Chapter 2
Adaptive Filtering

2.1 Introduction

Adaptive linear filters are linear dynamical system with variable or adaptive structure and parameters. They have the property to modify the values of their parameters, i.e. their transfer function, during the processing of the input signal, in order to generate signal at the output which is without undesired components, degradation, noise and interference signals. The goal of the adaptation is to adjust the characteristics of the filter through an interaction with the environment in order to reach the desired values, [4, 5]. The operation of adaptive filters is based on the estimation of the statistical properties of the signal in its environment, while modifying the value of its parameters in order to minimize a certain criterion function. The criterion function may be determined in a number of ways, depending on the particular purpose of the adaptive filter, but usually it is a function of some reference signal. The reference signal may be defined as the desired response of the adaptive filter, and in that case the role of the adaptive algorithm is to adjust the parameters of the adaptive filter in such a way to minimize the error signal, which represents the difference between the signal at the output of the adaptive filter and the reference signal.

The basic processes included in adaptive filtering are digital filtering and adaptation or adjustment, i.e. the estimation of parameters (coefficients) of the filter. The choice of the filter structure and the criterion function used during the adaptation process, have the crucial influence to the characteristics of the adaptive filter as a whole.

2.2 Structures of Digital Filters

There are several types of digital filters usable in the design of adaptive filters; most often they are linear discrete systems, although there is a significant application of nonlinear adaptive filters, among which a large group are neural networks [22]. Digital filters are often categorized either in dependence on the duration of

B. Kovačević et al., *Adaptive Digital Filters*,
DOI: 10.1007/978-3-642-33561-7_2,

the impulse response or on their structure. Two basic types are the IIR (Infinite Impulse Response) and the FIR (Finite Impulse Response) filters [2–5].

The impulse response of a digital filter is its output signal, i.e. its response, obtained when a unit impulse (Kronecker delta-impulse) is brought to its input

$$\delta(k) = \begin{cases} 1, & \text{for } k = 0 \\ 0, & \text{for } k \neq 0. \end{cases}$$

The digital filters, in which the duration of the impulse filter response is, theoretically, infinite, are called the infinite impulse response filters or the IIR filters. Contrary to them, the filters with finite impulse response are denoted as the FIR filters.

When categorizing based on the structure, one starts from the output signal. The filter output may be a function of the actual and the previous samples of the output signal, $x(k)$, as well as the previous values of the output signal, $y(k)$. If the actual value of the output is a function of the previous values of the output, then there must be a feedback or a recursive mechanism in the structure, and thus such filters are denoted as recursive. Contrary to them, if the output is only a function of the input signal, we speak about non-recursive filters. In order to obtain an infinite impulse response, one has to use some kind of a recursive filter, and this is the reason why the terms of the IIR and the recursive filter are sometimes used interchangeably. Similar to that, finite duration of the impulse response is obtained with non-recursive structures, and thus the terms of the FIR and the non-recursive digital filters are used as synonyms.

2.2.1 Filters with Infinite Impulse Response (IIR Filters)

The most general structure of a digital filter is the recursive filter shown in Fig. 2.1. It contains a direct branch with multipliers, with their values determined by the parameters $b_i,\ i = 0,\ 1,\ \ldots, M$ and a return branch with multipliers, determined by the parameters $a_i,\ i = 1,\ 2,\ \ldots, N$. The actual value of the output signal $\hat{y}(k)$ is determined by a linear combination of the following weighted variables: the actual and the previous values of the input signal samples $x(k-i),\ i = 0,\ 1,\ \ldots, M$, as well as the previous values of the output signal samples $\hat{y}(k-i),\ i = 1,\ 2,\ \ldots,\ N$.

In digital signal processing literature the block diagram 2.1 is denoted as the direct realization [1–3]. This structure represents the design of the filter transfer function with zeroes and poles, such that the position of the poles is determined by the values of the parameters a_i, while the position of the zeroes is determined by the parameters b_i [3]. The number of the poles and zeroes is determined by the number of the delay elements (z^{-1}). This structure has a very large memory, theoretically infinite, and thus it is denoted as the filter with infinite impulse response (IIR). In other words, the impulse response of the filter, which represents

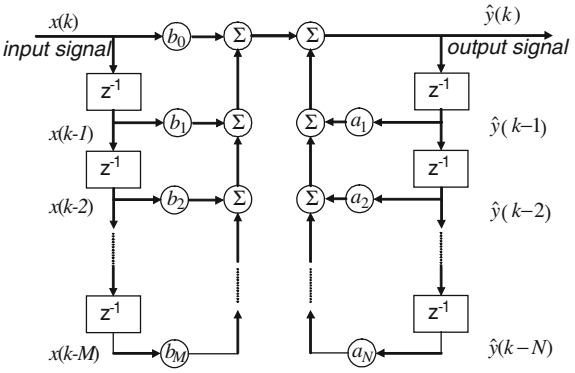

Fig. 2.1 Structure of an IIR recursive filter

the output signal of the filter, $\hat{y}(k)$, to impulse excitation $x(k) = \delta(k)$, will last infinitely long, i.e. the signal $\hat{y}(k)$ will decrease to zero only after infinite time (the transient response of the filter will last infinitely before the filter output reaches zero value in a steady or equilibrium state).

According to Fig. 2.1, the output signal, $\hat{y}(k)$, of the IIR filter is given as a linear difference equation

$$\hat{y}(k) = \sum_{i=0}^{M} b_i(k)x(k-i) + \sum_{j=1}^{N} a_j(k)\hat{y}(k-j), \tag{2.1}$$

where $b_i(k); i = 0, 1, 2 \ldots M$ are the parameters of the IIR filter in the direct branch in a k-th discrete moment, and $a_j(k)$; $j = 1, 2 \ldots N$ are the parameters of the IIR filter in the return branch in the given k-th moment. In general case $M \leq N$, thus N represents the order of the filter (N represents the minimal number of the delay elements z^{-1} necessary to physically implement the relation (2.1) using digital electronic components [2]).

Besides calculating the filter output, $\hat{y}(k)$, the adaptive IIR filter should update $M+N$ parameters a_i and b_i, in order to optimize a previously defined criterion function. The parameter update is a more complex task in the case of IIR filters than for FIR filters due to two reasons. The first reason is that an IIR filter may become unstable during optimization if the filter poles become positioned outside the stable region (unit circle in the complex z-plane), and the other is that the criterion function to be optimized, generally taken, may have a multitude of local minima, with a possible consequence that the optimization process ends in some of the local minima, instead in the global one. Contrary to them, the criterion functions of the FIR filters (MSE for instance, as will be shown later) usually have only one minimum, which also represents the global minimum. In spite of the quoted difficulties, the recursive adaptive filters find significant practical applications in the control (regulation) systems, especially if the system to be controlled is recursive. In these applications the adaptive IIR filters with several parameters may have better properties than FIR filters with several thousands of parameters [23].

Relation (2.1) can be also written in the polynomial form

$$\hat{y}(k) = \mathbf{B}_k\left(z^{-1}\right)x(k) + \left(1 - \mathbf{A}_k\left(z^{-1}\right)\right)\hat{y}(k), \tag{2.2}$$

where the polynomials are

$$\mathbf{B}_k\left(z^{-1}\right) = \sum_{i=0}^{M} b_i(k)z^{-i}; \;\; \mathbf{A}_k\left(z^{-1}\right) = 1 - \sum_{j=1}^{N} a_j(k)z^{-j}. \tag{2.3}$$

Let us note that in the adopted notation the symbol z^{-1} has the meaning of unit delay, i.e. that $z^{-1}x(k) = x(k-1)$ and $z^{-1}\hat{y}(k) = \hat{y}(k-1)$. The polynomial relation (2.2) can be also written in an alternative form

$$\mathbf{A}_k\left(z^{-1}\right)\hat{y}(k) = \mathbf{B}_k\left(z^{-1}\right)x(k). \tag{2.4}$$

If we assume that the filter parameters do not change with time, i.e. that they do not change with the time index k, the filter transfer function, $G(z)$, is obtained as the ratio of z complex forms of the output, $\hat{y}(k)$, and the input, $x(k)$, signal, assuming that the initial conditions in the difference equations are zero, i.e. that the values of the samples of the corresponding signals in (2.1) are equal to zero for negative values of the time index k, i.e.

$$G(z) = \frac{\mathcal{Z}[\hat{y}(k)]}{\mathcal{Z}[x(k)]} = \frac{\hat{Y}(z)}{X(z)} = \frac{\mathbf{B}(z^{-1})}{\mathbf{A}(z^{-1})}, \tag{2.5}$$

where according to (2.1) the polynomials are

$$\mathbf{B}\left(z^{-1}\right) = \sum_{i=0}^{M} b_i z^{-i}; \quad \mathbf{A}\left(z^{-1}\right) = 1 - \sum_{j=1}^{N} a_j z^{-j}; \; N \geq M, \tag{2.6}$$

while N represents the filter order. In the above relation (2.5) z is a complex variable, and the roots of the equation $\mathbf{B}(z^{-1}) = 0$ determine the zeroes of the filter, while the roots of the equation $\mathbf{A}(z^{-1}) = 0$ define the poles of the filter (zeroes and poles of the filter are also denoted in literature as critical frequencies, and their position in the z-plane is denoted as the critical frequency spectrum. The dynamical response of the filter to the input signal is dominantly dependent on the position of the poles in the z-plane and the necessary and sufficient condition of the filter stability is that the poles are located within the unit circle, $|z| = 1$. Generally speaking, a filter is stable if the filter output in equilibrium or steady state, occurring after transient process ceased, is dictated solely by the excitation signal [3].

2.2.2 Filters with Finite Impulse Response (FIR Filters)

One of the ways to overcome the lack of potential instability in an IIR digital filter is to design a filter with zeroes only, which in comparison to a recursive IIR

structure has only the direct branch or a non-recursive structure. The memory of such filters is limited, i.e. their impulse response is equal to zero outside some limited time interval, and because of that they are denoted as the filters with finite impulse response (FIR). In other words, a transient process in such a system, which is initiated immediately after bringing excitation and which lasts until the output signal assumes a stationary value, i.e. the system enters equilibrium or steady state, has a finite duration. A good property of the FIR filters is that their phase characteristic is completely linear (the transfer function $G(z)$ for $z = \exp(j\omega T)$, $-\pi \leq \omega T \leq +\pi$, where j denotes imaginary zero, is denoted as amplitude-phase frequency (spectral) characteristic; at that $|G(\exp(j\omega T))|$ is called the amplitude, and $\arg\{G(\exp(j\omega T))\}$ is the phase frequency (spectral) characteristic). Another good property is that they have unconditional stability, and because of that they represent the basis of the systems for adaptive signal processing [4, 5]. Two basic structures for realization of FIR filters are the transversal structure and the lattice structure. Figure 2.2 shows the structure of a transversal FIR filter. The filter contains adders, delay elements (z^{-1}) and multipliers, defined by the parameters $\{b_i,\ i = 0, 1, 2, \ldots, M\}$.

The number of the delay elements, *M,* denotes the order of the filter and the duration of its impulse response. The output signal of the filter, $\hat{y}(k)$, is determined by the values of the parameters $\{b_i\}$ and it represents a linear combination of the actual and the previous samples of the input signal, $x(k)$ These parameters are the object of estimation in an adaptive process, i.e. they vary with time index *k.* In this manner, according to Fig. 2.2, the filter output signal is defined by the linear difference equation

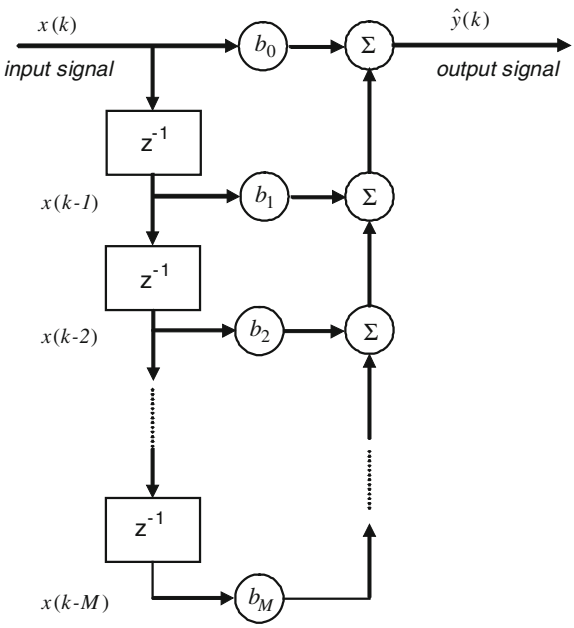

Fig. 2.2 Structure of a transversal FIR recursive filter

$$\hat{y}(k) = \sum_{i=0}^{M} b_i(k)x(k-i). \tag{2.7}$$

If the delay operator is introduced, z^{-1}, i.e. $z^{-1}x(k) = x(k-1)$, the above relation can be written in the polynomial form

$$\hat{y}(k) = \mathbf{B}_k\left(z^{-1}\right)x(k), \tag{2.8}$$

where the polynomial is

$$\mathbf{B}_k\left(z^{-1}\right) = \sum_{i=0}^{M} b_i(k)z^{-i}, \tag{2.9}$$

and M represents the filter order. In the case of a stationary or time-invariant system in which the parameters b_i are constant and do not depend on the time index k, the filter transfer function is defined as the ratio between z-complex forms of the output and the excitation signal (it is assumed that the values of the signal sample are equal to zero for negative values of the time index k):

$$G(z) = \frac{\mathcal{Z}[\hat{y}(k)]}{\mathcal{Z}[x(k)]} = \frac{\hat{Y}(z)}{X(z)} = \frac{z^M\mathbf{B}(z^{-1})}{z^M}. \tag{2.10}$$

The roots of the polynomial equation $z^M\mathbf{B}(z^{-1}) = 0$ determine the filter zeroes, while according to the expression (2.10) it is concluded that the filter has a pole $z = 0$ with a multiplicity M (the roots of the equation $z^M = 0$ are the poles of the system). Since these poles are located within the unit circle $|z| = 1$ within the plane of the complex variable z, the FIR filter represents a system with unconditional stability. Because of that fact, it is customary in literature to say that the FIR filter transfer function has only zeroes, while it is said for the transfer function of an IIR filter that it has both zeroes and poles (zeroes are the roots of the polynomial in the numerator, while poles are the roots of the polynomial in the denominator of the rational function representing the filter transfer function). Specifically, if the excitation $x(k)$ is a unit impulse, i.e. $x(0) = \delta(0) = 1$ and $x(k) = \delta(k) = 0$ za $k \neq 0$, according to (2.7) it is concluded that $\hat{y}(0) = b_0$, $\hat{y}(1) = b_1$, …, $\hat{y}(M) = b_M$, $\hat{y}(k) = 0$ for $k > M$, i.e. the impulse response of the filter will last $M + 1$ samples, while the coefficients b_i denote the values of the samples of the filter impulse response in the corresponding discrete and equidistant moments of signal sampling $t_i = iT$, $i = 0, 1, 2, \ldots, M$, where T is the sampling or discretization period.

2.3 Criterion Function for the Estimation of FIR Filter Parameters

The concepts of optimal linear estimation represent the basis for the analysis and synthesis of adaptive filters [7–10]. The problem of adaptive filtering encompasses two estimation procedures, the estimation of the desired output signal from the

filter and the estimation of the filter coefficients necessary to achieve the desired goal. The definition of these estimators depends on the choice of the criterion function, which defines the quality of the estimation based on the difference between the estimator input, $\hat{y}(k)$, and the reference or the desired output signal, i.e. the response $y(k)$.

If we denote the column vector of input data with a length $M+1$ as $\mathbf{X}(k)$

$$\mathbf{X}(k) = [\,x(k) \quad x(k-1) \quad x(k-2) \quad \ldots \quad x(k-M)\,]^T, \tag{2.11}$$

and the column vector of the estimated parameters or filter : coefficients in the k-th discrete moment of signal sampling as $\hat{\mathbf{H}}(k)$

$$\hat{\mathbf{H}}(k) = [\,\hat{b}_0(k) \quad \hat{b}_1(k) \quad \hat{b}_2(k) \quad \ldots \quad \hat{b}_M(k)\,]^T, \tag{2.12}$$

where k denotes the actual discrete time moment, and T is the matrix operation of transposing, then the signal at the FIR filter output, $\hat{y}(k)$, may be defined in the form of a linear regression equation, i.e. as a scalar product of the corresponding vectors

$$\hat{y}(k) = \mathbf{X}(k)^T \hat{\mathbf{H}}(k), \tag{2.13}$$

i.e.

$$\hat{y}(k) = \hat{\mathbf{H}}(k)^T \mathbf{X}(k). \tag{2.14}$$

While designing the optimum solution, the filter is optimized according to the corresponding criterion function or the performance index. The filter is determined by the parameter vector $\hat{\mathbf{H}}(k)$, so that the optimization problem reduces to the choice of parameters which will minimize the chosen criterion function. The choice of the criterion function is a complex problem and most often depends on the particular application of the filter [9, 16].

2.3.1 Mean Square Error (Risk) Criterion: MSE Criterion

One often meets in practice a criterion function defined as the mean square error, MSE, which represents the averaged (expected) value of the squared difference between the reference signal, $y(k)$, and the estimated actual value of the output signal, $\hat{y}(k)$. The goal is to minimize the mean square value of the error signal, which in the ideal case has the consequence that the statistical mean error value tends to zero, and that the filter output signal is as close as possible to the desired reference signal.

The error signal (Fig. 2.3) is defined according to (2.13) in the following manner

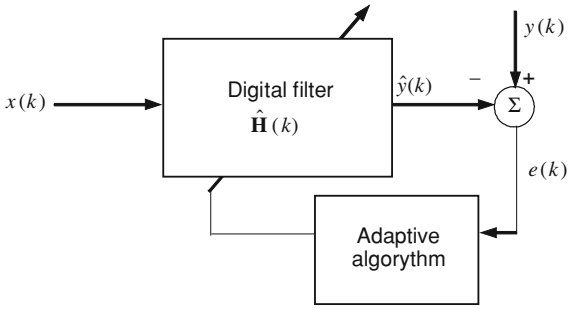

Fig. 2.3 Structure of an adaptive digital filter. The error signal $e(k)$ appears as the difference between the reference signal $y(k)$ and the actual filter output $\hat{y}(k)$, and the adaptive algorithm generates in each step k the parameter vector $\hat{\mathbf{H}}(k)$, as well as the estimation of the unknown parameters $\mathbf{H}(k)$

$$e(k) = y(k) - \hat{y}(k) = y(k) - \mathbf{X}(k)^T \hat{\mathbf{H}}(k). \tag{2.15}$$

Assuming that $e(k)$, $y(k)$ and $x(k)$ are stationary random series (the statistical properties of these signals do not change with time) and that the elements of the vector $\hat{\mathbf{H}}(k)$ are constant, the criterion function J is defined as

$$\begin{aligned} MSE \hat{=} J &= E\left[e(k)^2\right] = E\left[\left(y(k) - \mathbf{X}(k)^T \widehat{\mathbf{H}}\right)^2\right] \\ &= E\left[y^2(k) + \widehat{\mathbf{H}}^T \mathbf{X}(k)\mathbf{X}(k)^T \widehat{\mathbf{H}} - 2y(k)\mathbf{X}(k)^T \widehat{\mathbf{H}}\right], \end{aligned} \tag{2.16}$$

where $E[\cdot]$ denotes the mathematical expectation with regard to the random variables x and y. The nature of the criterion function (2.16) is probabilistic and in such an environment all signals are taken as realizations of stochastic processes, which points out to the fact that it is necessary for the design of the filter to know the suitable statistical indicators regarding the signals under consideration, i.e. the aggregate function of the probability density for the random variables x and y.

Since the mathematical expectation is a linear operator, i.e. the mathematical expectation of a sum is equal to the sum of mathematical expectations, and the mathematical expectation of a product is equal to the product of mathematical expectations only for statistically independent variables [7, 9, 10], it follows

$$E\left[e(k)^2\right] = E\left[y(k)^2\right] + \hat{\mathbf{H}}^T E\left[\mathbf{X}(k)\mathbf{X}(k)^T\right]\hat{\mathbf{H}} - 2E\left[y(k)\mathbf{X}(k)^T\right]\hat{\mathbf{H}}. \tag{2.17}$$

The index k is omitted from the vector $\hat{\mathbf{H}}(k)$ because of the assumption about its constant value in the current consideration. If

$$\mathbf{R} = E\left[\mathbf{X}(k)\mathbf{X}(k)^T\right]$$

denotes the auto-correlation matrix of the input signal

$$\mathbf{R} = E\begin{bmatrix} x(k)^2 & x(k)x(k-1) & x(k)x(k-2) & \dots & x(k)x(k-M) \\ x(k-1)x(k) & x(k-1)^2 & x(k-1)x(k-2) & \dots & x(k-1)x(k-M) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x(k-M)x(k) & x(k-M)x(k-1) & x(k-M)x(k-2) & \dots & x(k-M)^2 \end{bmatrix} \tag{2.18}$$

and

$$\mathbf{D} = E\left[y(k)\mathbf{X}^T(k)\right]$$

denote the cross-correlation vector of the input and the reference signal

$$\mathbf{D} = E[\,y(k)x(k) \quad y(k)x(k-1) \quad y(k)x(k-2) \quad \dots \quad y(k)x(k-M)\,]^T, \tag{2.19}$$

it follows that the mean square error or statistical mean value of the error signal is

$$J = E\left[y(k)^2\right] + \hat{\mathbf{H}}^T\mathbf{R}\hat{\mathbf{H}} - 2\mathbf{D}^T\hat{\mathbf{H}}. \tag{2.20}$$

2.3.2 Minimization of the Criterion of Mean Square Error (Risk)

In a general case, if the number of the coefficients in an adaptive process to be estimated is equal to M, then (2.20) represents a surface in the M-dimensional parametric space. The adaptation process represents the process of searching for a point on that surface which corresponds to the minimal value of the MSE in (2.20), or to the optimal value of the parameter vector $\hat{\mathbf{H}}$.

To determine the global minimum of the criterion function one most often uses some of the algorithms with random search. An important property of the adaptive systems implemented in the form of FIR filters is that their criterion function represents a second-order surface, i.e. a square function of $\hat{\mathbf{H}}$. Here the MSE criterion function has only one, global minimum. In that case one can use much more powerful, deterministic methods of minimization of the criterion function, based on the use of gradients or some estimation of the gradients, instead of the stochastic methods which are used in the case when the criterion function has more local minima, as is the case in the IIR systems [12, 24, 25].

If there is only one parameter, the MSE is described by a parabola (Fig. 2.4); for two parameters, the MSE is a paraboloid (Fig. 2.5) and in the general case when there is a larger number of parameters, i.e. when M is larger than 2, the surface is described by a hiper-paraboloid. Since the MSE is by definition a positive value, the criterion function represents a concave surface, i.e. it spreads in the direction of increasing MSE.

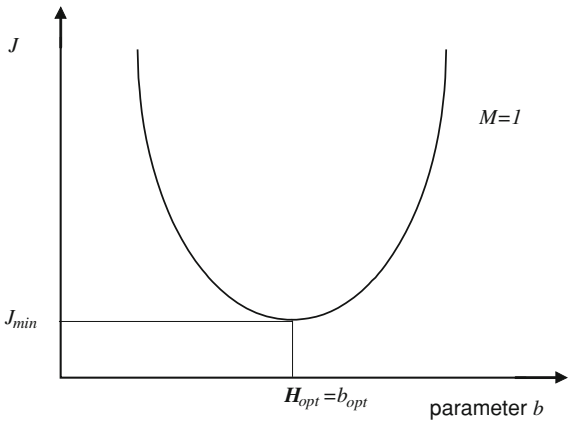

Fig. 2.4 The form of MSE for the case when M = 1. The criterion function is represented by a parabola, and the parameter vector contains only a single parameter, b

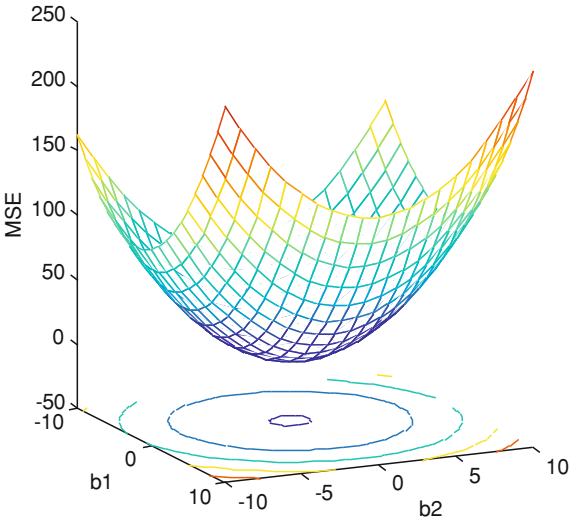

Fig. 2.5 The form of MSE for the case when M = 2. The contours for constant values of the MSE are represented by the ellipses at the bottom of the graph

The determination of the minimum of the criterion function can be done using the criterion method [12, 24, 25]. Namely, the MSE gradient is a vector that is always directed towards the fastest increment of the criterion function and with a value equal to the slope of the tangent to the criterion function. In the point of the minimum of the criterion function the slope is zero, so it is necessary to determine the gradient of the criterion function and equal it to zero in order to obtain the optimum values of the parameters minimizing the criterion function. The gradient of the criterion function J, denoted as ∇J or only ∇, is obtained by differentiating the expression (2.20) with regard to $\hat{\mathbf{H}}$,

$$\nabla = \frac{\partial J}{\partial \hat{\mathbf{H}}} = \left[\frac{\partial J}{\partial b_0} \quad \frac{\partial J}{\partial b_1} \quad \frac{\partial J}{\partial b_2} \quad \cdots \quad \frac{\partial J}{\partial b_M} \right]^T = 2\mathbf{R}\hat{\mathbf{H}} - 2\mathbf{D}. \tag{2.21}$$

If (2.21) is made equal to zero, we obtain the Wiener-Hopf equation, and by solving it we obtain the optimal solution for the parameter vector

$$\nabla = 2\mathbf{R}\hat{\mathbf{H}} - 2\mathbf{D} = 0 \tag{2.22}$$

$$\mathbf{H}_{opt} = \mathbf{R}^{-1}\mathbf{D}. \tag{2.23}$$

$\mathbf{H}_{opt}$ represents the vector of optimal values of the FIR filter parameters, i.e. those values of the parameter $J_{\min}$. According to (2.20) and (2.23), one obtains

$$\begin{aligned} J_{\min} &= E\left[y^2(k)\right] + \mathbf{H}_{opt}^T \mathbf{R}\mathbf{H}_{opt} - 2\mathbf{D}^T\mathbf{H}_{opt} \\ &= E\left[y^2(k)\right] - \mathbf{H}_{opt}^T \mathbf{R}\mathbf{H}_{opt}. \end{aligned} \tag{2.24}$$

Starting from expressions (2.20) and (2.23), the criterion function may be represented as

$$\begin{aligned} J &= E\left[y^2(k)\right] + \hat{\mathbf{H}}^T \mathbf{R}\hat{\mathbf{H}} - 2\mathbf{H}_{opt}^T \mathbf{R}\hat{\mathbf{H}} \\ &= E\left[y^2(k)\right] - \mathbf{H}_{opt}^T \mathbf{R}\hat{\mathbf{H}} + \left(\hat{\mathbf{H}} - \mathbf{H}_{opt}\right)^T \mathbf{R}\hat{\mathbf{H}} \\ &= E\left[y^2(k)\right] - \hat{\mathbf{H}}^T \mathbf{R}\mathbf{H}_{opt} + \left(\hat{\mathbf{H}} - \mathbf{H}_{opt}\right)^T \mathbf{R}\hat{\mathbf{H}} \end{aligned} \tag{2.25}$$

i.e., if one introduces (2.24), it follows

$$J = J_{\min} + \left(\hat{\mathbf{H}} - \mathbf{H}_{opt}\right)^T \mathbf{R}\left(\hat{\mathbf{H}} - \mathbf{H}_{opt}\right). \tag{2.26}$$

It is obvious from (2.26) that there is a quadratic dependence of J on $\hat{\mathbf{H}}$ and that this function reaches its minimum for $\hat{\mathbf{H}} = \mathbf{H}_{opt}$.

It should be mentioned that other criterion functions may be utilized besides MSE, for instance the (mean) absolute value of the estimated errors, higher order moments, etc.; however, such a choice, contrary to the MSE, leads to nonlinear optimization problems [9, 10, 16]. Namely, the complexity of their application and analysis is fundamentally increased, but nonlinear criterion functions nevertheless have an important role in some applications [9, 16].

In most practical cases the appearance of the criterion function is not known, and its analytical description is also not known. From (2.23) it follows that to determine $J_{\min}$ it is necessary to know the statistical properties of the input and the reference signal, i.e. the values of the correlation matrix $\mathbf{R}$ and the correlation vector $\mathbf{D}$. Most often one knows only the measurement sequences of the mentioned signals, and their statistical properties can be obtained only by estimation, based on experimental data. The values of the points on the surface defining the criterion function may be measured or estimated by averaging the MSE in time, in the sense of the approximation of the mathematical expectation by the appropriate arithmetic means. The problem of the determination of the optimal values for the filter

parameters reduces to defining an adequate numerical procedure or algorithm able to describe the curve or, in a general case, the surface determined by the criterion function, as well as to determine its minimum. The values of the parameters defining the minimum of the criterion function represent the optimal vector $\mathbf{H}_{opt}$, which is often also denoted as the "accurate" values of the parameters.

The majority of the adaptive algorithms is based on the standard iterative procedures for the solution of the minimization problems in real time. To clarify the properties of the usual adaptive algorithms for the minimization of the criterion function, we will consider two basic numerical methods for iterative minimization of the criterion function: the Newton's method and the steepest descent method. The both methods are used for the estimation of the gradient, ∇, for the determination of the minimum of the criterion function instead of the accurate value of the gradient, which is not even known in the general case [12, 24, 25].

2.3.2.1 Newton's Method

By multiplying (2.21) with ½ $\mathbf{R}^{-1}$ we obtain

$$\frac{1}{2}\mathbf{R}^{-1}\nabla = \hat{\mathbf{H}} - \mathbf{R}^{-1}\mathbf{D}. \tag{2.27}$$

By combining (2.23) and (2.27) it follows

$$\mathbf{H}_{opt} = \hat{\mathbf{H}} - \frac{1}{2}\mathbf{R}^{-1}\nabla. \tag{2.28}$$

Equation (2.28) represents the Newton's method for the determination of the root of the vector equation obtained by making the gradient of the criterion function equal to zero (the necessary condition of the minimum of the adopted criterion). Knowing the value of $\hat{\mathbf{H}}$ in any moment of time, together with the $\mathbf{R}$ and the corresponding gradient ∇, one can determine the optimal solution $\mathbf{H}_{opt}$ in just a single step. In practical situations, however, the available information are insufficient to perform a single-step adaptation. The value of the correlation matrix of the input signal, $\mathbf{R}$, changes with time under nonstationary conditions and, in the best case, can be only estimated, similar to the unknown value of the criterion function gradient ∇ which must be estimated in each iteration. In order to reduce the effect of "noisy" or fluctuating values of these estimations, one modifies (2.28) on order to reach the algorithm which updates the parameter vector $\hat{\mathbf{H}}$ in small increments and converges to $\mathbf{H}_{opt}$ after a number of iterations. In this manner, starting from (2.28), one reaches the Newton's method in an iterative (recursive) form [12, 24, 25]

$$\hat{\mathbf{H}}(k+1) = \hat{\mathbf{H}}(k) - \frac{1}{2}\mathbf{R}^{-1}\nabla(k), \qquad \{k = 0, 1, 2, \ldots\} \tag{2.29}$$

where the index k with the gradient of the criterion function denotes that it is estimated in each iteration according to (2.21). The expression (2.29) can be generalized by introducing a constant μ, i.e. a dimensionless variable determining the convergence speed of the iterative process

$$\hat{\mathbf{H}}(k) = \hat{\mathbf{H}}(k-1) - \mu \mathbf{R}^{-1} \nabla(k-1). \tag{2.30}$$

According to (2.21) it follows that $\nabla(k-1) = 2\mathbf{R}\hat{\mathbf{H}}(k-1) - 2\mathbf{D}$, and thus according to (2.30) one obtains

$$\hat{\mathbf{H}}(k) = (1-2\mu)\hat{\mathbf{H}}(k-1) + 2\mu \mathbf{R}^{-1}\mathbf{D}. \tag{2.31}$$

Arranging further (2.31), and taking into account (2.23), one can write

$$\begin{aligned}
&\hat{\mathbf{H}}(k) = (1-2\mu)\hat{\mathbf{H}}(k-1) + 2\mu \mathbf{H}_{opt} \\
&\hat{\mathbf{H}}(k) = (1-2\mu)^2 \hat{\mathbf{H}}(k-2) + 2\mu[1 + (1-2\mu)]\mathbf{H}_{opt} \\
&\vdots \\
&\hat{\mathbf{H}}(k) = (1-2\mu)^k \hat{\mathbf{H}}(0) + 2\mu \mathbf{H}_{opt} \sum_{i=0}^{k-1} (1-2\mu)^i.
\end{aligned} \tag{2.32}$$

The vector $\hat{\mathbf{H}}$ obviously converges to the optimal value of $\mathbf{H}_{opt}$ only in the case when the condition is fulfilled that the geometric series $\sum_{i=0}^{k-1} (1-2\mu)^i$ is convergent, i.e.

$$|1-2\mu| < 1, \tag{2.33}$$

that is

$$0 < \mu < 1, \tag{2.34}$$

and in that case

$$\hat{\mathbf{H}}(k) = (1-2\mu)^k \hat{\mathbf{H}}(0) + \mathbf{H}_{opt}\left[1 - (1-2\mu)^k\right]. \tag{2.35}$$

From (2.35) it follows that the final solution can be reached in one step for $\mu = 0.5$, but only under the condition that one knows the accurate values of the inverse correlation matrix of the input signal, $\mathbf{R}^{-1}$, and the gradient of the criterion function, ∇, i.e. the cross-correlation vector $\mathbf{D}$. In the case when $\mathbf{R}^{-1}$ and ∇ are estimated, one usually utilizes values $\mu \ll 1$, typically smaller than 0.01, to overcome the problems appearing because of the error introduced by the estimation of the unknown variables $\mathbf{R}$ and ∇.

Newton's method is fundamentally important from the mathematical point of view, however it is very demanding in practical applications because of the need to estimate $\mathbf{R}$ and ∇ in each step. It is the method of gradient search, a consequence of which is that all elements of the vector $\hat{\mathbf{H}}$ change in each iteration, with the goal

to determine the optimum values of the parameters. These changes are always toward the minimum of the gradient function, but, as (2.30) shows, not necessarily in the direction of the gradient itself.

As mentioned, the main problem with the Newton's algorithm is its application under the conditions when one does not know the value of the inverse correlation matrix of the input signal and the value of the gradient of the criterion function, i.e. the cross-correlation of the input and the reference signal. Regretfully, it is a common case in practice [6]. In that case one most often assumes that the non-diagonal elements of the correlation matrix are equal to zero. The methods based on this assumption bear a common name of the steepest descent method and we consider them in the further text.

2.3.2.2 Steepest Descent Method

The steepest descent method is an optimization technique utilizing the gradient of the criterion function to determine its minimum. This method, contrary to Newton's method, in each iteration updates the values of the vector $\hat{\mathbf{H}}$, only in the direction of negative value of the gradient. Since the gradient represents the direction of the fastest increment of the criterion function, the movement in the direction of negative gradient should ensure the fastest approach to the minimum of the criterion function, which is why this method obtained its name.

According to its definition, the steepest descent method can be described in the following manner [12, 24, 25]

$$\hat{\mathbf{H}}(k+1) = \hat{\mathbf{H}}(k) + \beta(-\nabla(k)). \tag{2.36}$$

The steepest descent method starts from some initial value $\hat{\mathbf{H}}(0)$. The estimation in the next step $\hat{\mathbf{H}}(k+1)$ is equal to the current estimation $\hat{\mathbf{H}}(k)$ corrected by the value in the direction opposite to that of the direction of the fastest increment of the function, i.e. of the gradient, in the point $\hat{\mathbf{H}}(k)$. The last term in Eq. (2.36) represents the estimated gradient of the criterion function in the k-th iteration. The scalar parameter β is the convergence factor determining the size of the correction step and influences the stability and the adaptation speed of the algorithm. The dimension of this factor is equal to the reciprocal value of the dimension of the input signal power.

The graphical presentation of this method for $M = 1$ is given in Fig. 2.6. It can be shown that the convergence conditions are satisfied for [6]

$$0 < \beta < \frac{1}{\lambda_{\max}}, \tag{2.37}$$

where $\lambda_{\max}$ is the largest eigenvalue of the correlation matrix of the input signal $\mathbf{R}$, which depends on the input signal power, i.e. on the mean expected value of the squared amplitude of the input signal.

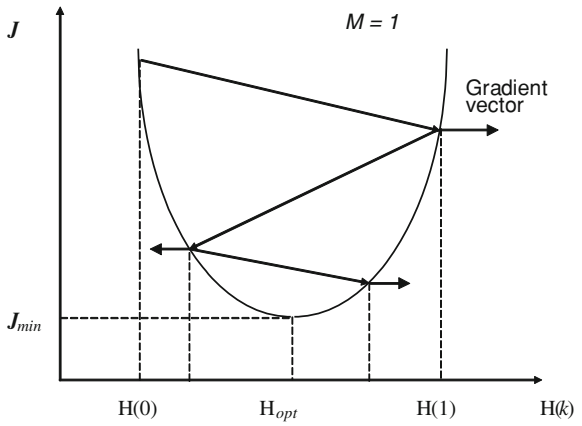

Fig. 2.6 Graphical presentation of the steepest descent method

When comparing Eqs. (2.36) and (2.29), one should note that in the case of Newton's method the information about the gradient is corrected with the value of the inverse correlation matrix of the input signal, $\mathbf{R}^{-1}$, and with the scalar parameter μ.

This means that in this method the direction of the criterion function search is corrected to keep it always toward the minimum of the criterion function, while in the steepest descent method this direction coincides with the fastest increase (decrease) of the function. The two quoted direction may not coincide in a general case, and the search path of the criterion function in the application of Newton's method is shorter, which suggests that the optimization process is faster compared to the steepest descent method (Fig. 2.7). This advantage stems from the fact that Newton's method utilizes much more information about the criterion function in comparison to the steepest descent method. Also, compared to the steepest descent method, Newton's algorithm is much more complex, since it requires the calculation or the estimation of the inverse correlation input matrix in each iteration. However, under real circumstances, in the presence of noise while estimating the gradient and the input data correlation matrix, it may happen that the steepest descent method converges much more slowly toward the minimum of the MSE in comparison to Newton's method or that, for the sake of speed, converges into a larger value of the MSE criterion.

2.4 Adaptive Algorithms for the Estimation of Parameters of FIR Filters

Adaptive digital filters, generally taken, consist of two separate units: the digital filter, with a structure determined to achieve desired processing (the structure is known with an accuracy to the unknown parameter vector) and the adaptive

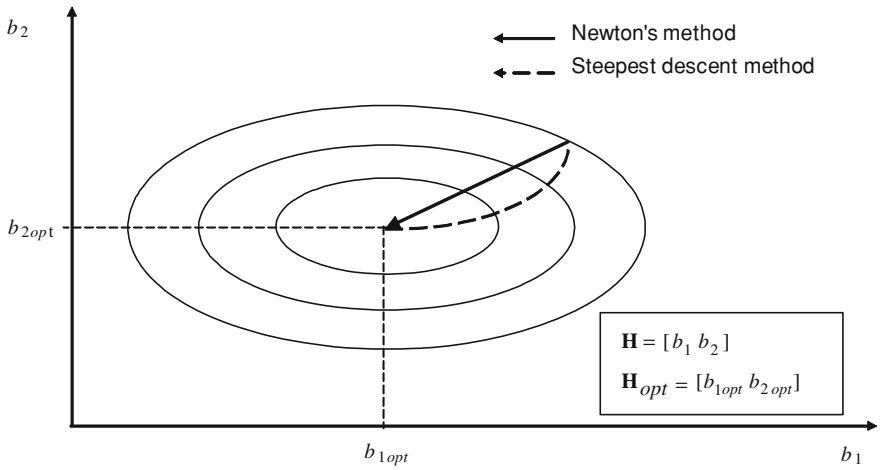

Fig. 2.7 Directions of the determination of the minimum of the criterion function for the steepest descent method and for the Newton's method

algorithm for the update of filter parameters, with a goal to ensure their fastest possible convergence to the optimum parameters from the point of view of the adopted criterion. According to this, it is possible to implement a larger number of combinations of filter structures and adaptive algorithms for parameter estimation. Most of the adaptive algorithms represent modifications of the standard iterative procedures for the solution of the problem of minimization of criterion function in real time. Two important parameters determining the choice of the adaptive algorithm are the adaptation speed and the expected accuracy of the parameter estimation after the adaptation is finished. In a general case, there is a discrepancy between these two requirements. For a given class of adaptive algorithms an increase of adaptation speed will decrease the accuracy of the estimated parameters and vice versa. In this section we consider two basic algorithms for parameter estimation for the FIR adaptive filters: the Least Mean Square (LMS) and the Recursive Least Square (RLS) algorithms. The LMS algorithm has a relatively large importance in applications where it is necessary to minimize the computational complexity, while the RLS algorithm is popular in the fields of system identification (the determination of the system model based on experimental data on input and output signals) and time series analysis (experimentally recorded signal samples) [24, 25].

2.4.1 Least Mean Square (LMS) Algorithm

LMS (Least Mean Square) algorithm belongs to the method of steepest descent, but it utilizes a special estimation of the gradient, i.e. it takes the actual value of

the squared error signal $e^2(k)$ instead of the mathematical expectation MSE in (2.16). The criterion function in LMS algorithm is thus defined as [4]

$$J(k) = e^2(k). \tag{2.38}$$

Based on the values of $\hat{\mathbf{H}}(k)$ and $e(k)$ defined in (2.12) and (2.15) the estimation of gradient is obtained as

$$\hat{\nabla}(k) = \frac{\partial e^2(k)}{\partial \hat{\mathbf{H}}} = 2e(k)\frac{\partial e(k)}{\partial \hat{\mathbf{H}}} = 2e(k)\begin{bmatrix} \frac{\partial e(k)}{\partial b_0} \\ \vdots \\ \frac{\partial e(k)}{\partial b_M} \end{bmatrix} = -2e(k)\mathbf{X}(k), \tag{2.39}$$

i.e. the current estimation of the gradient is the product of the vector of input signals in k-th iteration and the corresponding error signal. This represents the basis of the simplicity of the LMS algorithm, since only a single multiplication operation per each parameter is necessary for the estimation of the gradient.

Starting from the general form of the steepest descent method (2.36) and (2.39), one may define the LMS algorithm as

$$\hat{\mathbf{H}}(k+1) = \hat{\mathbf{H}}(k) - \beta\hat{\nabla}(k), \tag{2.40}$$

$$\hat{\mathbf{H}}(k+1) = \hat{\mathbf{H}}(k) + 2\beta e(k)\mathbf{X}(k), \quad k = 0,\ 1,\ 2,\ \ldots, \tag{2.41}$$

where $\hat{\nabla}(k)$ represents the estimation of the gradient (2.39), and β is a scalar parameter influencing the adaptation speed, the stability of the adaptive algorithm and the error value after the adaptation process is finished. Since the change of the value of the parameter vector is based on the estimation of the gradient without averaging (only the actual value of the error signal is used, i.e. one its realization) and not on its real value, one may expect that the adaptive process will be noisy, i.e. that the estimation of the parameters will fluctuate (the estimation of the parameters represents random variables with a corresponding variance).

It can be shown that (2.39) represents an unbiased estimate of the gradient for the case when the values of the parameters are constant

$$\begin{aligned} E\left[\hat{\nabla}(k)\right] &= -2E[e(k)\mathbf{X}(k)] \\ &= -2E\left[y(k)\mathbf{X}(k) - \mathbf{X}(k)\mathbf{X}^T(k)\hat{\mathbf{H}}\right] \\ &= 2\mathbf{R}\hat{\mathbf{H}} - 2\mathbf{D} = \nabla(k). \end{aligned} \tag{2.42}$$

Since the mathematical expectation of the gradient estimation is equal to the accurate value of the gradient, such an estimation of the gradient is unbiased. Because the estimation of the gradient is unbiased, i.e. the mean or expected value of the gradient estimation (2.42) is equal to its accurate value (2.21), the LMS algorithm can be classified as a steepest descent method, but with a limitation that the gradient is estimated in each iteration, while the values of the parameter vector $\hat{\mathbf{H}}$ are updated after several iterations, when one can claim that the estimation

$\hat{\nabla}(k)$, formed as the arithmetic means of the previously calculated estimated gradients with a goal to approximately determine the mathematical expectation of such an estimation of the gradient, describes well the accurate value $\nabla(k)$. Bearing in mind that the parameter vector $\hat{\mathbf{H}}$ in (2.40) is practically updated in each iteration ($k = 0, 1, \ldots$), it is necessary to limit the value of the scalar parameter β according to (2.37), in order to ensure the convergence of the parameter vector $\hat{\mathbf{H}}$ to $\mathbf{H}_{opt}$ [6].

As a consequence of updating of the parameter vector $\hat{\mathbf{H}}$ in each iteration, according to insufficiently accurate estimation of the gradient, the adaptive process is noisy, i.e. it does not follow the steepest descent line toward $\mathbf{H}_{opt}$. This noise decreases in time with the advance of the adaptive process, since near $\mathbf{H}_{opt}$ the value of the gradient is small and the correction term in (2.40) is also small, so that the estimations of the parameters are close to their previous values.

From (2.41) it is obvious that the algorithm is simple for implementation, because it does not require the operations of squaring, averaging or differentiating. In the LMS algorithm each parameter of the filter is updated in such a manner that one adds weighted value of error signal to its actual value

$$b_i(k+1) = b_i(k) + 2\beta e(k)x(k-i), \quad i = 0, 1, \ldots, M; \quad k = 0, 1, \ldots \tag{2.43}$$

The error signal, $e(k)$, is common for all coefficients, while the weight factor is $2\beta x(k-i)$ and it is proportional to the values met in a k-th moment in the i-th delay section of the FIR filter. To calculate $2\beta e(k)x(k)$ one needs $(M+1)$ arithmetic operations (multiplication and addition). Because of that, each step of the algorithm requires $(2M+1)$ operations, which makes this algorithm convenient for the real time application [26].

Generally taken, for larger values of β one reaches greater convergence speeds, but the estimation error is also larger, while for smaller β one also has smaller asymptotic error of the parameter estimation. Also, according to (2.37) it can be seen that the value of β is limited by $\lambda_{\max}$, i.e. by the input signal power $x(k)$. In order to overcome this problem, one may modify the expression (2.43) so that the correction factor is normalized with regard to the input signal power

$$b_i(k+1) = b_i(k) + \frac{\alpha[e(k)x(k-i)]}{\sum_{j=0}^{M} x^2(k-j)}. \tag{2.44}$$

The expression (2.44) represents the Normalized Least Mean Square Algorithm – NLMS, and α is a constant which may have a value within the range $0 < \alpha < 2$, [27]

The simplicity and easy implementation make the LMS algorithm very attractive for many practical applications. Its main deficiency regards its convergence properties which are slow and depend on the characteristics of the input signal. The LMS algorithm has only a single variable, the parameter β, whose change influences the convergence properties and which has a limited range of possible values, according to (2.37).

There is a number of algorithms in literature with better convergence properties than the LMS algorithm, but this was achieved through an increase of the computational complexity of the algorithms. One of these is the Least Squares (LS) algorithm.

2.4.2 Least Squares Algorithm (LS Algorithm)

Let us consider the structure of a digital FIR filter shown in Fig. 2.8 which is known with an accuracy up to an unknown set of parameters $b_i,\ i = 0,\ 1,\ \ldots,\ M$.

The error signal, $e(k)$, for this case can be defined as

$$e(k) = y(k) - x(k)b_0 - x(k-1)b_1 - \ \ldots\ - x(k-M)b_M. \tag{2.45}$$

The same as above, the vector of unknown parameters in the k-th moment of sampling is denoted by $\hat{\mathbf{H}}(k)$ and defined as

$$\hat{\mathbf{H}}^T(k) = [\, b_0(k) \quad b_1(k) \quad b_2(k) \quad \ldots \quad b_M(k)\,]. \tag{2.46}$$

The least squares method is based on the criterion according to which the estimation of parameters is optimal if the sum of error squares is minimal. Thus the criterion function for the LS algorithm is defined by [6]

$$J(k) = \frac{1}{2}\sum_{i=0}^{k} e^2(i). \tag{2.47}$$

Let us note that expression (2.47) for the LS criterion represents an approximation of the expression (2.16) for the MSE criterion in which the mathematical expectation is replaced by the corresponding sum. In the LMS criterion (2.38) this sum contains only a single term, the square of the actual error signal.

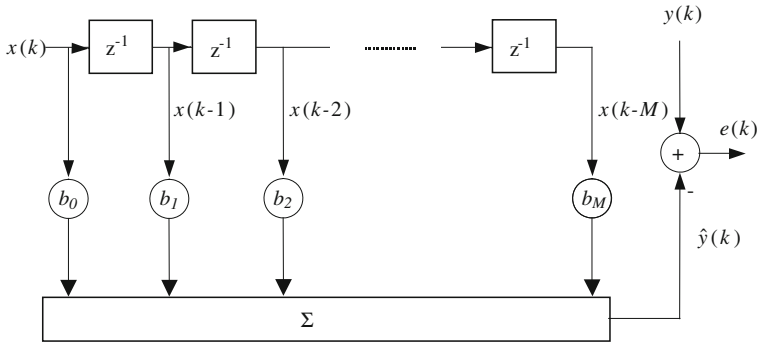

Fig. 2.8 Direct realization of FIR filter

If the expression (2.45) is written in matrix form, using the whole data package $\{e(i),\ i = 0, 1, \ldots, k\}$, one obtains

$$\begin{bmatrix} e(0) \\ e(1) \\ e(2) \\ \vdots \\ e(k) \end{bmatrix} = \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(k) \end{bmatrix} - \begin{bmatrix} x(0) & 0 & 0 & \cdots & 0 \\ x(1) & x(0) & 0 & \cdots & 0 \\ x(2) & x(1) & x(0) & \cdots & 0 \\ & & \ddots & & \\ x(k) & x(k-1) & x(k-2) & & x(k-M) \end{bmatrix} \begin{bmatrix} b_0(k) \\ b_1(k) \\ b_2(k) \\ \vdots \\ b_M(k) \end{bmatrix} \tag{2.48}$$

or in matrix notation

$$\mathbf{e}(k) = \mathbf{y}(k) - \mathbf{Z}(k)\hat{\mathbf{H}}(k), \tag{2.49}$$

where $\mathbf{e}(k)$ represents the error vector, $\mathbf{y}(k)$ is the reference signal vector, and $\mathbf{Z}(k)$ is the input matrix. When forming this equation it was adopted that $y(k) = 0$ for $k \leq 0$ (causal signal). The criterion function given by (2.38) can be expressed using the vector $\mathbf{e}(k)$, given as (2.49), in the form of a scalar product

$$J(k) = \frac{1}{2}\mathbf{e}^T(k)\mathbf{e}(k), \tag{2.50}$$

or in expanded form

$$J(k) = \frac{1}{2}\left[\mathbf{y}^T(k)\mathbf{y}(k) - \mathbf{y}^T(k)\mathbf{Z}(k)\hat{\mathbf{H}}(k) - \hat{\mathbf{H}}^T(k)\mathbf{Z}^T(k)\mathbf{y}(k) + \hat{\mathbf{H}}^T(k)\mathbf{Z}^T(k)\mathbf{Z}(k)\hat{\mathbf{H}}(k)\right]. \tag{2.51}$$

By differentiating the criterion (2.51) over the parameter vector $\hat{\mathbf{H}}$ one obtains

$$\frac{\partial J(k)}{\partial \hat{\mathbf{H}}(k)} = -\mathbf{Z}^T(k)\mathbf{y}(k) + \mathbf{Z}^T(k)\mathbf{Z}(k). \tag{2.52}$$

The vector minimizing the criterion function (2.50) is obtained by making the expression (2.52) equal to zero, i.e.

$$\hat{\mathbf{H}}(k) = \left[Z^T(k)\mathbf{Z}(k)\right]^{-1}\mathbf{Z}^T(k)\mathbf{y}(k). \tag{2.53}$$

When deriving the expression (2.52) the following rules for the differentiation of scalar over vector were used [7]

$$\frac{\partial \mathbf{y}^T\mathbf{x}}{\partial \mathbf{x}} = \mathbf{y};\ \frac{\partial \mathbf{x}^T\mathbf{y}}{\partial \mathbf{x}} = \mathbf{y};\ \frac{\partial \mathbf{x}^T\mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = 2\mathbf{A}\mathbf{x}; \tag{2.54}$$

where $\mathbf{x}$ and $\mathbf{y}$ are column vectors with the corresponding dimensions, and $\mathbf{A}$ is a square and symmetric matrix. So for instance for the second addend in (2.51) it is adopted that $\mathbf{y}^T = \mathbf{y}^T(k)\mathbf{Z}(k)$ and $\mathbf{x} = \hat{\mathbf{H}}(k)$, while for the last addend in (2.51) $\mathbf{x} = \hat{\mathbf{H}}(k)$ and $\mathbf{A} = \mathbf{Z}^T(k)\mathbf{Z}(k)$.

The presented procedure for optimal parameter estimation (2.53), obtained by the minimization of the sum of squared errors, represents the least squares (LS) algorithm. It is non-recursive and it requires a rather complex computational procedure in each iteration, since the filter coefficients (2.53) are each time calculated from the beginning (such algorithms are denoted as package or, in Anglo-Saxon literature, the off-line algorithms). A much more convenient form of this algorithm is the recursive form, which is able to update the values of coefficients utilizing their previous estimation and newly obtained measurements of the input signal and the reference signal.

2.4.3 Recursive Least Squares (RLS) Algorithm

Let the gain matrix in (2.53) be denoted as

$$\mathbf{P}(k) = \left[\mathbf{Z}^T(k)\mathbf{Z}(k)\right]^{-1}. \tag{2.55}$$

In that case, according to (2.48), the gain matrix in the next step (sampling moment) is

$$\begin{aligned}\mathbf{P}(k+1) &= \left[\mathbf{Z}^T(k+1)\mathbf{Z}(k+1)\right]^{-1} \\ &= \left[\begin{bmatrix}\mathbf{Z}^T(k) & \mathbf{X}(k+1)\end{bmatrix}\begin{bmatrix}\mathbf{Z}(k) \\ \hline \mathbf{X}^T(k+1)\end{bmatrix}\right]^{-1} \\ &= \left[\mathbf{Z}^T(k)\mathbf{Z}(k) + \mathbf{X}(k+1)\mathbf{X}^T(k+1)\right]^{-1}, \\ &= \left[\mathbf{P}^{-1}(k) + \mathbf{X}(k+1)\mathbf{X}^T(k+1)\right]^{-1}\end{aligned} \tag{2.56}$$

where according to (2.11)

$$\mathbf{X}^T(k+1) = [\,x(k+1) \quad x(k) \quad x(k-1) \quad \ldots \quad x(k-M+1)\,], \tag{2.57}$$

and the dashed line denotes the columns and the lines to be added to the existing matrix $\mathbf{Z}(k)$ to form the matrix $\mathbf{Z}(k+1)$.

Using the identity (lemma on matrix inversion) [24]

$$[\mathbf{A} + \mathbf{BCD}]^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}\left(\mathbf{C} + \mathbf{DA}^{-1}\mathbf{B}\right)^{-1}\mathbf{DA}^{-1} \tag{2.58}$$

valid for all matrices with corresponding dimensions and the nonsingular matrix $\mathbf{A}$, one may write the value of the gain matrix in the $(k+1)$ discrete moment

$$\mathbf{P}(k+1) = \mathbf{P}(k) - \mathbf{P}(k)\mathbf{X}(k+1)\left[1 + \mathbf{X}^T(k+1)\mathbf{P}(k)\mathbf{X}(k+1)\right]^{-1}\mathbf{X}^T(k+1)\mathbf{P}(k). \tag{2.59}$$

The expression (2.59) is obtained directly from (2.57) and (2.58) if one adopts in (2.58) that $\mathbf{A} = \mathbf{P}^{-1}(k)$; $\mathbf{B} = \mathbf{X}(k-1)$, $\mathbf{C} = 1$; $\mathbf{D} = \mathbf{X}^T(k+1)$. According to (2.53) and (2.59) it follows

$$\begin{aligned}\hat{\mathbf{H}}(k+1) &= \mathbf{P}(k+1)\left[\mathbf{Z}^T(k) \,|\, \mathbf{X}(k+1)\right]\begin{bmatrix}\mathbf{y}(k)\\ y(k+1)\end{bmatrix}\\ &= \mathbf{P}(k+1)\left[\mathbf{Z}^T(k)\mathbf{y}(k) + \mathbf{X}(k+1)y(k+1)\right]\end{aligned} \tag{2.60}$$

Bearing in mind that according to (2.53) and (2.55)

$$\hat{\mathbf{H}}(k) = \mathbf{P}(k)\mathbf{Z}^T(k)\mathbf{y}(k), \tag{2.61}$$

it is further concluded that

$$\mathbf{Z}^T(k)\mathbf{y}(k) = \mathbf{P}^{-1}(k)\mathbf{H}(k). \tag{2.62}$$

Using (2.56), the expression (2.62) can be written in the form

$$\mathbf{Z}^T(k)\mathbf{y}(k) = \left[\mathbf{P}^{-1}(k+1) - \mathbf{X}(k+1)\mathbf{X}^T(k+1)\right]\mathbf{H}(k). \tag{2.63}$$

By replacing (2.63) into the relation (2.60) one finally obtains the recursive least squares algorithm for the estimation of the unknown parameter vector, $\mathbf{H}$, of an FIR digital filter

$$\hat{\mathbf{H}}(k+1) = \hat{\mathbf{H}}(k) + \mathbf{K}(k+1)\left[y(k+1) - \mathbf{X}^T(k+1)\hat{\mathbf{H}}(k)\right], \tag{2.64}$$

where

$$\mathbf{K}(k+1) = \mathbf{P}(k+1)\mathbf{X}(k+1), \tag{2.65}$$

or, after the expression (2.59) is replaced into the relation (2.65)

$$\mathbf{K}(k+1) = \mathbf{P}(k)\mathbf{X}(k+1)\left[1 + \mathbf{X}^T(k+1)\mathbf{P}(k)\mathbf{X}(k+1)\right]^{-1}. \tag{2.66}$$

The starting value of the gain matrix $\mathbf{P}$ can be obtained by setting $\mathbf{P}(0) = \sigma^2\mathbf{I}$, where σ^2 is a large positive number and $\mathbf{I}$ is a unit matrix with adequate dimensions. The initial value for the parameter vector $\hat{\mathbf{H}}(0)$ can be set to zero value. Alternatively, the initial estimation $\hat{\mathbf{H}}(0)$ of the unknown parameter vector $\mathbf{H}$ of the digital filter can be determined by the non-recursive least squares algorithm (2.53), utilizing the initial package of input signal measurements, *x,* and the desired output, *y,* of the filter with a length of several tens of samples.

The variable

$$e(k+1) = y(k+1) - \mathbf{X}^T(k+1)\hat{\mathbf{H}}(k) \tag{2.67}$$

is also denoted as the measurement residual or innovation, since according to the relation (2.45), (2.46) and (2.57), the expression

$$\hat{y}(k+1) = \mathbf{X}^T(k+1)\hat{\mathbf{H}}(k) \tag{2.68}$$

represents the prediction of the reference signal (desired response or output), $y(k+1)$, based on the input measurement vector $\mathbf{X}(k+1)$ and the previous estimation of the filter parameter vector $\hat{\mathbf{H}}(k)$. In this manner, the whole measurement of the desired output signal (response), $y(k+1)$, does not introduce a new information about the estimated parameters, since even before the measurement data $y(k+1)$ was obtained it was possible to anticipate its value according to (2.68), utilizing the model of digital filter (2.45) and the previous estimation of the filter parameter vector $\hat{\mathbf{H}}(k)$ (relation (2.68) defines the estimated output of the FIR filter before its value has been measured). If the estimation $\hat{\mathbf{H}}(k)$ is equal to the accurate value of the parameter vector in (2.45), then$y(k) = \hat{y}(k)$ and $e(k) = 0$ (see Fig. 2.8)

One of the basic deficiencies of this algorithm is actually its complexity. It can be shown that in each iteration one needs $2.5M^2 + 4M$ multiplication and addition operations (M is the filter order) [6]. With an increasing filter order M, the computational complexity increases with the squared filter order M, which often may be the limiting factor in certain applications. The computational complexity of the RLS algorithm is much larger compared to the $2M + 1$ operations per iteration required in the LMS algorithm. On the other hand, the initial convergence properties of the RLS algorithm are significantly better compared to the LMS algorithm. An advantage of the RLS algorithm is also its insensitiveness to the correlation properties of the input signal, contrary to the LMS algorithm. Namely, the LS and RLS algorithms do not require an a priori information about the statistical properties of the relevant signals, contrary to the LMS algorithms where the value of the factor β is limited by the maximal eigenvalue of the autocorrelation matrix of input signals.

The complete RLS algorithm is systematized in Table 2.1.

2.4.4 Weighted Recursive Least Squares (WRLS) Algorithm with Exponential Forgetting Factor

Recursive least squares (RLS) algorithm in its original version is suitable for the estimation of parameters for stationary conditions, i.e. constant estimated parameters. Basically it is an algorithm with an unlimited memory, where all previous results are equivalently taken into consideration and based on it the estimation of the parameters in the next moment is performed. In the case of a time-variable system this means that the criterion (2.57) will furnish an estimation of the average behavior of the process in the time interval under consideration, and thus such an estimation will not be able to follow correctly the momentary changes of parameters in the digital filter model. To overcome this problem it is necessary

Table 2.1 Flow diagram of RLS algorithm

1. Initialization:

$\hat{\mathbf{H}}(0) = 0, \ \mathbf{P}(0) = \sigma^2\mathbf{I}, \ \sigma^2 \gg 1$

Read in the first sample of the input signal vector $\mathbf{X}(1) = [x(1)\ 0\ \ldots\ 0]$

2. In each discrete moment of time $k = 1, 2, \ldots$, assuming that $\hat{\mathbf{H}}(k-1)$, $\mathbf{X}(k)$ and $\mathbf{P}(k-1)$ are known, calculate:

- Input signal estimation: $\hat{y}(k) = \mathbf{X}^T(k)\hat{\mathbf{H}}(k-1)$
- Error signal: $e(k) = y(k) - \hat{y}(k) = y(k) - \mathbf{X}^T(k)\hat{\mathbf{H}}(k-1)$
- Gain matrix: $\mathbf{P}(k) = \mathbf{P}(k-1) - \dfrac{\mathbf{P}(k-1)\mathbf{X}(k)\mathbf{X}^T(k)\mathbf{P}(k-1)}{1 + \mathbf{X}^T(k)\mathbf{P}(k-1)\mathbf{X}(k)}$

$\mathbf{K}(k) = \mathbf{P}(k)\mathbf{X}(k)$

- Filter filter:coefficients: $\hat{\mathbf{H}}(k) = \hat{\mathbf{H}}(k-1) + \mathbf{K}(k)e(k)$
- Update input vector:

$\mathbf{X}^T(k+1) = [\,x(k+1) \quad x(k) \quad x(k-1) \quad \ldots \quad x(k-M+1)\,]$, assuming that $x(i) = 0$ for $i \leq 0$ (causal excitation signal)

3. Increment counter k by 1 and repeat the procedure from the step 2

to utilize an algorithm with limited memory, which essentially reduces to the introduction of the forgetting factor. In other words, the criterion (2.47) should be modified in such a manner that the "older" measurements take part in it with a decreased weight, in order to enable the RLS algorithm to follow the parameter changes. This means that in the case of filter parameters variable in time one should replace the criterion function (2.47) with an exponentially weighted sum of the squares of error signals

$$J(k) = \frac{1}{2}\sum_{i=0}^{k} \rho^{k-i} e^2(i), \tag{2.69}$$

where ρ represents the forgetting factor (FF) determining the effective memory of the algorithm and its value is within the range

$$0 < \rho \leq 1. \tag{2.70}$$

For stationary conditions (not changing in time) one applies $\rho = 1$, and in this case the criterion function defined by (2.69) becomes equal with (2.47), and the algorithm that recursively minimizes the given criterion has an unlimited memory. In this way, the estimated parameters have a high accuracy, since, asymptotically taken, one eliminates the influence of noisy states by averaging them. For the conditions when the estimated parameters change, the forgetting factor $\rho = 1$ is not convenient, because the adaptation of the estimated parameters towards the real values is relatively slow. Because of that one should use $\rho < 1$. By utilizing $\rho < 1$ one obtains different weightings for previous measurements, i.e. the previous measurements are taken with a smaller weight compared to the more recent ones.

Assuming that a nonstationary signal consists of stationary segments of a given length, the forgetting factor ρ can be determined in the following manner. Starting from the assumption that the value of ρ is close to zero, one may write

$$\rho^k = e^{k\ln\rho} = e^{k\ln(1+\rho-1)} \approx e^{-k(1-\rho)},$$

i.e.

$$\rho^k = e^{-k/\tau};\ \tau = \frac{1}{1-\rho}. \tag{2.71}$$

In this manner, by choosing a forgetting factor of $\rho < 1$, the effective memory of the algorithm becomes

$$\tau = \frac{-1}{\log\rho}, \tag{2.72}$$

which, in the case when the values of ρ are close to one, is approximately equal to

$$\tau = \frac{1}{1-\rho}. \tag{2.73}$$

Expression (2.71) shows that the measurements older than τ $(k > \tau)$ are allocated a weight smaller than $e^{-1} \approx 0.36$ $(k = \tau)$ compared to the unit weight allocated to the current measurement $(k = 0)$. In other words, thus chosen weight factor ρ in the criterion (2.69) corresponds to the exponentially decaying memory of the algorithm, where the time constant τ of the exponential curve corresponds to the memory length in the adopted units of time or to the number of periods of signal sampling.

Through minimization of the criterion function (2.69) one arrives to the Weighted Recursive Least Squares—WRLS algorithm. The derivation of the WRLS algorithm is identical to that of the RLS algorithm. Namely, the criterion (2.69) can be written in square matrix form

$$J(k) = \frac{1}{2}\mathbf{e}^T(k)\mathbf{W}(k)\mathbf{e}(k), \tag{2.74}$$

where

$$\mathbf{e}(k) = \begin{bmatrix} e(0) \\ e(1) \\ \vdots \\ e(k) \end{bmatrix};\ \mathbf{W}(k) = \begin{bmatrix} w(0) & 0 & \cdots & 0 \\ 0 & w(1) & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & w(k) \end{bmatrix};\ w(i) = \rho^{k-i},\ i = 0, 1, \ldots, k. \tag{2.75}$$

Since according to (2.49)

$$\mathbf{e}(k) = \mathbf{y}(k) - \mathbf{Z}(k)\hat{\mathbf{H}}(k),$$

by replacing this expression to (2.74) one obtains

$$J(k) = \frac{1}{2}[\mathbf{y}^T(k)\mathbf{W}(k)\mathbf{y}(k) - \hat{\mathbf{H}}^T(k)\mathbf{Z}^T(k)\mathbf{W}(k)\mathbf{y}(k) - \mathbf{y}^T(k)\mathbf{W}(k)\mathbf{Z}(k)\hat{\mathbf{H}}(k) + \hat{\mathbf{H}}^T(k)\mathbf{Z}^T(k)\mathbf{W}(k)\mathbf{Z}(k)\hat{\mathbf{H}}(k)] \tag{2.76}$$

Similarly to the derivation of the standard RLS algorithm, if one further applies the rules (2.54) for differentiation of the corresponding terms in (2.76) over the vector $\hat{\mathbf{H}}$, one obtains the necessary condition for the minimum of the criterion (2.74)

$$\frac{\partial J(k)}{\partial \hat{\mathbf{H}}(k)} = -\mathbf{Z}^T(k)\mathbf{W}(k)\mathbf{y}(k) + \mathbf{Z}^T(k)\mathbf{W}(k)\mathbf{Z}(k)\hat{\mathbf{H}}(k) = 0, \tag{2.77}$$

and the nonrecursive algorithm of weighted least squares (non-recursive WRLS algorithm) directly follows from it

$$\hat{\mathbf{H}}(k) = \left[\mathbf{Z}^T(k)\mathbf{W}(k)\mathbf{Z}(k)\right]^{-1}\mathbf{Z}^T(k)\mathbf{W}(k)\mathbf{y}(k). \tag{2.78}$$

The recursive version of the WRLS algorithm is obtained from (2.78) in a manner identical to the one used to derive the recursive algorithm (2.38), (2.64)–(2.66) from its non-recursive form (2.53). According to the expression (2.78) one may write the block-matrix relation

$$\begin{aligned}\hat{\mathbf{H}}(k+1) &= \left[\mathbf{Z}^T(k) \quad \mathbf{X}(k+1)\right]\begin{bmatrix}\rho\mathbf{W}(k) & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\mathbf{y}(k)\\ \hline y(k+1)\end{bmatrix}\\ &= \mathbf{P}(k+1)\left[\rho\mathbf{Z}^T(k)\mathbf{W}(k)y(k) + \mathbf{X}(k+1)y(k+1)\right],\end{aligned} \tag{2.79}$$

where

$$\begin{aligned}&\mathbf{P}(k+1) = \left[\mathbf{Z}^T(k+1)\mathbf{W}(k+1)\mathbf{Z}(k+1)\right]^{-1};\ \mathbf{Z}^T(k+1) = \left[\mathbf{Z}^T(k) \quad \mathbf{X}(k+1)\right]\\ &\mathbf{W}(k+1) = \begin{bmatrix}\rho\mathbf{W}(k) & 0\\ 0 & 1\end{bmatrix};\ \mathbf{y}(k+1) = \begin{bmatrix}\mathbf{y}(k)\\ \hline y(k+1)\end{bmatrix}\end{aligned} \tag{2.80}$$

According to (2.80) it follows further that

$$\begin{aligned}\mathbf{P}^{-1}(k+1) &= \rho\mathbf{Z}^T(k)\mathbf{W}(k)\mathbf{Z}(k) + \mathbf{X}(k+1)\mathbf{X}^T(k+1)\\ &= \rho\mathbf{P}^{-1}(k) + \mathbf{X}(k+1)\mathbf{X}^T(k+1)\end{aligned} \tag{2.81}$$

By applying the lemma on matrix inversion (2.58) to the relation (2.81), adopting that $\mathbf{A} = \rho\mathbf{P}^{-1}(k)$, $\mathbf{B} = \mathbf{X}(k+1)$, $\mathbf{C} = 1$ and $\mathbf{D} = \mathbf{X}^T(k+1)$ it can be written

$$\begin{aligned}\mathbf{P}(k+1) = &\frac{1}{\rho}\mathbf{P}(k)\\ &-\frac{1}{\rho}\mathbf{X}(k+1)\left[1 + \mathbf{X}^T(k+1)\frac{1}{\rho}\mathbf{P}(k)\mathbf{X}(k+1)\right]^{-1}\mathbf{X}^T(k+1)\frac{1}{\rho}\mathbf{P}(k)\end{aligned} \tag{2.82}$$

Bearing in mind that according to (2.78) and (2.80)

$$\hat{\mathbf{H}}(k) = \mathbf{P}(k)\mathbf{Z}^T(k)\mathbf{W}(k)\mathbf{y}(k), \tag{2.83}$$

we conclude that

$$\mathbf{Z}^T(k)\mathbf{W}(k)\mathbf{y}(k) = \mathbf{P}^{-1}(k)\hat{\mathbf{H}}(k), \tag{2.84}$$

from where, after introducing the expression (2.79), one obtains

$$\mathbf{Z}^T(k)\mathbf{W}(k)\mathbf{y}(k) = \frac{1}{\rho}\left[\mathbf{P}^{-1}(k+1) - \mathbf{X}(k+1)\mathbf{X}^T(k+1)\right]\hat{\mathbf{H}}(k). \tag{2.85}$$

By replacing the expression (2.85) in relation (2.81), one obtains

$$\hat{\mathbf{H}}(k+1) = \hat{\mathbf{H}}(k) + \mathbf{P}(k+1)\mathbf{X}(k+1)\left[y(k+1) - \mathbf{X}^T(k+1)\hat{\mathbf{H}}(k)\right]. \tag{2.86}$$

The expression

$$\mathbf{K}(k+1) = \mathbf{P}(k+1)\mathbf{X}(k+1) \tag{2.87}$$

defines the gain matrix of the recursive algorithm for the estimation of digital filter parameters (2.86). By replacing the expression (2.82) in the relation (2.87) the latter can be written in the alternative form

$$\mathbf{K}(k+1) = \mathbf{P}(k)\mathbf{X}(k+1)\left[\rho + \mathbf{X}^T(k+1)\mathbf{P}(k)\mathbf{X}(k+1)\right]^{-1} \tag{2.88}$$

Relations (2.82), (2.86) and (2.87) or (2.88) define the recursive WRLS algorithm, i.e.

$$\hat{\mathbf{H}}(k) = \hat{\mathbf{H}}(k-1) + \mathbf{K}(k)\left[y(k) - \mathbf{X}^T(k)\hat{\mathbf{H}}(k-1)\right], \tag{2.89}$$

where

$$\mathbf{P}(k) = \frac{1}{\rho}\left\{\mathbf{P}(k-1) - \mathbf{P}(k-1)\mathbf{X}(k)\left[\rho + \mathbf{X}^T(k)\mathbf{P}(k-1)\mathbf{X}(k)\right]^{-1}\mathbf{X}^T(k)\mathbf{P}(k-1)\right\}. \tag{2.90}$$

To start the recursive procedure (2.88)–(2.90) it is necessary to adopt the initial values $\mathbf{P}(0)$ and $\hat{\mathbf{H}}(0)$ and they are chosen in the same manner as in the RLS algorithm, i.e. $\hat{\mathbf{H}}(0) = \mathbf{0}$ and $\mathbf{P}(0) = \sigma^2\mathbf{I}$, where $\sigma^2 \gg 1$, and $\mathbf{I}$ is the unit matrix with corresponding dimensions.

The very name "forgetting factor", ρ, suggests that it represents the measure of taking into account the previous measurements in the estimation process. In other words, the choice of the values for the forgetting factor determines how quickly one neglects the influence of the previous measurements. In the estimation of stationary parameters it is desirable that the algorithm similarly takes into account all previous measurements, because the system does not change with time and in this case one assumes $\rho = 1$, i.e. $\tau = \infty$. However, the situation is completely

different if parameters vary with time. In this case the so-called "older" measurements do not have a large significance, because they do not bear information about the newly occurring changes. Because of that their importance is decreased by the choice of the forgetting factor with a value smaller than 1. For instance, $\rho = 0.9$ corresponds, according to (2.73), to the value of memory of the algorithm of $\tau = 10$ signal samples.

Through the choice of the forgetting factor $\rho < 1$ one achieves faster adaptation of the parameters to accurate values in such a manner that better results are obtained with smaller values of ρ (which corresponds to smaller τ), but one simultaneously increases the variance of the parameter estimation because of the influence of noisy measurements. On the other hand, through the use of a fixed forgetting factor $\rho < 1$, the gain matrix $\mathbf{P}(k)$ is constantly divided by a factor smaller than 1, which may lead to the consequence that the gain matrix (2.89) achieves a very high value, and thus the algorithm becomes very sensitive to random disturbances or numerical errors propagating through the residual of measurements

$$e(k) = y(k) - \mathbf{X}^T(k)\hat{\mathbf{H}}(k-1). \tag{2.91}$$

The basic deficiency of the application of RLS algorithms with a fixed FF in time-variant systems follows from here. The choice of the time constant τ, i.e. the forgetting factor ρ, depends on the expected dynamics of the filter parameter change and they should be chosen to keep parameters approximately constant on the interval with a length of τ signal samples. Starting from expression (2.71), for nonstationary signals containing intervals of quasi-stationarity, on nonstationary segments it is useful to utilize $\rho < 1$, which corresponds to small τ. For stationary segments one should adopt $\rho \approx 1$, which corresponds to a large value of τ $(\tau \to \infty)$. In order to achieve an adequate ability for adaptation in the change of time-variant systems, which includes nonstationary changes, and simultaneously to avoid a significant influence to the variance of the estimated parameters in the intervals without changes, as well as the influence to their accuracy, it is necessary to vary adaptively the forgetting factor during the operation of the algorithm itself.

There is a wider discussion in Chap. 3 about the strategy of choice of variable forgetting factor while applying the adaptive algorithm itself. In practical situations one sometimes adopts that the forgetting factor ρ varies over time within the quasi-stationarity interval and that it exponentially increases to 1 [24]. This corresponds to choosing

$$\rho(k) = \rho_0\rho(k-1) + (1-\rho_0); \quad k = 1, 2, \ldots \tag{2.92}$$

where the usual choice is $\rho_0 = 0.99$ and $\rho(0) = 0.95$. According to (2.92) one may write

$$\rho(1) = \rho_0\rho(0) + (1-\rho_0)$$
$$\rho(2) = \rho_0\rho(1) + (1-\rho_0) = \rho_0^2\rho(0) + \rho_0(1-\rho_0) + (1-\rho_0),$$

from which one inductively concludes that

$$\rho(k) = \rho_0^k \rho(0) + (1 - \rho_0) \sum_{i=0}^{k-1} \rho_0^{k-1-i},$$

i.e., if one substitutes $k - 1 - i = j$

$$\begin{aligned} \rho(k) =& \rho_0^k \rho(0) + (1 - \rho_0) \sum_{j=0}^{k-1} \rho_0^j \\ =& \rho_0^k \rho(0) + (1 - \rho_0) \frac{1 - \rho_0^k}{1 - \rho_0}. \\ =& 1 - \rho_0^k [1 - \rho_0] \end{aligned} \tag{2.93}$$

According to the above expression one obtains

$$\lim_{k \to \infty} \rho(k) = 1. \tag{2.94}$$

Table 2.2 systematizes the WRLS algorithm

2.5 Adaptive Algorithms for the Estimation of the Parameters of IIR Filters

FIR adaptive filters have a unimodal criterion function with a single global minimum and are not susceptible to instability with a change of the value of their parameters, because the adequate filter transfer function has all poles in the origin, $z = 0$, of the z-complex plane, i.e. all poles of the transfer function are within the stability region (unit circle $|z| = 1$) [3]. The process of convergence of the parameters of an FIR filter towards the optimal values, corresponding to the minimum of the adopted criterion, is well researched and the results about it are available in literature [4]. These properties make them more desirable than other structures and because of that they have a very wide practical application [17]. However, with an increase of the length of the impulse response of the modeled system one must proportionally increase the number of the filter parameters. This leads to an increased complexity of the adaptive algorithm, a decrease of the convergence speed, and in the case of exceptionally long impulse response also to unacceptably high complexity of the suitable digital hardware.

It is possible to overcome this deficiency by using adaptive filters with infinite impulse response, the IIR filters.

The main advantage of the adaptive IIR filters in comparison to the adaptive FIR filters is that by using the same or even a smaller number of parameters one is able to significantly better describe a given system for signal transfer and processing. The response of this system can be much better described by the output

Table 2.2 Flow diagram of WRLS algorithm

1. Initialization
   - $\hat{\mathbf{H}}(0) = 0, \ \mathbf{P}(0) = \sigma^2\mathbf{I}, \ \sigma^2 \gg 1$
- Read in the first sample of the input signal vector $\mathbf{X}(1) = [x(1)\ 0\ \ldots\ 0]$
2. In each discrete moment of time $k = 1, 2, \ldots$, assuming that $\hat{\mathbf{H}}(k-1)$, $\mathbf{X}(k)$ and $\mathbf{P}(k-1)$ are known, calculate:
- Output signal estimation: $\hat{y}(k) = \mathbf{X}^T(k)\hat{\mathbf{H}}(k-1)$
- Error signal: $e(k) = y(k) - \mathbf{X}^T(k)\hat{\mathbf{H}}(k-1)$
- Gain matrix:
- $\mathbf{P}(k) = \frac{1}{\rho}\left\{\mathbf{P}(k-1) - \dfrac{\mathbf{P}(k-1)\mathbf{X}(k)\mathbf{X}^T(k)\mathbf{P}(k-1)}{\rho + \mathbf{X}^T(k)\mathbf{P}(k-1)\mathbf{X}(k)}\right\}$

  $\mathbf{K}(k) = \mathbf{P}(k-1)\mathbf{X}(k)\left[\rho + \mathbf{X}^T(k)\mathbf{P}(k-1)\mathbf{X}(k)\right]^{-1}$
- Filter coefficients: $\hat{\mathbf{H}}(k) = \hat{\mathbf{H}}(k-1) + \mathbf{K}(k)e(k)$
- Update input vector

  $\mathbf{X}^T(k+1) = [\,x(k+1) \quad x(k) \quad x(k-1) \quad \ldots \quad x(k-M+1)\,]$, assuming that $x(i) = 0$ for $i \leq 0$ (causal excitation signal)
3. Increment the counter k by 1 and repeat the procedure from the step 2

signal from a filter whose transfer function has both zeroes and poles (IIR) in comparison to a filter whose transfer function "has zeroes only" (FIR). So for example an adaptive IIR filter with a sufficiently high order can accurately model an unknown system described by a certain number of zeroes and poles, while an adaptive FIR filter can only approximate it. In other words, to describe some system with a given accuracy, an IIR filter generally requires a much smaller number of coefficients that the corresponding FIR filter.

Figure 2.1 shows a general structure of an adaptive filter, which may be an adaptive IIR filter. $x(k)$ denotes the input signal, $y(k)$ is output signal, $e(k)$ is error signal, $\mathbf{H}$ is an unknown parameter vector of the estimated filter, and $y(k)$ is the reference signal. The adaptive IIR filter consists from two basic parts: a digital IIR filter determined by the values of the variable parameters of the vector $\mathbf{H}$ and the corresponding adaptive algorithm according to which the unknown parameters are updated to minimize a given criterion function which is a function of the error signal $e(k)$.

Basically there are two approaches to adaptive digital IIR filtering, which correspond to different formulations of the error signal $e(k)$. They are denoted as the equation error (EE) method and the output error (OE) method. The EE method is characterized by the updating of the feedback coefficients of the IIR adaptive filter in the domain of zeroes, which basically leads to the adaptive FIR filters and the corresponding adaptive algorithms from their domain.

The adaptive IIR filters based on the EE method are shown schematically in Fig. 2.9. The signal $y_e(k)$ is defined by the following expression

$$y_e(k) = \sum_{i=1}^{N} a_i(k)y(k-i) + \sum_{i=0}^{M} b_i(k)x(k-i), \tag{2.95}$$

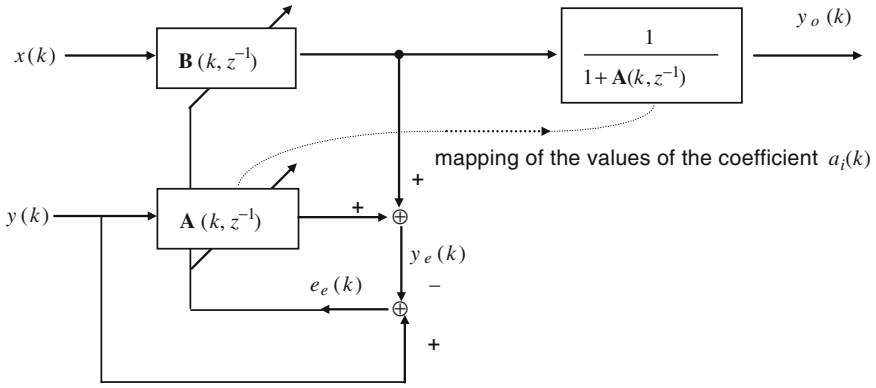

Fig. 2.9 Block diagram of an EE IIR digital filter

where $\{a_i(k)\}$ and $\{b_i(k)\}$ are variable parameters of the IIR filter, estimated by a suitable adaptive algorithm (the adaptive algorithm recursively minimizes the adopted criterion function).

The signal $y_e(k)$ is a function of the momentary value and the M previous values of the input signal $x(k)$, as well as of N previous values of the reference signal $y(k)$. One should note that it does not depend on the values of the filter output signal, $y_o(k)$. Neither the reference signal $y(k)$, nor the input signal $x(k)$ depend on the filter coefficients, so the determination of the values of $y_e(k)$ with regard to the filter parameters is a non-recursive procedure. Expression (2.95) can be written in its polynomial form

$$y_e(k) = \mathbf{B}\left(k, z^{-1}\right)x(k) + \mathbf{A}\left(k, z^{-1}\right)y(k), \tag{2.96}$$

where polynomials are in a discrete moment kT (T – signal sampling period)

$$\begin{aligned} \mathbf{A}\left(k, z^{-1}\right) &= \sum_{i=1}^{N} a_i(k) z^{-i} \\ \mathbf{B}\left(k, z^{-1}\right) &= \sum_{i=0}^{M} b_i(k) z^{-i}. \end{aligned} \tag{2.97}$$

Here z^{-1} represents the operator of unit delay, i.e. $x(k)z^{-1} = x(k-1)$.

In the EE model the error signal $e_e(k)$ is defined in the following manner

$$e_e(k) = y(k) - y_e(k) = y(k) - \mathbf{B}\left(k, z^{-1}\right)x(k) - \mathbf{A}\left(k, z^{-1}\right)y(k). \tag{2.98}$$

It follows from expressions (2.96)–(2.98) that $e_e(k)$ is a linear function of the coefficients of the polynomials **A** and **B** (elements of the vector **H**), i.e. that the criterion function is defined according to the EE mean square error, MSE = $E\left\{e_e^2(k)\right\}$, a quadratic equation with a unique global minimum. Therefore the properties of the EE adaptive IIR filters are similar to those of the adaptive FIR

filters (where $\mathbf{A}(k, z^{-1}) = 0$). They utilize similar adaptive algorithms, with similar convergence properties as the FIR adaptive filters [1]. Equation (2.95) can be also represented in the vectorial form

$$y_e(k) = \mathbf{H}^T(k)\mathbf{X}_e(k), \tag{2.99}$$

which represents a scalar product of the following two vectors

$$\mathbf{H}^T(k) = [b_0(k)\ b_1(k) \ldots b_M(k)\ a_1(k)\ a_2(k)\ \ldots a_N(k)], \tag{2.100}$$

$$\mathbf{X}_e(k) = [x(k)\ x(k-1)\ \ldots\ x(k-M)\ y(k-1)\ y(k-2)\ \ldots\ y(k-N)]^T. \tag{2.101}$$

The vector $\mathbf{H}(k)$ contains estimated parameters in a discrete moment k, and the vector $\mathbf{X}_e(k)$ contains actual and delayed values of the input and reference signal. It is important to note that $\mathbf{X}_e(k)$ is not a function of the vector $\mathbf{H}(k)$. Various algorithms may be used for the estimation of parameters, like Recursive Least Square (RLS) method, Weighted Recursive Least Square (WRLS), Least Mean Square (LMS) method and others [4, 6]. These algorithms were described in detail in the previous sections.

IIR adaptive algorithms based on the EE model may converge to values shifted in comparison to the optimal ones, which leads to an erroneous estimation of parameters. Although EE adaptive IIR filters have good convergence properties, in principle they may be completely unacceptable models if this shift is significant in the estimation of parameters. Let us note that the estimation of the parameter vector is unbiased if the mathematical expectation of the parameter vector estimation (the estimated parameters represent a random vector) is equal to their accurate (optimal) value.

OE adaptive IIR filters update the coefficients of the IIR filters in the return branch directly both in the domain of zeroes and in the domain of poles. In this case the estimated parameters are not shifted compared to the optimal values, but the adaptive algorithm may converge to a local minimum of the criterion function. This means that the estimated values do not have to correspond to the optimal values. The block diagram of the OE adaptive IIR filter is shown in Fig. 2.10. In this case the equation of the output error, OE, is defined as

$$e_o(k) = y(k) - y_o(k). \tag{2.102}$$

The error signal $e_o(k)$ is a nonlinear function of the filter coefficients, and the criterion function $MSE = E\{e_0^2(k)\}$ may have more than one local minimum. The corresponding adaptive algorithms in principle converge slower than the EE algorithm and may converge to local minima. However, the distinction of the OE method compared to the EE method is that the adaptive filter generates the output $y_o(k)$ based on the input signal $x(k)$ only, while in the case of the EE method the reference signal, $y(k)$, also takes part in the adaptation process.

The signal at the output from the OE IIR adaptive filter is defined by the difference equation

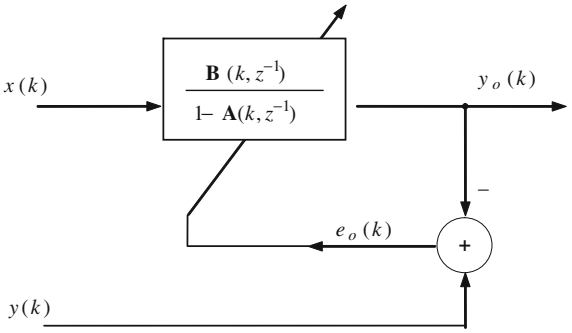

Fig. 2.10 Block diagram of the OE adaptive digital IIR filter

$$y_o(k) = \sum_{i=1}^{N} a_i(k) y_o(k) + \sum_{i=0}^{M} b_i(k) x(k-i). \tag{2.103}$$

It is seen from (2.103) that the output signal $y_o(k)$ is a function of its N previous values, as well as of the actual and the M previous values of the input signal $x(k)$. Such a feedback over the output signal significantly influences the adaptive algorithm, making it much more complex compared to the EE approach. Analogously to expressions (2.96) and (2.97), the expression (2.103) can be represented in the polynomial form

$$y_o(k) = \left(\frac{\mathbf{B}(k, z^{-1})}{1 - \mathbf{A}(k, z^{-1})} \right) x(k), \tag{2.104}$$

or like a vectorial equation (scalar product)

$$y_o(k) = \mathbf{H}^T(k) \mathbf{X}_o(k), \tag{2.105}$$

where the vector of variable parameters $\mathbf{H}(k)$ is defined by (2.100), and $\mathbf{X}_o(k)$

$$\mathbf{X}_o^T(k) = [x(k)\ x(k-1)\ \ldots\ x(k-M)\ y_o(k-1)\ y_o(k-2)\ \ldots\ y_o(k-N)] \tag{2.106}$$

The output $y_o(k)$ is a nonlinear function of the parameters of the vector $\mathbf{H}$, because $\{y_o(k-i),\ i = 1, 2, \ldots, N\}$ like an element of the parameter $\mathbf{X}_o$, is a function of the coefficients of the filter in the previous k iterations. This fact significantly complicates synthesis of adaptive algorithms for the estimation of parameters of a digital filter.

Adaptive algorithms which will be described in this section can be represented in a general form denoted as the Gauss-Newton algorithm [25].

The Gauss-Newton algorithm represents a stochastic version of the Newton deterministic algorithm (2.30). To illustrate this algorithm, let us consider the model of a stochastic signal described by the difference equation (the linear regression equation)

$$y(k) = \mathbf{H}^T \mathbf{X}(k) + e(k), \tag{2.107}$$

where $y(k)$ and $\mathbf{X}(k)$ are measurable variables, and $\mathbf{H}$ is the unknown parameter vector to be determined. In the above expression $e(k)$ represents a random residual or error and the natural way to determine $\mathbf{H}$ is to minimize the variance of error, i.e. the mean square error

$$J(\mathbf{H}) = \frac{1}{2} E\{e^2(k)\} = \frac{1}{2} E\Big\{ \big(y(k) - \mathbf{H}^T \mathbf{X}(k)\big)^2 \Big\}, \tag{2.108}$$

where $E\{\cdot\}$ denotes mathematical expectation. Since $J(\mathbf{H})$ is a quadratic function of the argument $\mathbf{H}$, its minimum is obtained by solving the equation

$$-\frac{\partial}{\partial \mathbf{H}} J(k) = -\nabla J(\mathbf{H}) = E\big\{\mathbf{X}(k)\big[y(k) - \mathbf{H}^T \mathbf{X}(k)\big]\big\} = 0. \tag{2.109}$$

The quoted problem cannot be exactly solved, since the aggregate function of the probability density of random variables $(y(k), \mathbf{X}(k))$, which is necessary to determine the mathematical expectation, is unknown. One of the ways to overcome this difficulty is to replace the unknown mathematical expectation by the corresponding arithmetical means, i.e. to adopt the approximation

$$E\{f(x)\} = \frac{1}{M} \sum_{i=1}^{M} f(i), \tag{2.110}$$

which leads to the least square algorithm, described in detail in the precious section. Another possibility is to apply a stochastic version of the Newton deterministic scheme (2.30)

$$\mathbf{H}(k) = \mathbf{H}(k-1) - \gamma(k)\big[\nabla^2 J(\mathbf{H}(k-1))\big]^{-1} \nabla J(\mathbf{H}(k-1)), \tag{2.111}$$

where the Hessian is

$$\nabla^2 J(\mathbf{H}(k-1)) = \frac{d^2}{d\mathbf{H}^2} J(k) = \frac{d}{d\mathbf{H}} \nabla J(\mathbf{H}(k-1)) = E\big\{\mathbf{X}(k)\mathbf{X}^T(k)\big\}. \tag{2.112}$$

It can be seen that the Hessian is independent on $\mathbf{H}$. The Hessian can be determined as the solution $\mathbf{R}$ of the equation

$$E\big\{\big[\mathbf{X}(k)\mathbf{X}^T(k) - \mathbf{R}\big]\big\} = 0. \tag{2.113}$$

To iteratively solve this equation one may further use the Robbins-Monro stochastic approximation procedure [24, 28].

A typical problem of stochastic approximation may be formulated in the following manner. Let $\{e(k)\}$ represent a series of stochastic variables, with an identical distribution function, where k denotes the index of a discrete moment. Let one further have a given function $Q(x, e(k))$ of two arguments x and $e(k)$, whose form does not have to be known accurately, but for each adopted x and the

obtained $e(k)$ one can determine the value of the function $Q(\cdot,\cdot)$. The problem is now to determine the solution of the equation

$$E\{Q(x,e(k))\} = f(x) = 0, \tag{2.114}$$

where $E\{\cdot\}$ denotes the mathematical expectation with regard to the random variable $e(k)$, where it is assumed that the user does not know the distribution function, i.e. the probability density, of the stochastic variable $e(k)$. The posed problem reduces to the determination of the series $x(k)$, $k = 1, 2, \ldots$, the calculation of the corresponding values of $Q(x,e(k))$ and the determination of the solution of the Eq. (2.114). This equation is also denoted as the regression equation. Its trivial solution consists in fixing the variable x, determining a large number of values of $Q(x,e(k))$ for the adopted x, with the aim to obtain a good estimation of the $f(x)$, and repeating such procedures for a certain number of new values of the variable x until the solution of the regression equation is found (2.114). Obviously such a procedure is not efficient, since much time is spent to estimate $f(x)$ for the values of the variable x which significantly differ from the looked-for solution (2.114). Robbins and Monro proposed the following iterative solution for the determination of the root of the Eq. (2.114)

$$\hat{x}(k) = \hat{x}(k-1) + \gamma(k)Q(\hat{x}(k-1), e(k)), \tag{2.115}$$

where $\{\gamma(k)\}$ is a series of positive scalar variables which tend to zero with an increase of the index k. The convergence properties of the proposed procedure were analyzed by Robbins and Monro, Blum and Dvoretsky, where it was shown that the series (2.115) under certain conditions will converge to the solution of the Eq. (2.114). Typical assumptions in these analyses were that the terms of the series $\{e(k)\}$ are independent stochastic vectors, which is not fulfilled in a general case [24, 28]. Especially for the problem under consideration (2.113)

$$\hat{x} = \mathbf{R};\ e(k) = \mathbf{X}(k);\ Q(\hat{x}, e(k)) = \mathbf{X}(k)\mathbf{X}^T(k) - \mathbf{R}, \tag{2.116}$$

so that the algorithm (2.115) reduces to the form

$$\begin{aligned}\mathbf{R}(k) &= \mathbf{R}(k-1) + \gamma(k)\left[\mathbf{X}(k)\mathbf{X}^T(k) - \mathbf{R}(k-1)\right] \\ &= [1-\gamma(k)]\mathbf{R}(k-1) + \gamma(k)\mathbf{X}(k)\mathbf{X}^T(k).\end{aligned} \tag{2.117}$$

In this manner, our estimation of the Hessian $\nabla^2 J(\mathbf{H})$ in the moment k is represented by the matrix $\mathbf{R}(k)$. Using this estimation, the unknown parameter vector in the moment k can be estimated using the stochastic Newton algorithm (2.111), i.e.

$$\mathbf{H}(k) = \mathbf{H}(k-1) + \gamma(k)\mathbf{R}^{-1}(k)\mathbf{X}(k)\left[y(k) - \mathbf{X}^T(k)\mathbf{H}(k-1)\right], \tag{2.118}$$

where $\mathbf{X}(k)\left[y(k) - \mathbf{X}^T(k)\mathbf{H}(k-1)\right]$ represents the approximation of the gradient $\nabla J(\mathbf{H})$ in the moment $(k-1)$, which is obtained if the mathematical expectation in (2.109) is approximated by only a single realization of the random process.

A modified version of this algorithm, useful for the estimation of the unknown parameters of the IIR filter, is given by the expression [24]

$$\hat{\mathbf{H}}(k+1) = \hat{\mathbf{H}}(k) + \alpha \mathbf{R}^{-1}(k+1)\mathbf{X}_F(k)e_G(k), \tag{2.119}$$

where $\mathbf{X}_F(k)$ and $e_G(k)$ are filtered versions of the input signal vector $\mathbf{X}(k)$ and the error signal $e(k)$ in accordance with

$$\mathbf{X}_F(k) = F(k,z)\mathbf{X}(k); \;\; e_G(k) = G(k,z)e(k), \tag{2.120}$$

where $\mathbf{X}(k)$ and $e(k)$ may be $\mathbf{X}_e(k)$ and $e_e(k)$ for the EE method, or $\mathbf{X}_o(k)$ and $e_o(k)$ for the OE method, respectively. The filters, i.e. the polynomials, $F(k,z)$ and $G(k,z)$ are defined as

$$F(k,z) = \sum_{i=0}^{n} f_i(k)\, z^{-i}, \tag{2.121}$$

$$G(k,z) = \sum_{i=0}^{m} g_i(k)\, z^{-i}. \tag{2.122}$$

The scalar variable α controls the convergence speed of the algorithm, and the matrix $\mathbf{R}(k)$ is updated as

$$\mathbf{R}(k+1) = \rho \mathbf{R}(k) + \alpha \mathbf{X}_F(k)\mathbf{X}_F^T(k), \tag{2.123}$$

where $\rho = 1 - \alpha$ is the forgetting factor. Typical values for ρ are between 0.9 and 0.99, which corresponds to the effective memory between 10 and 100 samples, respectively [1] (see section 2.4.4).

From (2.119) it follows that to update the filter parameters it is necessary to know the values of the inverse matrix of $\mathbf{R}$, i.e. $\mathbf{R}^{-1}$. This is very complex from the computational point of view, and thus $\mathbf{R}^{-1}$ is most often directly updated, using the lemma on matrix inversion (2.58) and the expression (2.123)

$$\mathbf{R}^{-1}(k+1) = \frac{1}{\rho}\left(\mathbf{R}^{-1}(k) - \frac{\mathbf{R}^{-1}(k)\mathbf{X}_F(k)\mathbf{X}_F^T(k)\mathbf{R}^{-1}(k)}{\rho/\alpha + \mathbf{X}_F^T(k)\mathbf{R}^{-1}(k)\mathbf{X}_F(k)}\right). \tag{2.124}$$

The role of the matrix $\mathbf{R}^{-1}$ is to speed the convergence of the adaptive algorithm, and the price to pay is the increase of computational complexity. If the value $\mathbf{R}^{-1}(k+1)$ in (2.119) is replaced by a unit matrix $\mathbf{I}$, one obtains an algorithm with worse convergence properties, but also a lower complexity, of the order of $(M+N)$ compared to $(M+N)^2$ of arithmetic operations in the basic algorithm.

The parameter parameter: vector is most often initialized to $\mathbf{H}(0) = 0$, where $\mathbf{0}$ is the zero-vector with corresponding dimensions, whose all components are 0, and $\mathbf{R}(0) = \sigma^2\mathbf{I}$, where σ^2 is a small, positive and scalar variable. Other initial values may be defined too, but one has to take care that $\mathbf{R}$ is a positively definite matrix, in order to enable the determination of the inverse matrix $\mathbf{R}^{-1}$, and that the poles

Table 2.3 Flow diagram of EE-WRLS algorithm

1. Initialization
- $\hat{\mathbf{H}}(0) = 0;\ \mathbf{R}^{-1}(0) = \sigma^2\mathbf{I};\ \sigma^2 \ll 1$
- Generate the sample of the input signal $x(0)$ and the reference signal $y(0)$
- Initial error of Eq. $e_e(0) = y(0)$
- Read in the forgetting factor $0.9 \leq \rho \leq 0.99;\ \alpha = 1 - \rho$

2. In each discrete moment of time $k = 1,\ 2,\ \ldots$, assuming that $\hat{\mathbf{H}}(k-1)$, $e_e(k-1)$, $\mathbf{R}^{-1}(k-1)$ and $\mathbf{X}_e(k)$ are known, calculate:
- Gain matrix

$$\mathbf{R}^{-1}(k) = \frac{1}{\rho}\left(\mathbf{R}^{-1}(k-1) - \frac{\mathbf{R}^{-1}(k-1)\mathbf{X}_e(k-1)\mathbf{X}_e^T(k-1)\mathbf{R}^{-1}(k-1)}{\rho/\alpha + \mathbf{X}_e^T(k-1)\mathbf{R}^{-1}(k-1)\mathbf{X}_e(k-1)}\right)$$

- Filter coefficients

$$\hat{\mathbf{H}}(k) = \hat{\mathbf{H}}(k-1) + \alpha\mathbf{R}^{-1}(k)\mathbf{X}_e(k-1)e_e(k-1)$$

- Update data vector

$\mathbf{X}_e(k) = [x(k)\ x(k-1)\ \ldots\ x(k-M)\ y(k-1)\ y(k-2)\ \ldots\ y(k-N)]^T$ where $x(i) = y(i) = 0$ for $i < 0$(causal system)
- Calculate error of the Eq. $e_e(k) = y(k) - \mathbf{X}_e^T(k)\hat{\mathbf{H}}(k)$

3. Increment iteration counter k by 1 and repeat the procedure from the step 2

of the polynomial $1 - \mathbf{A}(k, z^{-1})$ in (2.104) are always within the unit circle of the z-complex plane, in order to ensure filter stability.

For the EE method one takes

$$F(k,z) = G(k,z) = 1, \tag{2.125}$$

so that

$$\mathbf{X}_F(k) = \mathbf{X}_e(k);\ e_G(k) = e_e(k). \tag{2.126}$$

The corresponding algorithm is the WRLS algorithm, and if one takes a unit matrix $\mathbf{I}$, for $\mathbf{R}(k+1)$ one obtains the LMS algorithm. The block diagram of the EE – WRLS algorithm is given in Table 2.3.

2.5.1 Recursive Prediction Error Algorithm (RPE Algorithm)

The recursive prediction error (RPE) algorithm updates the parameters of the vector $\mathbf{H}$ according to the process of minimum square error MSE, $\xi = \mathrm{E}\left[e_o^2(k)\right]$, where e_o is the output error. Since in general ξ is an unknown variable, the algorithm is designed to minimize in each iteration the estimated actual value of ξ, expressed as $\zeta = e_o^2(k)$, and the consequence of such approximation is a relatively noisy estimation of the filter parameters.

The proposed RPE algorithm updates the parameters of $\mathbf{H}(k)$ in the negative direction of the gradient of the criterion function $\zeta(k)$. Taking into account (2.102), the gradient of the criterion function

$$\zeta(k)=\frac{1}{2}e_0^2(k)$$

is

$$\nabla\zeta(k)=\frac{\partial\zeta(k)}{\partial\mathbf{H}(k)}=e_o(k)\nabla e_o(k)=-e_o(k)\nabla y_o(k), \tag{2.127}$$

where according to (2.103)

$$\nabla y_o(k)=\left[\frac{\partial y_o(k)}{\partial a_1(k)}\ \frac{\partial y_o(k)}{\partial a_2(k)}\ \cdots\ \frac{\partial y_o(k)}{\partial a_N(k)}\ \frac{\partial y_o(k)}{\partial b_0(k)}\ \frac{\partial y_o(k)}{\partial b_1(k)}\ \cdots\ \frac{\partial y_o(k)}{\partial b_M(k)}\right]^T. \tag{2.128}$$

The last term in (2.127) stems from the definition of the error of the output of OE and the fact that the reference signal $y(k)$ is independent on the values of the parameters of $\mathbf{H}(k)$.

Since according to (2.105),

$$\nabla y_o(k)=\nabla\left[\mathbf{H}^T(k)\mathbf{X}_o(k)\right],$$

and $\mathbf{X}_o(k)$ in (2.106) also contains previous values of the output, $y_0(k-i)$, which are a function of the previous estimations of the parameter parameter: vector $\mathbf{H}(k)$, an obvious dependence between $\mathbf{X}_o(k)$ and $\mathbf{H}(k)$ follows from it. According to the expression (2.103), one may write

$$\begin{aligned}\frac{\partial y_o(k)}{\partial a_j(k)}&=y_o(k-j)+\sum_{i=1}^{N}a_i(k)\frac{\partial y_o(k-i)}{\partial a_j(k)}\\ \frac{\partial y_o(k)}{\partial b_j(k)}&=x(k-j)+\sum_{i=1}^{N}a_i(k)\frac{\partial y_o(k-i)}{\partial b_j(k)}.\end{aligned} \tag{2.129}$$

If sufficiently small value is taken for the coefficient α in (2.119), which influences the convergence speed, so that the adaptation is sufficiently slow, one can then introduce the following approximation

$$\mathbf{H}(k)\approx\mathbf{H}(k-1)\approx\cdots\mathbf{H}(k-N+1), \tag{2.130}$$

i.e. one may neglect the mentioned dependence. This is acceptable in the majority of cases, especially if N is low, so that one may write

$$\frac{\partial y_o(k)}{\partial a_j(k)}=y_o(k-j)+\sum_{i=1}^{N}a_i(k)\frac{\partial y_o(k-i)}{\partial a_j(k)}=\left(\frac{1}{1-\mathbf{A}(k,z^{-1})}\right)y_o(k-j), \tag{2.131}$$

$$\frac{\partial y_o(k)}{\partial b_j(k)} = x(k-j) + \sum_{i=1}^{N} a_i(k) \frac{\partial y_o(k-i)}{\partial b_j(k)} = \left(\frac{1}{1 - \mathbf{A}(k, z^{-1})} \right) x(k-j), \quad (2.132)$$

where the polynomial $\mathbf{A}(k, z^{-1})$ is defined by (2.97). While deriving relations (2.131) and (2.132) the operator of unit delay z^{-1} was introduced, so that $\frac{\partial y_o(k-1)}{\partial a_j(k)} = z^{-1} \frac{\partial y_o(k)}{\partial a_j(k)}$ and $\frac{\partial y_o(k-1)}{\partial b_j(k)} = z^{-1} \frac{\partial y_o(k)}{\partial b_j(k)}$. It follows that in the general form of the algorithm (2.119)–(2.122)

$$F(k, z) = \frac{1}{1 - \mathbf{A}(k, z^{-1})}; \; G\left(k, z^{-1}\right) = 1. \quad (2.133)$$

Now it is possible to write the complete RPE algorithm

$$\hat{\mathbf{H}}(k+1) = \hat{\mathbf{H}}(k) + \alpha \mathbf{R}^{-1}(k+1) \left(\frac{1}{1 - \mathbf{A}(k, z^{-1})} \right) \mathbf{X}_o(k) e_o(k), \quad (2.134)$$

where $\mathbf{X}_o(k)$ is defined by the expression (2.106), $e_o(k)$ by (2.102) and (2.105), and $\mathbf{R}(k)$ by the expression (2.124), while the polynomial $\mathbf{A}(k)$ is given by the expression (2.97).

Relation (2.134) can be written in an alternative form

$$\hat{\mathbf{H}}(k+1) = \hat{\mathbf{H}}(k) + \alpha \mathbf{R}^{-1}(k+1) \mathbf{X}_{of}(k) e_o(k), \quad (2.135)$$

where

$$\mathbf{X}_{of}(k) = \frac{1}{1 - \mathbf{A}(k, z^{-1})} \mathbf{X}_o(k) = F\left(k, z^{-1}\right) \mathbf{X}_o(k) \quad (2.136)$$

is the filter vector $\mathbf{X}_o(k)$ of input–output data.

Starting from the definition (2.97) for the polynomial $\mathbf{A}(k, z^{-1})$, the coefficient of the polynomial

$$F(k, z^{-1}) = 1 + \sum_{i=1}^{N} f_i(k) z^{-i} \quad (2.137)$$

in (2.121) $(n = N)$ can be determined according to the relation [8]

$$f_i(k) = \sum_{j=1}^{i} a_j(k) f_{i-j}(k); \;\; i = 1, 2, \ldots, N; \;\; f_0 = 1. \quad (2.138)$$

In this way, the vector of filtered input–output data is defined by the expression

$$\mathbf{X}_{of}^T(k) = \left\{ x_f(k), \; x_f(k-1), \; \ldots, \; x_f(k-M), \; y_f(k-1), \; y_f(k-2), \; \ldots, y_f(k-N) \right\}, \quad (2.139)$$

where

$$x_f(j) = F\left(k, z^{-1}\right)x(j) = x(j) + \sum_{i=1}^{N} f_i(k)x(j-i);\ j = k,\ k-1,\ \ldots,\ k-M, \tag{2.140}$$

$$y_f(j) = F\left(k, z^{-1}\right)y_o(j) = y_o(j) + \sum_{i=1}^{N} f_i(k)y_o(j-i);\ j = k-1,\ \ldots,\ k-N. \tag{2.141}$$

Introducing the parameter vector

$$f^T(k) = \{f_1(k),\ f_2(k),\ \ldots,\ f_N(k)\} \tag{2.142}$$

and the vectors of input–output data

$$\mathbf{X}^T(j) = \{x(j-1),\ x(j-2),\ \ldots,\ x(j-N)\};\ \ j = k, k-1,\ \ldots,\ k-M, \tag{2.143}$$

$$\mathbf{Y}_o^T(j) = \{y_o(j-1),\ y_o(j-2),\ \ldots,\ y_o(j-N)\};\ \ j = k-1, k-2,\ \ldots,\ k-N, \tag{2.144}$$

the relations (2.140) and (2.141) can be written in the vectorial form

$$x_f(j) = x(j) + \mathbf{X}^T(j)f(k);\ j = k, k-1,\ \ldots, k-M, \tag{2.145}$$

$$y_f(j) = y_o(j) + \mathbf{Y}_o^T(j)f(k);\ j = k-1, k-2,\ \ldots, k-N. \tag{2.146}$$

If the estimations of parameters are close to their optimum values, the procedure can be simplified by filtering only the last input and output data instead of the whole sequence in (2.145) and (2.146), respectively.

The main disadvantage of the RPE algorithm is that the filter poles, also used to calculate the derivatives (2.131) and (2.132), may be located outside the unit circle in the complex z-plane, which implies the appearance of instability. If the poles remain longer in this region during the adaptation process, a possibility occurs that the algorithm will diverge. There is a possibility that the poles appear outside the unit circle, especially because of the noisy estimation of the gradient, since the approximation $\zeta(k)$ is used instead of the gradient $\xi(k)$. In order to avoid this, it is necessary to permanently monitor the system stability. One of the simplest tests to check stability is to inspect if $\sum_i |a_i| < 1$ in each iteration. However, there are cases, especially for large value of N, that this criterion is not satisfactory. On the other hand, there are tests to establish system instability with certainty, but computationally they are very complex [3].

In a majority of cases when the test shows that new parameters lead to instability, most often they are simply neglected, i.e. one takes that $\hat{\mathbf{H}}(k+1) = \hat{\mathbf{H}}(k)$. Naturally, this degrades the properties of the algorithm and makes the algorithm a non-robust one, since it may remain in that state for an indeterminate period of time. If the poles are located within the unit circle, the filter will be stable only if it represents a linear and time-invariant system. For the systems variable in time, such as the adaptive IIR filters, it is not sufficient only to follow the position of poles in discrete time intervals in order to have the realized system efficient in practical situations.

A block diagram of the RPE algorithm is given in Table 2.4.

Table 2.4 Flow diagram of RPE algorithm

1. Initialization
- $\hat{\mathbf{H}}(0) = 0;\ \mathbf{R}^{-1}(0) = \sigma^2\mathbf{I};\ \sigma^2 \ll 1$
- Generation of the sample of the input signal $x(0)$ and the reference signal $y(0)$
- Initial output error $e_o(0) = y(0) - y_o(0);\ y_o(0) = 0$
- Read in the forgetting factor $0.9 \le \rho \le 0.99$
- Calculation of the convergence factor $\alpha = 1 - \rho$
- Forming of the initial vector of filtered data $\mathbf{X}_{of}(0) = \mathbf{0}$

2. Assuming that $\hat{\mathbf{H}}(k-1)$, $e_o(k-1)$, $\mathbf{R}^{-1}(k-1)$ and $\mathbf{X}_{of}(k-1)$ are known, in each discrete moment of time $k = 1,\ 2,\ \ldots$, calculate:
- Gain matrix

$$\mathbf{R}^{-1}(k) = \frac{1}{\rho}\left(\mathbf{R}^{-1}(k-1) - \frac{\mathbf{R}^{-1}(k-1)\mathbf{X}_{of}(k-1)\mathbf{X}_{of}^T(k-1)\mathbf{R}^{-1}(k-1)}{\rho/\alpha + \mathbf{X}_{of}^T(k-1)\mathbf{R}^{-1}(k-1)\mathbf{X}_{of}(k-1)}\right)$$

- Filter coefficients

$$\hat{\mathbf{H}}(k) = \hat{\mathbf{H}}(k-1) + \alpha\mathbf{R}^{-1}(k)\mathbf{X}_{of}(k-1)e_o(k-1)$$

- Update data vector

$\mathbf{X}_o^T(k) = [x(k)\ x(k-1)\ \ldots\ x(k-M)\ y_o(k-1)\ y_o(k-2)\ \ldots\ y_o(k-N)]^T$ where $x(i) = y(i) = 0$ for $i < 0$(causal system)
- Calculate output error

$$e_o(k) = y(k) - \mathbf{X}_o^T(k)\hat{\mathbf{H}}(k) = y(k) - y_o(k)$$

- Calculate coefficients of the filter that filters $\mathbf{X}_o(k)$

$$f_i(k) = \sum_{j=1}^{N} \hat{\mathbf{H}}_{M+1+j}(k) f_{i-j}(k);\ \ f_0(k) = 1;\ \ i = 1, \ldots, N$$

- Form the vector of coefficients

$f^T(k) = [f_1(k)\ f_2(k)\ \ldots\ f_N(k)]$

Filter input and output data

$x_f(k) = x(k) + \mathbf{X}^T(k)f(k)$, where

$\mathbf{X}^T(k) = [x(k-1)\ldots x(k-N)];\ x(i) = 0$ for $i < 0$

$y_f(k-1) = y_o(k-1) + \mathbf{Y}_o^T(k)f(k)$, where

$\mathbf{Y}_0^T(k) = [y_0(k-2)\ldots y_0(k-1-N)];\ y_0(i) = 0$ for $i < 0$

Forming of the vector of filtered data

$\mathbf{X}_{of}^T(k) = \left[x_f(k)\ x_f(k-1)\ \ldots\ x_f(k-M)\ y_f(k-1)\ y_f(k-2)\ \ldots\ y_f(k-N)\right]^T$

where $x_f(i) = y_f(i) = 0$ for $i \le 0$

3. Increment iteration counter k by 1 and repeat the procedure from the step 2

2.5.2 Pseudo-Linear Regression (PLR) Algorithm

Pseudo-linear Regression (PLR) algorithm represents a simplification of the RPE algorithm by introducing

$$F(k,z) = G(k,z) = 1. \tag{2.147}$$

The algorithm itself may be expressed as

$$\hat{\mathbf{H}}(k+1) = \hat{\mathbf{H}}(k) + \alpha \mathbf{R}^{-1}(k+1)\mathbf{X}_o(k)e_o(k). \tag{2.148}$$

Here the gradient $\nabla y_o(k)$ is approximated by $\nabla y_o(k) \approx \mathbf{X}_o(k)$. The name of the algorithm stems from the fact that the output from the adaptive filter is a nonlinear function of the parameter $\mathbf{H}$, while in the algorithm itself when calculating the gradient (2.128) one neglects that $\mathbf{X}_o(k)$ is dependent on the parameters of $\mathbf{H}$. $\mathbf{X}_o(k)$ is also often denoted as the regression vector and is defined by expression (2.106), while the output signal $y_o(k)$ is defined by expression (2.105).

The PLR algorithm is very similar to the RLS algorithm, so their computational complexities are comparable and they are much lower than that of the RPE algorithm.

A disadvantage of this algorithm is that it does not have obligatory to converge to the minimum of the MSE criterion, except in the case when the polynomial in the denominator of the transfer function (2.104), denoted as $1 - \mathbf{A}(k, z^{-1})$, satisfies the Strictly Positive Real (SPR) condition; let us note that the discrete transfer

Table 2.5 Flow diagram of the PLR algorithm

1. Initialization
- $\hat{\mathbf{H}}(0) = 0;\ \mathbf{R}^{-1}(0) = \sigma^2\mathbf{I};\ \sigma^2 \ll 1$
- Generation of the sample of the input signal $x(0)$ and the reference signal $y(0)$
- Initial output error $e_o(0) = y(0) - y_o(0) = y(0)$
- Read in the forgetting factor $0.9 \leq \rho \leq 0.99$
- Calculation of the convergence factor $\alpha = 1 - \lambda$
- Forming of the initial vector of filtered data $\mathbf{X}_o(0) = [x(0)\ 0\ \ldots\ 0]$

2. Assuming that $\hat{\mathbf{H}}(k-1)$, $e_o(k-1)$, $\mathbf{R}^{-1}(k-1)$ and $\mathbf{X}_o(k-1)$ are known, in each discrete moment of time $k = 1,\ 2,\ \ldots$, calculate:
- Gain matrix

$$\mathbf{R}^{-1}(k) = \frac{1}{\rho}\left(\mathbf{R}^{-1}(k-1) - \frac{\mathbf{R}^{-1}(k-1)\mathbf{X}_o(k-1)\mathbf{X}_o^T(k-1)\mathbf{R}^{-1}(k-1)}{\rho/\alpha + \mathbf{X}_o^T(k-1)\mathbf{R}^{-1}(k-1)\mathbf{X}_o(k-1)}\right)$$

- Filter filter:coefficients

$$\hat{\mathbf{H}}(k) = \hat{\mathbf{H}}(k-1) + \alpha\mathbf{R}^{-1}(k)\mathbf{X}_o(k-1)e_o(k-1)$$

- Form data vector where $x(i) = y(i) = 0$ for $i < 0$(causal signals)
- Calculate output $y_o(k) = \hat{\mathbf{H}}^T(k)\mathbf{X}_o(k)$
- Calculate output error *OE*

$$e_o(0) = y(k) - y_o(k)$$

3. Increment counter k by 1 and repeat the procedure from the step 2

function $G(z^{-1})$ is denoted as SPR if $\mathrm{Re}\{G(e^{j\omega})\} > 0$ for $\forall\omega, -\pi < \omega < \pi$, where j is the imaginary unit. If not, the obtained results may be absolutely unacceptable [24, 28].

Contrary to the RPE algorithm, here it is not necessary to monitor stability during the parameter update. Because of that the PLR algorithm can be used in combination with the RPE algorithm. When RPE algorithm becomes unstable one adopts the PLR algorithm until the poles return to a stable area. In this way it is possible to improve the properties of the RPE algorithm, which will ignore the obtained results in the time intervals when the estimated poles are in the unstable area, until the stability criterion is satisfied (Table 2.5).

Let us note at the end that the theory of adaptive IIR filters is still insufficiently researched, since their analysis includes nonlinear systems of high order, and this too is a reason of their relatively narrow application. Prior analyses and computer simulations are often necessary to determine with certainty the properties of IIR adaptive algorithms [29, 30]. Thus the analysis and synthesis of the adaptive IIR filters in various tasks of processing and transfer of noise-contaminated signals still represents a subject matter with both theoretical and practical interest.

http://www.springer.com/978-3-642-33560-0

Adaptive Digital Filters

Kovačević, B.; Banjac, Z.; Milosavljević, M.

2013, XIV, 211 p. 66 illus., Hardcover
ISBN: 978-3-642-33560-0