73-A, pp. 1196–1209.
12. *Karni S. and Zeng G.* The analysis of the continuous time LMS algorithm. IEEE Trans. on Acoustic, Speech and Signal Processing, vol. 37, No. 4, pp. 595–597, 1989.
13. *Linder, T., Zojer, H., and Seger, B.* Fully analog LMS adaptive notch filter in BICMOS technology. IEEE Journal of Solid State Circuits, 1996, vol. 31, no. 1, pp. 61–65.
14. *Kuo, J., Harris, J., and Principe, J.* Analog Hardware Implementation of Adaptive Filter Structure. Proc. of the ICNN 1997 pp. 916–921.
15. *Shoval, A., Johns, D., and Snelgrove, M.* Comparison of DC offset effects in four LMS adaptive algorithms. IEEE Trans. on Circuit and Systems, 1995, vol. 42, no. 3, pp. 176–185.
16. *Pérez, H., Nakano, M., Nino de Rivera, L., and Vazquez, R.* Filtros Adaptivos Gamma y Filtros Adaptivos de Laguerre: Comparación Teórica y Experimental, Memoria Técnica de CONIELECOM 98, pp. 262–265, Puebla, Mexico, 1998.
17. *Espinosa, G, Díaz, J., Pérez, J., and Sánchez J.* Ecualizador adaptivo de tiempo continuo BICMOS de bajo voltaje utilizando una red neuronal de Hopfield. Revista Mexicana de Ingeniería Electrónica y Eléctrica, July 1997, vol. 2, no. 1, pp. 1–4.
18. *Nakano, M. and Pérez, H.* Analog adaptive decision feedback equalizer structure for land mobile communications, Proc. of The First Analog VLSI Workshop, pp. 77–81, Ohio USA, 1997.

19. *Nakano*, *M. and Perez*, *H.* Analog adaptive filtering based on a modified Hopfield network, Nov. 1997, vol. E80A, no. 11, pp. 2238–2245.
20. *Nino de Rivera*, *L.*, *Perez*, *H.*, *and Sanchez*, *E.* A Modular analog NLMS structure for adaptive filtering. Proc. of The First Analog VLSI Workshop, pp. 77–81, Ohio, USA, 1997.
21. *Fines*, *P. and Aghvami*, *A.H.* A New Medium and High Bit Rate 16-Ary QAM Demodulator for Land Mobile Satellite Communications. IEICE Trans. August 1991, vol. E-74, no. 8.
22. *Falconer*, *D. and Ljung*, *L.* Application of Fast Kalman to adaptive Equalization. IEEE Trans. on Communications, vol. COM-26, no. 10, pp. 1439–1446.
23. *Chen*, *S.*, *McLaughlin*, *S.*, *Mulgrew*, *B.*, *and Grant*, *P.M.* Adaptive Bayesian Decision Feedback Equalizer for Dispersive Radio Channels. IEEE Trans. on Communications, May 1995, vol. 43, no. 5.
24. *Tsatsanis*, *M.K. and. Giannakis*, *G.B.* Equalization of Rapidly Fading Channels: Self-Recovering Methods. IEEE Trans. on Communications, May 1996, vol. 44, no. 5.
25. *Mulgrew B.* Applying Radial Basis Functions. IEEE Signal Processing Magazine, March 1996, pp. 50–65.
26. *Nino-de-Rivera*, *L.*, *Perez*, *H.*, *and Sanchez*, *E.* Continuous-time normalized LMS adaptive filter structure. The Journal of Signal Processing, July 1998, vol. 2. no. 4, pp. 309–317.
27. *Sánchez*, *J.*, *Nakano*, *M.*, *and Pérez H.* A modified Hopfield network based analog decision feedback equalizer for land mobile communications. Journal of Signal Processing, Sept. 1999, vol. 3, no. 5, pp. 347–356.
28. *Wu*, *W.*, *Chen*, *R.*, *and Chang*, *S.* An Analog Architecture for Estimation of ARMA Models. IEEE Trans. Signal Processing, Sept. 1993, vol. 41, pp. 2946–2953.
29. *Kosko*, *B.* Neural Networks for Signal Processing, Prentice Hall, Englewood Cliff, NJ, 1992.
30. *Hertz*, *J.*, *Krogh*, *A.*, *and Palmer*, *R.G.* Introduction to the Theory of Neural Computation, Reading, MA, Addison-Wesley, 1994.
31. *Abe*, *S. and Gee*, *A.H.* Global Convergence of the Hopfield neural Network with Nonzero Diagonal Elements. IEEE Trans. CAS-II, Jan. 1995, vol. 42, no. 1, pp. 39–45.
32. *Widrow and Stearn*, Adaptive Signal Processing, Prentice Hall, NJ, 1985.
33. *Nino-de-Rivera*, *L.*, *Pérez*, *H.*, *and Sanchez*, *E.* A modular analog NLMS structure for adaptive filtering. The Journal of Analog Integrated Circuits and Signal Processing vol. 21, pp. 127–142, Kluwer, 1999.
34. *Perez-Meana*, *H. and Tsujii*, *S.* A system identification algorithm using orthogonal functions. IEEE Trans. on Signal Processing, 1991, vol. 39, no. 3, pp. 752–755.
35. *Sommen*, *C. and Van Vanburg C.* Efficient realization of adaptive filter using and orthogonal projection method, Proc. of The International Conference on Speech Acoustic and Signal Processing, pp. 940–943, 1989.
36. *Riegler*, *R.L. and Compton*, *R.T.* An adaptive array for interference rejection. Proc. of The IEEE, 1973, vol. 61, no. 6, pp. 748–758.
37. *Morgan*, *D. and Craig S.* Real time adaptive linear prediction using the Leas Mean Square gradient algorithm. IEEE Trans. Acoustic, Speech and Signal Processing, 1976, vol. 24, no. 6, pp. 494–507.
38. *Casco*, *F.*, *Perez Meana*, *H.*, *Nakano Miyatake*, *M.*, *and López M.* A Variable Step Size NLMS Algorithm Based on Correlation Criterion. IEICE Trans. on Fundamentals of Electronics, Communications and Computer Science, Aug. 1995, vol. E78-A, no. 8, pp. 1004–1009.
39. *Balaban*, *P. and Salz*, *J.* Dual Diversity Combining and Equalization in Digital Cellular Mobile Radio. IEEE Trans. Veh. Technol., 1991, vol. 40, no. 2, pp. 342–354.
40. *D'Aria*, *G.*, *Muratore*, *F.*, *and Palestini V.* Simulation and performance of the pan-European land mobile radio system. IEEE Trans. Veh. Technol., 1992, vol. 41, pp. 177–189.
41. *Ziegler*, *R.A. and Cioffi*, *J.M.* Estimation of Time-Varying Digital Radio Channels. IEEE Trans. Veh. Technol., May 1992, vol. 41, no. 2, pp. 134–151.
42. *Pahlavan*, *K. and Matthews*, *J.W.* Performance of adaptive Matched Filter Receivers Over Fading Multipath Channels. IEEE Trans. Comm. Dec. 1990, vol. 38, no. 12, pp. 2106–2113.

43. *Braun*, *W.R. and Dersch*, *U.* A Phisical Mobile Channel Model. IEEE Trans. Veh. Technol., May 1991, vol. 40, no. 2, pp. 472–482,
44. *Pérez*, *H. and Nakano*, *M.* A continuous time structure for filtering and prediction using Hopfield neural networks, Lectures Notes in Computer Science, Biological and Artificial Computation: From Science to Technology, Springer Verlag, pp. 1241–1250, 1997.
45. *Pérez*, *H.*, *Nakano*, *M.*, *and Nino de Rivera*, *L.* Source distortion in analog adaptive filters, Proc. of International Symposium on Information Theory and Its Applications, pp. 18–21, México, 1998.
46. *Giordano*, *A. and Hsu*, *F.M.* Least Square Estimation with Applications to Digital Signal Processing, Wiley-International, 1985.
47. *Proakis*, *J.G.* Digital Communications, Chapter 6, Digital Signaling over a Channel with Intersymbol Interference and Additive Gaussian Noise, McGraw-Hill 1985.
48. *Giordano*, *A.A. and Hsu F.M.* Applications to Digital Signal Processing, cap. 6, «Equalization», Wiley Interscience 1985.
49. *Kosko*, *B.* Fuzzy Engineering, Chapter. 2, Additive Fuzzy Systems, Prentice Hall 1997.
50. *Sarwal*, *P. and Srinath*, *M.D.* A Fuzzy Logic System for Channel Equalization. IEEE Transactions on Fuzzy Systems, May, 1995, vol. 3, no. 2, pp. 246–249.
51. *Li-Xin Wang*, *P. and Mendel*, *J.M.* Fuzzy Adaptive Filters, with Aplication to Nonlinear Channel Equalization. IEEE Transactions on Fuzzy Systems, August 1990, vol. 1, no. 3, pp. 161–170.
52. *Yang*, *Young-Sheng*, *Ch.*, *Francis H.Y.*, *Lam*, *F.K.*, *and Nguyen*, *H.* A New Fuzzy Classifier with Triangular Membership Functions, 0–7803–4122–8/97 IEEE.
53. *Sampei*, *S.* Development of Japanese Adaptive Equalizing Technology Toward High Bit Rate Data Transmission in Land Mobile Communications. IEICE Transactions, June 1991, vol. E74, no. 6, pp. 1512–1521.
54. *Bastian*, *J. Ambrosio*, *Nakano-Miyatake*, *M.*, *and Perez-Meana*, *H.* Algoritmo Neurodifuso para Ecualizacion de Canales, Proc. of The CONIELECOMP, pp. 120–123, February 2001.
55. *Kondo*, *S. and Milstein*, *L.B.* Performance of multicarrier DS CDMA systems. IEEE Trans. Commun., Feb. 1996, vol. 44, no. 2, pp. 238–246.
56. *Rapajic*, *P.B. and Vucetic*, *B.S.* Adaptive receiver structures for asynchronous CDMA systems. IEEE J. Select. Areas Commun., May 1994, vol. 12, pp. 685–697.
57. *Chen*, *D.S. and Roy*, *S.* «An adaptive multiuser receiver for CDMA systems». IEEE J. Select. Areas Commun., 1994, vol. 12, no. 5, pp. 808–816.
58. *Sadler*, *D.J. and Manikas*, *Athanassios.* Blind Reception of Multicarrier DS-CDMA Using Antenna Arrays. IEEE Trans. Wireless Commun., 2003, vol. 2, no. 6, pp. 1231–1239.
59. *Ping*, *H.*, *Tjhung*, *T.T.*, *and Rasmussen L.K.* Decision-Feedback Blind Adaptive Multiuser Detector for Synchronous CDMA System. IEEE Trans. On Vehicular Technology Commun., 2000, vol. 49, no. 1, pp. 159–166.
60. *Mangalvedhe*, *N.R. and Reed*, *J.H.* Blind adaptation algorithms for direct-sequence spread-spectrum CDMA single-user detection. Proc. of IEEE Veh. Technol. Conf., Phoenix, AZ, 1997, pp. 2133–2137.
61. *Godard D.N.* Self-recovering equalization and carrier tracking in twodimensional data communication systems. IEEE Trans. Commun., Nov. 1980, vol. COM-28, pp. 1867–1875.
62. *Zvonar Z.* Combined multiuser detection and diversity reception for wireless CDMA systems. IEEE Trans. Veh. Technol., Feb. 1996, vol. 45, no. 1, pp. 205–211.
63. *Tapia Sánchez*, *D.*, *Bustamante*, *R.*, *Pérez Meana*, *H.*, *and Nakano Miyatake*, *M.* Single Channel Active Noise Canceller Algorithm Using Discrete Cosine Transform. Journal of Signal Processing, 2005, vol. 9, no. 2, pp. 141–151.

About Authors

Victor F. Kravchenko Ph.D. (1968), Doctor of Physics and Mathematics (Dr. Sc. Phys.–Math.) (1986), Full Professor (1989), Honored Scientist of the Russian Federation (2005). He is working as Chief Researcher at the Kotel'nikov Institute of Radio Engineering and Electronics of the Russian Academy of Sciences and Professor of Bauman Moscow State Technical University. He is currently serving as Editor-in-Chief of International Journals *Electromagnetic Waves and Electronic Systems*. Dr. Kravchenko is a Member of the IEEE and the American Mathematical and Moscow Mathematical Societies. Main research activity: applied mathematics, atomic and R-functions, radio physics, signal/image processing, remote sensing, superconductivity, medical information, etc. He is the author of more than 450 scientific papers, 200 conference papers, 11 patents of the USSR, and twenty one scientific books.

Volodymyr I. Ponomaryov Ph.D. (1974), Doctor of Sciences (1981), Full Professor (1984). He is working as Professor at National Polytechnic Institute of Mexico (Mexico-city). He is currently serving as Associate Editor of the international journals: *Journal of Real Time Image Processing* (Springer) and *Uspekhi Sovremennoy Radioelectroniki* (Moscow). Dr. Ponomaryov is a Member of SPIE, IEEE, and IEICE (Japan). Also he is a member of the National System of Investigators and National Academy of Sciences of Mexico. Main research activity: signal/image processing, real-time filtering, medical sensors, remote sensing, etc. He is the author of more than 150 scientific papers, 300 conference papers, 22 patents of the USSR, Russia and Mexico, and several scientific books.

Hector M. Perez Meana M. Sci (1986), Ph.D. (1989). He is working as Professor at National Polytechnic Institute of Mexico (Mexico-city). He is currently serving as Associate Editor in journals: *Electromagnetic Waves and Electronic Systems*, *Uspekhi Sovremennoy Radioelectroniki* (Moscow), and *Cientifica* (Mexico). Dr. Perez Meana is a Member of the IEEE and IEICE (Japan). Also he is a member of the National System of Investigators and National Academy of Sciences of Mexico. Main research activity: adaptive filtering, signal/image processing, pattern recognition, etc. He is the author of more than 100 scientific papers, 200 conference papers, 8 patents of Mexico and Japan, and several chapters in scientific books.

In FIZMATLIT
the following books were published:

2006

Tatarenko, N. I., Kravchenko, V. F. **Field Emission Nanostructures and Devices on Their Basis.**

The analysis of the present state and tendencies of the development of vacuum micro- and nanoelectronics is given in this book. Physicochemical principles of the process of creating a new class of field emission nanostructures on the basis of nanoporous anodic alumina are under consideration. The results of studies of their geometrical parameters, elemental composition and emission characteristics are given. A principally novel integrated technology for fabricating nanostructural field emission microdevices and systems of their interconnections on the basis of thin films of valve metals and their anodic oxides is presented. Physical bases of modeling procedure and computation of characteristics of these microdevices are given. Their experimental and computated characteristics are demonstrated.

The book is intended for scientific and engineering researchers, postgraduates and students of senior courses of higher educational institutions specializing in the field of physical electronics, micro- and nanoelectronics.

Kravchenko, V. F., Rvachev V. L. **Algebra of Logic, Atomic Functions and Wavelets in Physical Applications.**

Methods of algebra of logic, theory of R-functions (Rvachev functions), atomic functions and wavelets are represented in the monograph. In the first two chapters the algebrological method of R-functions is described and in the third chapter – its application for calculations of physical- mechanical fields of various nature. The fourth chapter is dedicated to the theory of atomic functions for solving the present problems of radiophysics. In the fifth chapter a new class of W-system of Kravchenko-Rvachev functions was constructed and its application for the tasks of detection of short-time sign-changing and ultrawideband processes was studied.

The monograph is intended for specialists interested in the present methods of computational mathematics and its applications for solving boundary-value problems of various physical nature, digital signal and image processing, problems of the present radiophysics and electronics, mathematical modeling of physical processes and it is also aimed at students and post-graduates of higher educational institutions specializing in applied and computational mathematics, applied physics and radiophysics.

Kravchenko, V. F. **Electrodynamics of Superconductive Structures: Theory, Algorithms and Methods of Computation.**

Theoretical data of surface impedance of superconductors are represented and generalized in the monograph. Various impedance boundary conditions are considered and the boundaries of their use in boundary-value problems of electrodynamics are determined. A great number of physical models of various superconductive structures for inner and outer boundary-value problems were studied. New algorithms were obtained and methods of their computations were worked out as well.

The monograph is intended for scientific researchers, engineers engaged in the field of radiophysics and electronics, mathematical modeling problems of physical processes having place in various superconductive structures and it is also aimed at students and post-graduates of higher educational institutions specializing in applied physics and computational mathematics.

2007

DIGITAL SIGNAL AND IMAGE PROCESSING IN RADIO PHYSICAL APPLICATIONS

V.F. Kravchenko, M.A. Basarab, O.V. Goryachkin, V.K. Volosyuk, A.A. Zelenskii, A.V. Ksendzuk, B.G. Kutuza, V.V. Lukin, A.V. Totskii, V.P. Yakovlev

Edited by Honored Scientist of the Russian Federation, Doctor of Physical and Mathematical Science, Professor V.F. Kravchenko

Abstract. In the monograph, the new perspective trends of digital signal processing with their applications to radio physics and radio engineering problems are considered. The monograph consists of five chapters. The new methods of signal approximation on the basis of the Whittaker–Kotelnikov–Shannon theorem with using the compactly supported functions, including a new class of atomic ones, are considered in the first chapter. The second chapter is devoted to the use of the bispectral analysis in digital signal processing. The third chapter is devoted to multiposition radar systems with synthetic antenna aperture. Some aspects of digital signal processing in radars and SAR are surveyed in the fourth chapter. The fifth chapter is devoted to the development of new mathematical methods, algorithms, and their applications to the blind signal processing. The sixth chapter consists of two parts. Its first part is devoted to the theory of R-functions and atomic functions (AF) with reference to the description of random shaped locuses. In the second part, the construction of Kravchenko 2D weight functions (windows) on the basis of irregular shape areas for digital multidimensional signal and image processing, resulting from the first part, is under consideration. Chapter 7 consists of tree parts. In the first part the foundation of wavelet transform, generalize Kotelnikov series and Levitan polynomial, which constructing on basis of atomic functions (AF) are studied. In the second part the new class analytical wavelets Kravchenko-Kotelnikov and Kravchenko-Levitan is considered. The third part is consecrate to formation of quality functional. The numerical experiment and physical analysis illustrated effectiveness of this approach for digital ultra wideband (UWB) signal processing.

The monograph is recommended for scientists, students and post-graduates specializing in radio physics, radio engineering, computational mathematics, and computational physics.

2008

STATISTICAL THEORY OF RADIOTECHNICAL SYSTEMS OF REMOTE SENSING AND RADAR

Volosyuk,V.K. and Kravchenko,V.F.

Edited by Honored Scientist of the Russian Federation,
Doctor of Physical and Mathematical Science, Professor V.F. Kravchenko

Fundamental characteristics of scattered and proper radiothermal emission of natural environment are considered. Analysis of different electrodynamic models of surfaces and surrounding atmosphere is given. Models of radiotechnical signals and their statistical characteristics in the area of registration by antenna systems are developed.

Bases of the theory of the optimal spatio-temporal processing of scattered and proper radiothermal emission fields are presented. Principles of appropriate construction and algorithmic support of contemporary active, passive and complex active-passive radiotechnical systems of remote sensing and also interpretation of experimental data are formulated.

The results of algorithms development of optimal and quasi-optimal measuring of electrophysical parameters of surfaces and atmosphere with active, passive and complex active- passive remote sensing are given. Extreme values of measuring errors for these parameters are also evaluated. Recommendations for selection of such conditions for remote sensing aerospace experiments, which provide the smallest measuring errors, are developed.

Solutions for a number of mapping and target selection problems using classic and modified techniques of antenna aperture synthesis are presented. Peculiarities of applying atomic functions and Kravchenko-Rvachev weighting windows for image processing and subsurface sensing are considered.

For researchers, engineers, postgraduate students and students of senior courses engaged in remote sensing and radiolocation problems.

METHODS OF MODELING AND DIGITAL SIGNAL PROCESSING IN GYROSCOPY

Basarab, M.A., Kravchenko, V.F., and Matveev, V.A.

Main principles of operating of modern sensitive elements for inertial navigation are considered, including solid-state wave gyro (hemisphere resonator gyro, HRG) and micromechanical gyros. In problems of modeling physical processes, error identification, and information processing, numerical approximation techniques on the base of the theories of R-functions and atomic functions, as well as artificial intelligence approaches (genetic algorithms and neural networks), are used.

2009

ELECTROMAGNETIC WAVES DIFFRACTION ON UNCLOSED CONICAL STRUCTURES

V.A. Doroshenko, V.F. Kravchenko

Edited by Honored Scientist of the Russian Federation, Doctor of Phys.-Math. Science, Professor *V. F. Kravchenko*

Rigorous methods and approaches for solving problems of electromagnetic wave diffraction on inhomogeneous conical structures are proposed and developed in the book. Physical processes in these structures are theoretically studied by using new mathematical models for analyzing nonstationary electromagnetic field formation by objects with angular parameters and geometrical singularities (tips, edges). The base for these investigations is a mathematical tool that is constructed and developed for solving a nonstationary electromagnetic wave diffraction on a complex conical structures with longitudinal slots. This tool is based on using new proposed methods and approaches without initial constrains on geometrical sizes of the structure. By using the considered problem solutions obtained by these methods and approaches one can study features, general rules and effects caused by electromagnetic wave diffraction on complex unclosed cones.

COMPUTING METHODS IN THE MODERN RADIO PHYSICS

V. F. Kravchenko, O. S. Labunko, A. M. Lerer, G. P. Sinyavsky

Edited by the Honoured Scientist of the Russian Federation,
Dr. of Phys.-Math. Sci., Professor *V. F. Kravchenko*

In this monograph, the basic ideas and the methods connected with working out of numerical models in boundary value problems of electrodynamics of SHF range and also digital signal and image processing are presented. It consists of four chapters. In the first chapter solving of various kinds of the space-time integral equations (IE) for planar and quasiplanar structures are considered. The second chapter is devoted to investigation of diffraction of electromagnetic impulses on two- and three-dimensional metal and dielectric bodies, on slits and holes in ideally conducting screen. In the third chapter the constructions of new orthogonal Kravchenko wavelets on the basis of atomic functions (AF) are considered. The new method of numerical differentiation based on WA-systems of functions is offered and proved. Application of new wavelets to model problems of digital signal processing (DSP) and numerical differentiation has shown their efficiency. The fourth chapter is devoted to construction of new designs of orthogonal wavelets on the basis of AF $h_a(x)$. Generalisation of ambiguity function (AmF) on time and frequency on the basis of family AF with reference to digital signal processing in antenna systems is proved. The new class of analytical Kravchenko–Rvachev wavelets (AKR-wavelets) is investigated. Advantages of the AKR-wavelets as compared with following wavelets: Daubechies, Morlet, Shannon and the others for the analysis of ultrawideband (UWB) signals are shown. By means of complex quality functional of the choice an optimum wavelet basis for each concrete model of UWB signal is made. The new approach based on combinations of AF with classical spectral kernels is considered. It is shown that the received new designs of the spectral kernels used by transfer and reception of the information have advantages as compared with ones in problems of spectral analysis of UWB signals.

The monograph is recommended for scientists, students and post-graduates specializing in radio physics, radio engineering, computational mathematics, and computational physics.