3

Mathematical Background

Discrete signals are represented by sequences of values which implies the discrete system description by difference equations instead of differential equations for continuous signals. Resulting discrete system analysis and processing involves the following basic mathematical disciplines:

- *Z-transform* based upon the complex variable theory used for signal and system description
- theory of *difference equations* used for system representation
- *discrete Fourier transform* covering signal component analysis
- statistical methods including *stochastic processes* and *the least square method* fundamental for adaptive signal processing

Topics mentioned above are described in many special books and we shall summarize basic results only with notes to further detail references.

3.1 Z-transform and Signal Description

Z-transform is a mathematical tool closely connected with the theory of complex variable enabling compact signal and system description and giving possibility of its simple processing [23, p.76], [40, p.124], [42].

3.1.1 Definitions and Basic Properties

Definition 3.1 *The two-sided Z-transform of a sequence* $\{x(n)\}$ *is defined as*

$$Z[x(n)] = X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \tag{3.1}$$

in the complex plane of variable z.

In case of a causal sequence having $x(n) = 0$ for $n < 0$ the transform is reduced to one-sided only with summation in Def. 3.1 for $n = 0, 1, \cdots, \infty$. In both cases the *region of convergence* covers the set of those z values for which the summation has a finite value.

Example 3.1 *Evaluate the Z-transform of the causal exponential sequence with its region of convergence.*

Solution: Using the unit step function it is possible to express the exponential sequence

$$x(n) = a^n u(n) = \begin{cases} a^n & \text{for } n \geq 0 \\ 0 & \text{for } n < 0 \end{cases}$$

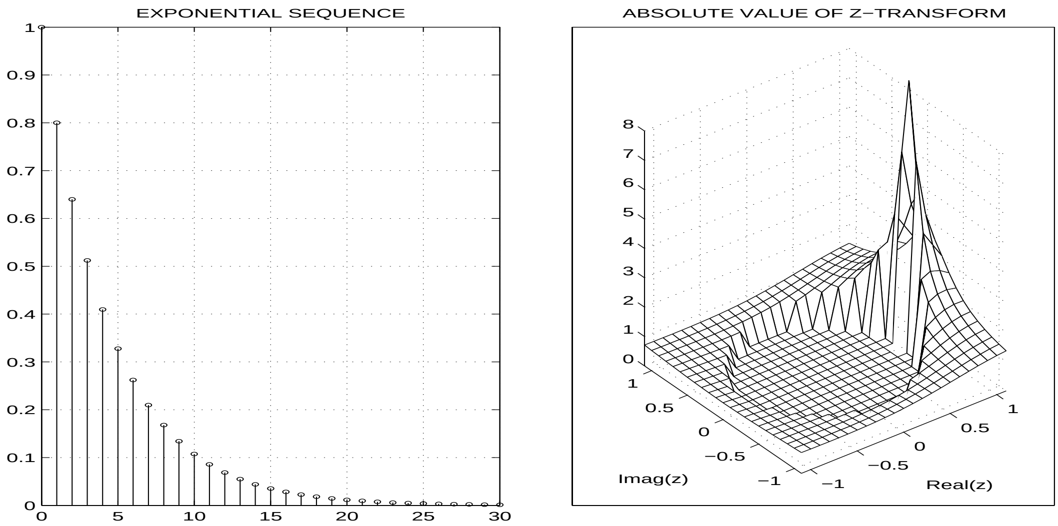

FIGURE 3.1. Exponential sequence $x(n) = a^n$ for $a = 0.8$ and absolute value of its Z-transform representation above the region of convergence for $|z| > |a|$ in the complex domain for $Re(z) \in \langle -1.1, 1.1\rangle$ and $Im(z) \in \langle -1.1, 1.1\rangle$

and to find

$$X(z) = \sum_{n=0}^{\infty} a^n z^{-n}$$

which represents the geometrical sequence having its value

$$X(z) = \frac{1}{1 - az^{-1}}$$

for quotient $|az^{-1}| < 1$ which implies $|z| > |a|$. Representation of original sequence and its Z-transform in the complex plane above the region of convergence is given in Fig. 3.1.

Using the Def. 3.1 it is possible to evaluate Z-transform of further sequences and the region of convergence as well. Some results are summarized in Tab. 3.1 presenting correspondence between original sequences $\{x(n)\}$ and their representation $X(z)$ in the complex plane. Advantages of such a transformation are obvious from the next section presenting possibilities of discrete system description and difference equation solution.

Example 3.2 *Evaluate the two-sided Z-transform of the exponential sequence*

$$x(n) = \begin{cases} a^n & \textit{for } n \geq 0 \\ b^n & \textit{for } n < 0 \end{cases}$$

Solution: Using the Def. 3.1 it is possible to find

$$X(z) = \sum_{n=-1}^{-\infty} b^n z^{-n} + \sum_{n=0}^{\infty} a^n z^{-n} = \sum_{n=1}^{\infty} b^{-n} z^n + \sum_{n=0}^{\infty} a^n z^{-n}$$

In case that quotients of these geometrical sequences are in absolute values less than one which means that

$$|b^{-1}z| < 1 \qquad \text{and} \qquad |az^{-1}| < 1$$

or

$$|a| < |z| < |b|$$

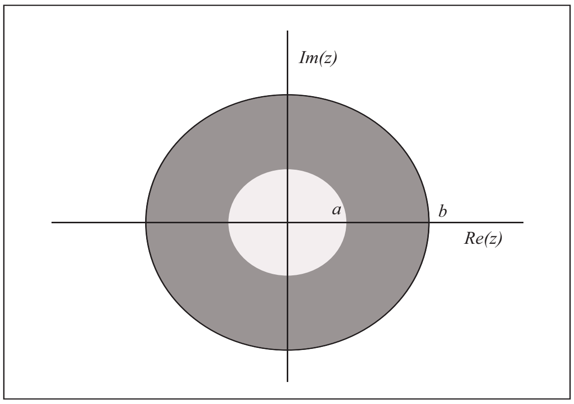

FIGURE 3.2. Region of convergence for the two-sided exponential sequence

it is possible to express the result in the form

$$X(z) = \frac{b^{-1}z}{1-b^{-1}z} + \frac{1}{1-az^{-1}} = -\frac{z}{z-b} + \frac{z}{z-a} = \frac{z(a-b)}{(z-a)(z-b)}$$

Region of convergence is given in Fig. 3.2. It is obvious that according to values $|a|$ and $|b|$ it can be empty as well.

Fundamental properties of the Z-transform can be stated in the following form

1. Linearity

$$Z[ax_1(n) + bx_2(n)] = aZ[x_1(n)] + bZ[x_2(n)] \tag{3.2}$$

2. Translation

$$Z[x(n)] = X(z) \quad \Rightarrow \quad Z[x(n-m)] = z^{-m}X(z) \tag{3.3}$$

3. Convolution in time domain

$$Z[\sum_{k=-\infty}^{\infty} x(k)y(n-k)] = Z[x(n)] \cdot Z[y(n)] \tag{3.4}$$

4. Initial value theorem (for causal sequences)

$$x(0) = \lim_{z\to\infty} X(z) \tag{3.5}$$

Proofs of these properties result from the Def. 3.1.

Sequence	*Definition*	*Z-transform*	*Region of convergence*
Unit sample	$d(n) = \begin{cases} 1 & \text{for } n = 0 \\ 0 & \text{for } n \neq 0 \end{cases}$	1	all z
Unit step	$u(n) = \begin{cases} 1 & \text{for } n \geq 0 \\ 0 & \text{for } n < 0 \end{cases}$	$\frac{z}{z-1}$	$\|z\| > 1$
Exponential	$x(n) = a^n u(n)$	$\frac{z}{z-a}$	$\|z\| > \|a\|$
Harmonic	$x(n) = \sin(2\pi f n)$	$\frac{z\sin(2\pi f)}{z^2 - 2z\cos(2\pi f) + 1}$	$\|z\| > 1$
	$x(n) = \cos(2\pi f n)$	$\frac{z^2 - z\cos(2\pi f)}{z^2 - 2z\cos(2\pi f) + 1}$	$\|z\| > 1$

TABLE 3.1. Basic sequences and their Z-transform

3.1.2 Inverse Z-transform

Polynomial $X(z)$ defined by Eq. (3.1) is determined by the complete sequence $x(n)$ and it enables its reconstruction as well confirming in this way the equivalence between the sequence definition in time and complex domains. This process of the inverse Z-transform can be performed in several ways.

The application of COMPLEX INVERSION INTEGRAL is based upon the complex variable theory [23, p.76], [42], [12, p.770] enabling the derivation of the Z-transform inversion formula in the form

$$x(n) = \frac{1}{2\pi j} \oint_C X(z) \frac{z^n}{z} dz \tag{3.6}$$

with C representing the closed contour laying inside the region of convergence.

As the Z-transform definition usually results in the rational representation of $X(z)$ the PARTIAL FRACTION EXPANSION METHOD may be used to express the original function as a sum of simple fractions in the following way.

1. Evaluation of poles $p_0, p_1, ..., p_N$ of

   $$X(z) = \frac{b_0 z^N + \cdots + b_N}{a_0 z^N + \cdots + a_N} \tag{3.7}$$

   with some possible zero coefficients and unequal order of numerator and denominator polynomials.

2. Partial fraction expansion of function $X(z)/z$ based upon the knowledge that z appears in the numerators of functions $X(z)$

   $$\frac{X(z)}{z} = \frac{c_0}{z - p_0} + \frac{c_1}{z - p_1} + \cdots + \frac{c_N}{z - p_N} + (k_1 + k_2 z + \cdots) \tag{3.8}$$

   and evaluation of coefficients $c_0, c_1, \cdots, c_N$. Direct terms with coefficients $k_1, k_2, \cdots$ appear for non-proper fractions only. As complex poles are in complex conjugate pairs they can be combined into second order terms before further processing.

3. Using expression

   $$X(z) = \frac{c_0 z}{z - p_0} + \cdots + \frac{c_N z}{z - p_N} + (k_1 z + k_2 z^2 + \cdots) \tag{3.9}$$

   with possible second order terms we can use the Z-transform table in connection with the knowledge of basic properties and the region of convergence to find the original sequence.

Example 3.3 *Evaluate the causal sequence $x(n)$ having its Z-transform in the form*

$$X(z) = \frac{0.3z}{z^2 - 0.7z + 0.1}$$

Solution: As

$$\frac{X(z)}{z} = \frac{0.3}{(z - 0.2)(z - 0.5)} = \frac{c_0}{z - 0.2} + \frac{c_1}{z - 0.5}$$

it is possible to find coefficients c_0, c_1 from the following equation

$$0.3 = c_0(z - 0.5) + c_1(z - 0.2)$$

The previous equation must be valid for all z which implies that

$$\begin{aligned} 0.3 &= -0.5c_0 - 0.2c_1 \\ 0 &= c_0 + c_1 \end{aligned}$$

giving solution $c_0 = -1, c_1 = 1$. As further

$$X(z) = -\frac{z}{z - 0.2} + \frac{z}{z - 0.5}$$

the Tab. 3.1 enables evaluation of

$$x(n) = -(0.2)^n u(n) + (0.5)^n u(n)$$

Computer processing of the partial fraction expansion method may be based upon the procedure presented in Alg. 3.1. For realization in other languages than MATLAB compact functions presented here must be realized in other way and in most cases by special subroutines.

Algorithm 3.1 *Evaluation of the partial fraction expansion for the rational function*

$$X(z) = \frac{[b(0) \cdots b(N)][z^N \cdots 1]'}{[a(0) \cdots a(N)][z^N \cdots 1]'} \tag{3.10}$$

- definition of vectors $\mathbf{b}$ and $\mathbf{a}$ of the rational function
- evaluation of vectors $\mathbf{c}, \mathbf{p}$ and $\mathbf{k}$ of expansion

  $$X(z) = \frac{c_0 z}{z - p_0} + \cdots + \frac{c_N z}{z - p_N} + (k_1 z + k_2 z^2 + \cdots)$$

  by function

  $[\mathbf{c}, \mathbf{p}, \mathbf{k}] = \text{residue } (\mathbf{b}, \mathbf{a})$
- as the inverse procedure can be realized by function

  $[\mathbf{b}, \mathbf{a}] = \text{residue } (\mathbf{c}, \mathbf{p}, \mathbf{k})$

  it can be used for connection of terms with complex conjugate poles to form second order terms of partial fraction expansion with real coefficients only to enable the following use of the Z-transform tables

Division method provides possibility of evaluation of individual members of the original sequence based upon the knowledge of its Z-transform $X(z)$ and the region of convergence. In case of causal sequences the expansion of $X(z)$ can be restricted to non-positive powers of z in the form

$$X(z) = x(0)z^0 + x(1)z^{-1} + x(2)z^{-2} + \cdots \tag{3.11}$$

with coefficients $x(0), x(1), \cdots$ defining the desired sequence.

Example 3.4 *Evaluate the inverse Z-transform of*

$$X(z) = \frac{0.3z}{z^2 - 0.7z + 0.1}$$

for region of convergence defined by $|z| > 0.5$.

Solution: Dividing the rational function $X(z)$ we obtain

$$\begin{array}{l}
+0.3z \qquad\qquad\qquad : (z^2 - 0.7z + 0.1) = 0.3z^{-1} + 0.21z^{-2} + 0.117z^{-3} + \cdots \\
\underline{\pm 0.3z \mp 0.21 \pm 0.030z^{-1}} \\
\qquad + 0.21 - 0.030z^{-1} \\
\qquad \underline{\pm 0.21 \mp 0.147z^{-1} \pm 0.0210z^{-2}} \\
\qquad\qquad + 0.117z^{-1} - 0.0210z^{-2} \\
\qquad\qquad \underline{\pm 0.117z^{-1} \mp 0.0890z^{-2} \pm 0.0117z^{-3}} \\
\qquad\qquad\qquad + 0.0609z^{-2} - 0.0112z^{-3}
\end{array}$$

The desired sequence has the following values:

$$\begin{aligned}
x(n) &= 0 \qquad \text{for } n \le 0 \\
x(1) &= 0.3 \\
x(2) &= 0.21 \\
&\cdots
\end{aligned}$$

Results of the example evaluated by Alg. 3.2 are presented in Fig. 3.3.

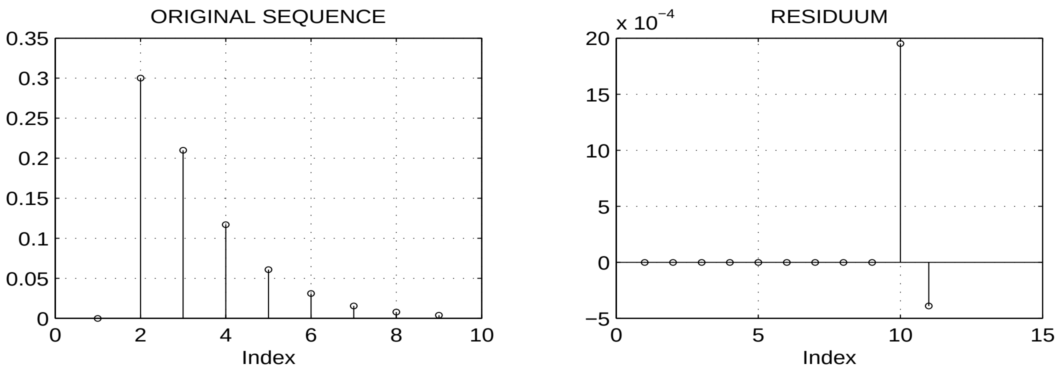

FIGURE 3.3. Sequence {x(n)} for $n = 1, \cdots, L+1$ evaluated as the inverse Z-transform to $X(z) = (0.3z)/(z^2 - 0.7z + 0.1)$ for $L = 8$ and the residuum sequence

3.2 Difference Equations and System Modelling

The linear shift invariant discrete system is an essential mathematical structure for the approximation of most continuous real systems. It can be used for their modelling, analysis and signal processing as well.

3.2.1 System Representation

The description of the linear shift invariant system can be given by the *difference equation* in the general form

$$y(n) + \sum_{k=1}^{N} a(k)y(n-k) = \sum_{k=0}^{N} b(k)x(n-k) \tag{3.13}$$

with some possible zero coefficients. This time domain representation can be further modified to enable more convenient ways of digital signal processing.

The *discrete transfer function* (system function) can be derived from the Z-transform of the difference Eq. (3.13) resulting in relation

$$Z[y(n)] + \sum_{k=1}^{N} a(k) Z[y(n-k)] = \sum_{k=0}^{N} b(k) Z[x(n-k)]$$

Using the translation property of the Z-transform we obtain

$$Y(z) + \sum_{k=1}^{N} a(k) z^{-k} Y(z) = \sum_{k=0}^{N} b(k) z^{-k} X(z)$$

and the transfer function in the form

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N} b(k) z^{-k}}{1 + \sum_{k=1}^{N} a(k) z^{-k}} \tag{3.14}$$

This representation enables simple evaluation of the system output in the following steps

- description of the input sequence $\{x(n)\}$ in the form of its Z-transform X(z)
- application of the transfer function $H(z)$ for evaluation of the output sequence Z-transform
$$Y(z) = H(z)X(z) \tag{3.15}$$
- evaluation of the output sequence $\{y(n)\}$ by the inverse Z-transform of Y(z)

Any method of the inverse Z-transform can be used in this stage.

Algorithm 3.2 *Polynomial division for the evaluation of the sequence* $\{x(n)\}$ *representing the inverse Z-transform in the form*

$$\begin{aligned} X(z) &= \frac{[b(0) \cdots b(N)][z^N \cdots 1]'}{[a(0) \cdots a(N)][z^N \cdots 1]'} \qquad (3.12) \\ &= [x(0) \cdots x(L)][z^0 z^{-1} \cdots z^{-L}]' + \frac{[r(L+1) \cdots r(L+N)][z^{N-L-1} \cdots z^{-L}]'}{[a(0) \cdots a(N)][z^N \cdots 1]'} \end{aligned}$$

- definition of vectors $\mathbf{d}$ and $\mathbf{a}$ where $\mathbf{d} = [\mathbf{b}, zeros(1, L)]$ represents a new numerator vector after the multiplication of the whole expression by z^L to enable expansion of the newly defined non-proper fraction to vector $\mathbf{x}$ connected with non-negative powers of $\mathbf{z}$ and the remainder vector $\mathbf{r}$.
- evaluation of values $[x(0), \cdots, x(L)]$ by function
  $[\mathbf{x}, \mathbf{r}] = \text{decom}\ (\mathbf{d}, \mathbf{a})$
- possible graphic representation of the evaluated sequence by function
  plot($\mathbf{x}$)

The *unit sample response* represents another possibility of system description. As $X(z) = 1$ for such a sequence the Z-transform of system output can be evaluated using (3.15) resulting in $Y(z) = H(z)$ which implies that the inverse Z-transform of $H(z)$ stands for the unit sample response $\{h(n)\}$. The transfer function for causal system is then defined by relation

$$H(z) = \sum_{n=0}^{\infty} h(n) z^{-n} \tag{3.16}$$

and it is equivalent to that defined by Eq. (3.14). As $H(z) = Y(z)/X(z)$ it is obvious that any input sequence $\{x(n)\}$ implies system output $\{y(n)\}$ in the form

$$y(n) = \sum_{k=0}^{\infty} h(k) x(n-k) = h(n) * x(n) \tag{3.17}$$

referred as convolution of sequences $\{h(n)\}$ and $\{x(n)\}$.

The system *frequency response* $H(e^{j\omega})$ can be evaluated after the application of the input sequence $x(n) = e^{j\omega n}$ to the system described by difference Eq. (3.13) or (3.17). Using Eq. (3.17) it is obvious that

$$y(n) = \sum_{k=0}^{\infty} h(k) e^{j\omega(n-k)} = e^{j\omega n} \sum_{k=0}^{\infty} h(k) e^{-j\omega k} = x(n) \sum_{k=0}^{\infty} h(k) e^{-j\omega k} \tag{3.18}$$

Comparing Eq. (3.16) and (3.18) it is possible to express the frequency response

$$H(e^{j\omega}) = H(z)|_{z=e^{j\omega}} \tag{3.19}$$

having its magnitude and phase part.

Results described above imply the basic role of the transfer function $H(z)$ in form of Eq. (3.14) or (3.16) enabling the difference equation or frequency response evaluation.

Example 3.5 *Use the transfer function*

$$H(z) = 0.2 \frac{z+1}{z^2 - z + 0.5} \tag{3.20}$$

of a causal system to evaluate its difference equation, unit sample response and frequency response.

Solution:

- As
  $$H(z) = \frac{Y(z)}{X(z)} = 0.2 \frac{z+1}{z^2 - z + 0.5} = 0.2 \frac{z^{-1} + z^{-2}}{1 - z^{-1} + 0.5 z^{-2}}$$
  we can find after the cross multiplication that
  $$Y(z)(1 - z^{-1} + 0.5 z^{-2}) = 0.2 X(z)(z^{-1} + z^{-2})$$
  which after the inverse Z-transform results in the difference equation
  $$y(n) - y(n-1) + 0.5 y(n-2) = 0.2(x(n-1) + x(n-2))$$
- One of possibilities how to evaluate the unit sample response is to use the transfer function to obtain
  $$Y(z) = H(z)\, X(z)$$
  Taking into account the unit sample Z-transform $X(z) = 1$ it is possible to use the division method to evaluate separate terms of $\{h(n)\}$. As

$$
\begin{array}{lll}
(+0.2z + 0.2) & & : (z^2 - z + 0.5) = 0.2z^{-1} + 0.4z^{-2} + 0.3z^{-3} \\
\underline{-0.2z \mp 0.2 \pm 0.1z^{-1}} & & \\
\quad +0.4 - 0.1z^{-1} & & \\
\quad \underline{-0.4 \mp 0.4z^{-1} \pm 0.2z^{-2}} & & \\
\qquad +0.3z^{-1} - 0.2z^{-2} & & \\
\qquad \underline{-0.3z^{-1} \mp 0.3z^{-2} \pm 0.15z^{-3}} & & \\
\qquad\qquad +0.1z^{-2} - 0.15z^{-3} & &
\end{array}
$$

the resulting sequence $\{h(n)\}_{n=0}^{\infty}$ has values $\{0, 0.2, 0.4, 0.3, \cdots\}$.

- The frequency response can be evaluated using Eq. (3.19) in form

$$H(e^{j\omega}) = H(z)\,|_{z=e^{j\omega}} = 0.2\frac{e^{j\omega}+1}{e^{2j\omega} - e^{j\omega} + 0.5}$$

After application of Euler relations it is possible to write

$$H(e^{j\omega}) = 0.2\frac{1 + \cos(\omega) + j\ sin(\omega)}{(\cos(2\omega) - \cos(\omega) + 0.5) + j(\sin(2\omega) - \sin(\omega))}$$

The magnitude and phase of this frequency response for $\omega \in \langle 0, \pi \rangle$ is presented in Fig. 3.4 together with the sketch of the given transfer function representation in the complex plane showing its poles and values on the unit circle for $z = e^{j\omega}$.

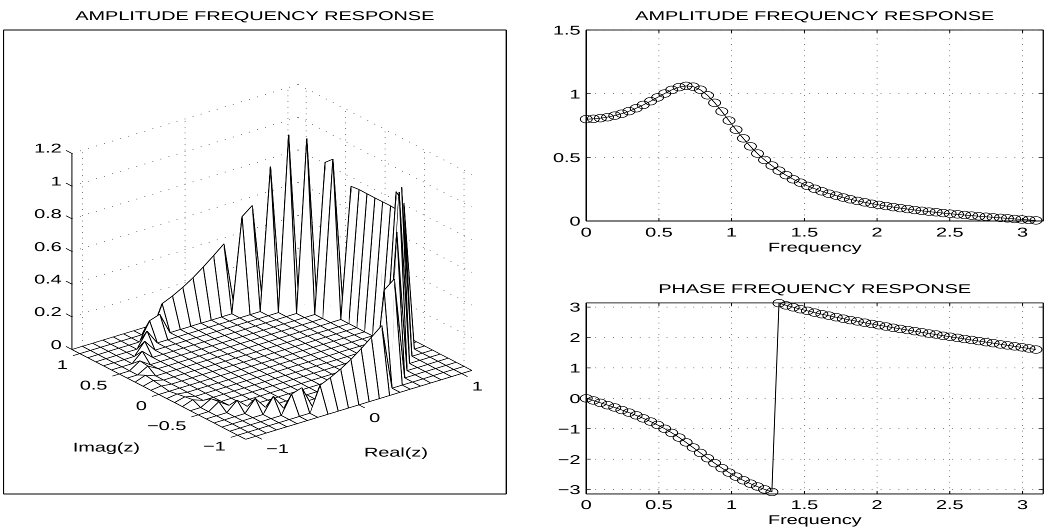

FIGURE 3.4. Magnitude and phase frequency response of the discrete system with the transfer function $H(z) = 0.2(z + 1)/(z^2 - z + 0.5)$ and its sketch in the complex plane.

Frequency response provides a very important information concerning the system behaviour with respect to the input signal frequency components. Computer processing of the system response and frequency response based upon the knowledge of the discrete transfer function can be summarized in Algorithm 3.3 and 3.4.

3.2.2 Linear Constant Coefficients Difference Equations

The *classical solution* of difference equations is very close to methods of solution of differential equations and it involves the estimation of the particular and homogenous solution as well [36, p.16].

Algorithm 3.3 *System response evaluation for the transfer function*

$$H(z) = \frac{[b(0), b(1) \cdots b(N)][1, z^{-1} \cdots z^{-N}]'}{1 + [a(1) \cdots a(N)][z^{-1} \cdots z^{-N}]'} \tag{3.21}$$

to the input sequence

$$\mathbf{x} = [x(0), x(1) \cdots]$$

- definition of vectors $\mathbf{b}, \mathbf{a}$ and $\mathbf{x}$.
- system output evaluation by function
  $\mathbf{y}$ = filter($\mathbf{b}, \mathbf{a}, \mathbf{x}$)
- possible graphic output of the original and evaluated sequence (with two pictures on the screen)
  clg; subplot(211);
  plot($\mathbf{x}$); plot($\mathbf{y}$)

The *Z-transform method* provides another possibility of a very simple way for solution of the equation

$$y(n) + \sum_{k=1}^{N} a(k)y(n-k) = f(n) \tag{3.23}$$

with a given set of initial conditions $\{y(-1), y(-2), \cdots, y(-N)\}$. The solution consists in principle of the following steps

- Z-transform application which transforms the difference equation into an algebraic equation
- evaluation of $Y(z)$ standing for the Z-transform of the solution
- inverse Z-transform application for evaluation of $\{y(n)\}$.

Algorithm 3.4 *Frequency response evaluation of system defined by its transfer function*

$$H(z) = \frac{[b(0), b(1) \cdots b(N)][1, z^{-1} \cdots z^{-N}]'}{1 + [a(1) \cdots a(N)][z^{-1} \cdots z^{-N}]'} \tag{3.22}$$

- definition of vectors $\mathbf{b}$ and $\mathbf{a}$.
- frequency response evaluation by function
  $[\mathbf{h}, \mathbf{w}]$ =freqz($\mathbf{b}, \mathbf{a}, n$)
  in n points between 0 and π defined in vector $\mathbf{w}$ with result in vector $\mathbf{h}$.
- possible separate plots of magnitude and phase of the frequency response (with two pictures on the screen)
  clg; subplot(211);
  plot($\mathbf{w}$, abs($\mathbf{h}$)); plot($\mathbf{w}$, angle($\mathbf{h}$))

The Z-transform of real causal sequence $\{y(n)u(n)\}$ can be defined as $Y(z)$. To obtain the Z-transform of the delayed truncated sequence we can evaluate

$$\begin{aligned} Z[y(n-k)u(n)] &= \sum_{n=-\infty}^{\infty} y(n-k)u(n)z^{-n} = \sum_{n=0}^{\infty} y(n-k)z^{-n} = \sum_{m=-k}^{\infty} y(m)z^{-(m+k)} = \\ &= \sum_{m=-k}^{-1} y(m)z^{-(m+k)} + z^{-k}\sum_{k=0}^{\infty} y(m)z^{-m} \end{aligned}$$

which implies that

$$Z[y(n-k)u(n)] = \sum_{m=-k}^{-1} y(m)z^{-(m+k)} + z^{-k}Y(z)$$

with the first term enabling to apply initial conditions of Eq. (3.23).

Example 3.6 *Evaluate the solution of the following linear constant difference equation*

$$y(n) - 0.5y(n-1) = 0.25^n$$

for $y(-1) = 1$.

Solution: After the Z-transform we shall receive

$$Y(z) - 0.5(y(-1) + z^{-1}Y(z)) = \frac{1}{1 - 0.25z^{-1}}$$

which implies that

$$Y(z) = \frac{\frac{1}{1-0.25z^{-1}} + 0.5}{1 - 0.5z^{-1}} = \frac{1.5 - 0.125z^{-1}}{(1 - 0.5z^{-1})(1 - 0.25z^{-1})} = \frac{1.5z^2 - 0.125z}{(z - 0.5)(z - 0.25)}$$

Using the partial fraction expansion method we obtain

$$Y(z) = \frac{2.5z}{z - 0.5} - \frac{z}{z - 0.25}$$

which implies the solution in the following form

$$y(n) = 2.5(0.5)^n - (0.25)^n$$

3.3 Discrete Fourier Transform and Signal Decomposition

Discrete signals and systems described by difference equations can be represented by Z-transform enabling their simple analysis and further manipulation. Another way of signal processing is based upon its decomposition into a linear combination of basis functions [23, p.257]. In linear time invariant system various methods can be then applied separately to signal components and results composed again. This method is essential in many engineering applications enabling signal analysis, filtering of signal parts, adaptive signal processing etc.

Physical bases of many signals enable their harmonic decomposition which implies that the weighted sum of complex exponentials is used very often. Therefore the discrete Fourier transform based upon the Fourier series for periodic signals is an essential mathematical tool for the theoretical analysis of many digital signal processing methods and it enables their implementation using an efficient algorithm of the fast Fourier transform [7] as well.

3.3.1 Definition and Basic Properties

To explain the definition of the discrete Fourier transform we can start with representation of periodic discrete-time signal $\{x(n)\}$ with period N by the weighted sum of complex exponentials in the form

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) e^{jk\frac{2\pi}{N}n} \tag{3.24}$$

for $n = 0, 1, \cdots, N-1$. This expression is in close connection with Fourier series applied to continuous signals for the infinitive sum reduced to the finite sum of N terms only caused by N distinct exponentials for frequencies

$$\omega_k = k\frac{2\pi}{N}, \qquad k = 0, 1, \cdots, N-1.$$

The multiplying constant $1/N$ in Eq. (3.24) has no substantial effect in this stage. To evaluate terms $X(k)$ we can multiply both sides of Eq. (3.24) by $e^{-jl(2\pi/N)n}$ and to sum over $n = 0, 1, \cdots, N-1$ to obtain after the interchange of the summation order [30, p.88]

$$\sum_{n=0}^{N-1} x(n) e^{-jl\frac{2\pi}{N}n} = \frac{1}{N}\sum_{k=0}^{N-1} X(k) \sum_{n=0}^{N-1} e^{j(k-l)\frac{2\pi}{N}n}$$

Relation

$$\frac{1}{N}\sum_{k=0}^{N-1} e^{j(k-l)\frac{2\pi}{N}n} = \begin{cases} 1 & \text{for } k-l = mN \\ 0 & \text{for } k-l \neq mN \end{cases}$$

implies that

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-jk\frac{2\pi}{N}n} \tag{3.25}$$

This result can be also applied to finite sequences of N samples in case that we define the periodic sequence based upon the periodic extension of original values. It is often assumed that the nonzero period is for $n \in \langle 0, N-1 \rangle$. The discrete Fourier transform is then defined by the following relations.

Definition 3.2 *Let us assume the finite sequence $\{x(n)\}$ for $n = 0, 1, \cdots, N-1$. Its discrete Fourier transform is then defined by relation*

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-jk\frac{2\pi}{N}n} \tag{3.26}$$

for $k = 0, 1, \cdots, N-1$.

The inverse transform can be evaluated by relation

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) e^{jk\frac{2\pi}{N}n} \tag{3.27}$$

for $n = 0, 1, \cdots, N-1$ and discrete frequencies $\omega_k = k2\pi/N$. Relation (3.26) defines in fact coefficients of Eq. (3.27) related to separate frequency components.

Example 3.7 *Evaluate the discrete Fourier transform of a given sequence*

$$x(n) = \begin{cases} 1 & \text{for } 0 \le n \le 2 \\ 0 & \text{for } 3 \le n \le 8 \end{cases} \tag{3.28}$$

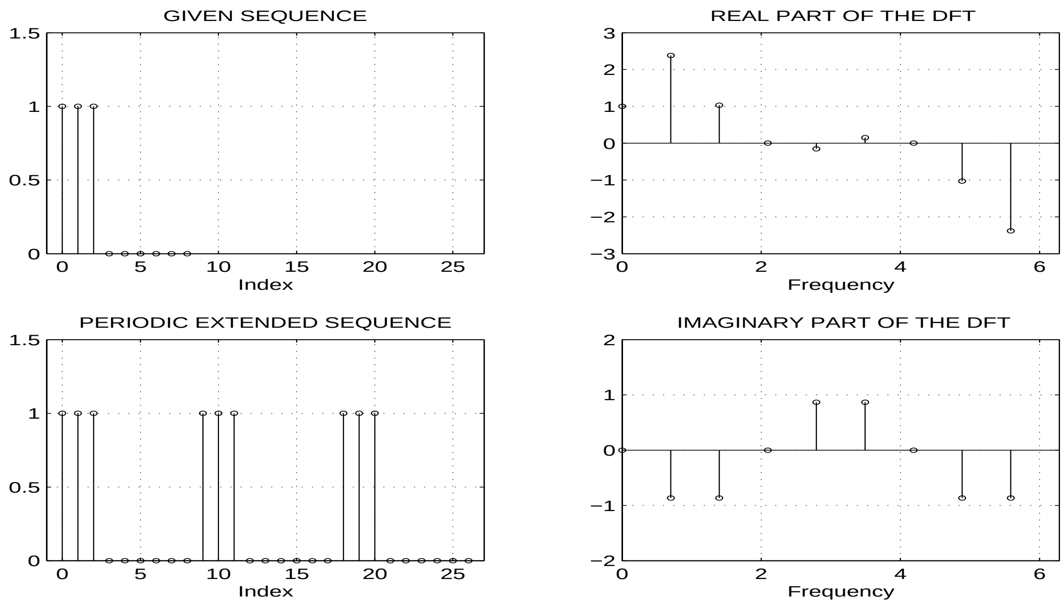

FIGURE 3.5. Discrete Fourier transform of a given sequence

Solution: Using the Def. 3.2 we can write

$$X(k) = \sum_{n=0}^{9} x(n) e^{-jk(2\pi/9)n} = \sum_{n=0}^{2} e^{-jk(2\pi/9)n}$$

which is a geometrical sequence implying

$$X(k) = \frac{1 - e^{-jk(2\pi/9)3}}{1 - e^{jk(2\pi/9)}} = \frac{e^{-jk(\pi/3)}}{e^{-jk(\pi/9)}} \frac{e^{jk(\pi/3)} - e^{-jk(\pi/3)}}{e^{jk(\pi/9)} - e^{-jk(\pi/9)}}$$

Using Euler relations we shall receive

$$X(k) = e^{-jk(2\pi/9)} \frac{\sin(k(\pi/3))}{\sin(k(\pi/9))} \tag{3.29}$$

Graphical representation of results in presented in Fig. 3.5.

Computer processing of the discrete Fourier transform can be based upon Algorithm 3.5 using simple MATLAB notation.

Discrete Fourier transform is closely related to the *Z-transform* which implies similar properties of both transforms as well. As the finite length sequence $\{x(n)\}$ for $n = 0, 1, \cdots, N-1$ has its Z-transform according to the definition in the form

$$X(z) = \sum_{n=0}^{N-1} x(n) z^{-n} \tag{3.30}$$

the comparison of Eqs. (3.30) and (3.26) results in relation

$$X(k) = X(z) \mid_{z=e^{jk2\pi/N}} \tag{3.31}$$

for $k = 0, 1, \cdots, N-1$. This result implies that the discrete Fourier transform represents equidistant values of $X(z)$ on the unit circle in the complex plane [30, p.90].

Example 3.8 *Evaluate the discrete Fourier transform of the exponential sequence*

$$x(n) = \begin{cases} a^n & \textit{for } n = 0, 1, \cdots, N-1 \\ 0 & \textit{for } n < 0 \textit{ and } n \geq N \end{cases}$$

Algorithm 3.5 *Evaluation of the direct and inverse discrete Fourier transform of sequence* $\{x(n)\}$, $n = 0, 1, \cdots, N-1$.

- definition of vector $\mathbf{x} = [x(0), \cdots, x(N-1)]$
- discrete Fourier transform evaluation
  $\mathbf{X} = \text{fft}(\mathbf{x})$
- graphic separate representation of the real and imaginary part
  subplot(211)
  plot((0 : $N-1$)./N, real($\mathbf{X}$))
  plot((0 : $N-1$)./N, imag($\mathbf{X}$))
- inverse discrete Fourier transform evaluation
  $\mathbf{y} = \text{ifft}(\mathbf{X})$

Solution: As

$$X(z) = \sum_{n=0}^{N-1} a^n z^{-n} = \frac{1-(az^{-1})^N}{1-az^{-1}} = \frac{z^N - a^N}{z^N - az^{N-1}}$$

it is possible to evaluate

$$X(k) = X(z) \mid_{z=e^{jk2\pi/N}} = \frac{e^{jk2\pi} - a^N}{e^{jk2\pi} - ae^{jk2\pi(N-1)/N}} = \frac{1-a^N}{1-ae^{jk2\pi(N-1)/N}}$$

for $k = 0, 1, \cdots, N-1$. The geometrical view presenting real and imaginary part of the discrete Fourier transform and its absolute value as a special case of the Z-transform on the unit circle in the complex plane for $N = 24$ discrete frequencies is given in Fig. 3.6. Separate plots of real and imaginary parts of the discrete Fourier transform are presented in Fig. 3.7 in connection with the complex plane interpretation again.

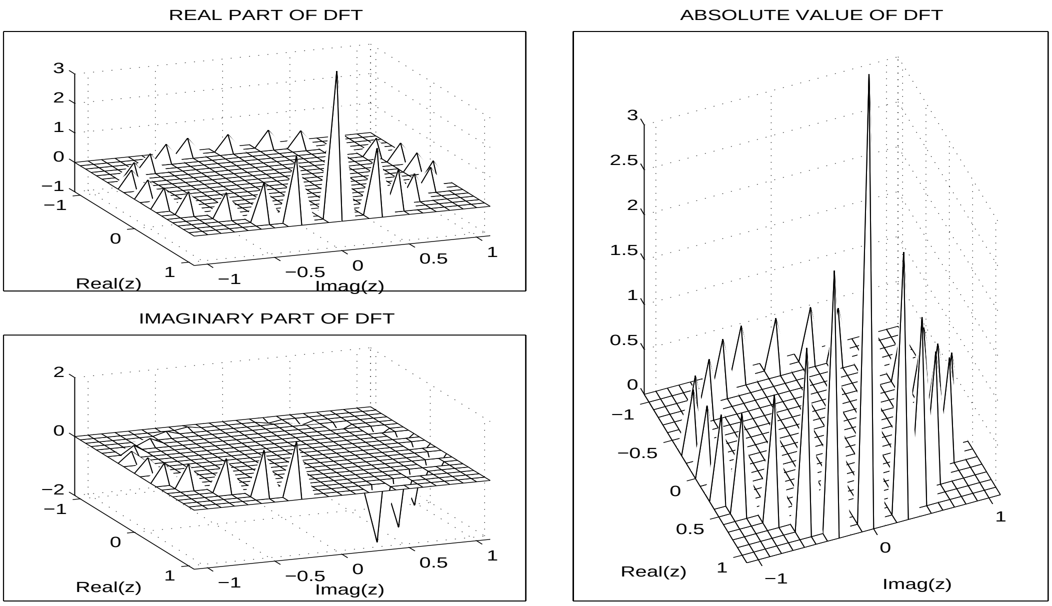

FIGURE 3.6. The discrete Fourier transform of exponential sequence related to its Z-transform in the complex plane

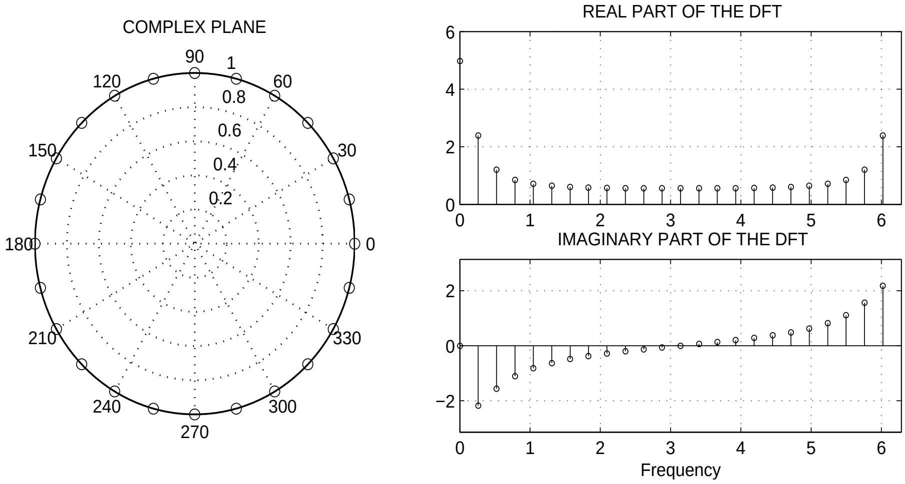

FIGURE 3.7. The real and imaginary part of the DFT of the exponential sequence in connection with the complex plane interpretation for $\omega_k = k2\pi/N, k = 0, 1, \cdots, N-1$ (for $N = 24$)

The graphic interpretation of the discrete Fourier transform given in the previous example enables better understanding of the frequency axis description given in Fig. 3.8 and it presents its *symmetry properties* as well. As terms $e^{jk2\pi/N}$ and $e^{j(N-k)2\pi/N}$ for $k = 0, 1, \cdots, N$ are complex conjugates the Eq. (3.26) implies that $X(k)$ and $X(N-k)$ are for real values of $\{x(n)\}$ in the same relation [39, p.252] which means that

- $real(X(k))$ is an even function in such a sense that $real(X(k)) = real(X(N-k))$
- $imag(X(k)$ is an odd function in such a sense that $imag(X(k)) = -imag(X(N-k))$
- $abs(X(k))$ is an even function

The definition of even and odd function is based upon the periodic extension of the analysed values. It is obvious that owing to this properties it is sufficient to evaluate $X(k)$ for $k = 0, 1, \cdots, N/2$ only.

Further *fundamental properties* of the discrete Fourier transform of a sequence $\{x(n)\}, n = 0, 1, \cdots, N-1$ can be stated in the following form [39, p.258].

1. Linearity
$$DFT[a_1 x_1(n) + a_2 x_2(n)] = a_1 DFT[x_1(n)] + a_2 DFT[x_2(n)] \tag{3.32}$$
2. Translation
$$DFT[x(n] = X(k) \quad \Rightarrow \quad DFT[x(n-m)] = e^{-jkm\frac{2\pi}{N}} X(k) \tag{3.33}$$
3. Convolution in time domain
$$DFT[\sum_{k=0}^{N-1} x(k) y(n-k)] = DFT[x(n)] \cdot DFT[y(n)] \tag{3.34}$$

Proofs of these properties result from the Def. 3.2.

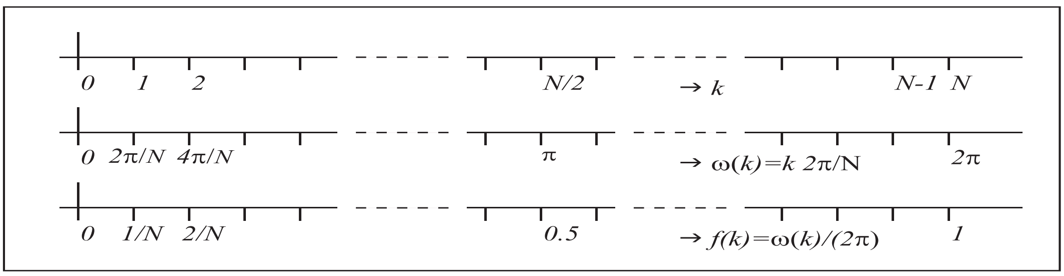

FIGURE 3.8. Frequency axis interpretation

3.3.2 Fast Fourier transform

Definition of the discrete Fourier transform (DFT) enables the estimation of basic numerical calculations of this method reaching the order of N^2 for complex multiplications and additions. The fast Fourier transform (FFT) algorithm reduces the required number of arithmetic operations to the order of $(N/2)log_2(N)$ which for $N = 512$ means the approximate reduction to 1% of the original value connected with time requirements as well.

Let us assume sequence $x(n)_{n=0}^{N-1}$ with its length N being a power of 2 and its discrete Fourier transform

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-jk\frac{2\pi}{N}n}.$$

The first stage of the algorithm [23, p.272] is based upon its breaking into the sum of even-indexed and odd-indexed data $\{x(n)\}$ to define the following expression

$$X(k) = \sum_{n=0}^{N/2-1} x(2n) e^{-jk\frac{2\pi}{N}2n} + \sum_{n=0}^{N/2-1} x(2n+1) e^{-jk\frac{2\pi}{N}(2n+1)} \tag{3.35}$$

which results in

$$X(k) = \sum_{n=0}^{N/2-1} x(2n) e^{-jk\frac{2\pi}{N/2}n} + e^{-jk\frac{2\pi}{N}} \sum_{n=0}^{N/2-1} x(2n+1) e^{-jk\frac{2\pi}{N/2}n} \tag{3.36}$$

It can be seen that computation of the DFT of length N has been reduced to the computation of two transforms of length $N/2$ and an additional $N/2$ complex multiplications for the complex exponential outside the second summation considering $k = 0, 1, \cdots, N/2 - 1$. It would appear at first sight that it is necessary to evaluate Eq. (3.36) for $k = 0, 1, \cdots, N-1$. However it is not the truth as may be seen by considering result for indices $k+N/2$ having the following form

$$X(k + \frac{N}{2}) = \sum_{n=0}^{N/2-1} x(2n) e^{-j(k+\frac{N}{2})\frac{2\pi}{N/2}n} + e^{-j(k+\frac{N}{2})\frac{2\pi}{N}} \sum_{n=0}^{N/2-1} x(2n+1) e^{-j(k+\frac{N}{2})\frac{2\pi}{N/2}n}$$

which owing to the periodicity results in

$$X(k + \frac{N}{2}) = \sum_{n=0}^{N/2-1} x(2n) e^{-jk\frac{2\pi}{N/2}n} - e^{-jk\frac{2\pi}{N}} \sum_{n=0}^{N/2-1} x(2n+1) e^{-jk\frac{2\pi}{N/2}n} \tag{3.37}$$

Comparing Eqs. (3.36) and (3.37) it is obvious that the only difference is in the sign between the two summations. Thus it is necessary to evaluate Eq. (3.36) for $k = 0, 1, \cdots, N/2–1$ only storing the result of the two summations separately for each k. The values of $X(k)$ and $X(k + N/2)$ can then be evaluated as the sum and difference of the two summations as indicated by Eqs. (3.36) and (3.37). Thus the computational load for an N-point DFT has been reduced from N^2 operations to $2(N/2)^2 + N/2$. The flow chart for incorporating this decomposition into the computation of an $N = 8$ point DFT is presented in Fig. 3.9.

The same process can be carried out on each of the $N/2$ points of the transform to reduce further the computations. The flow chart for incorporating this extra stage of decomposition into the computation of the $N = 8$ point DFT is shown in Fig. 3.10. It can be seen that if $N = 2^M$ then the process can be repeated M times to reduce the

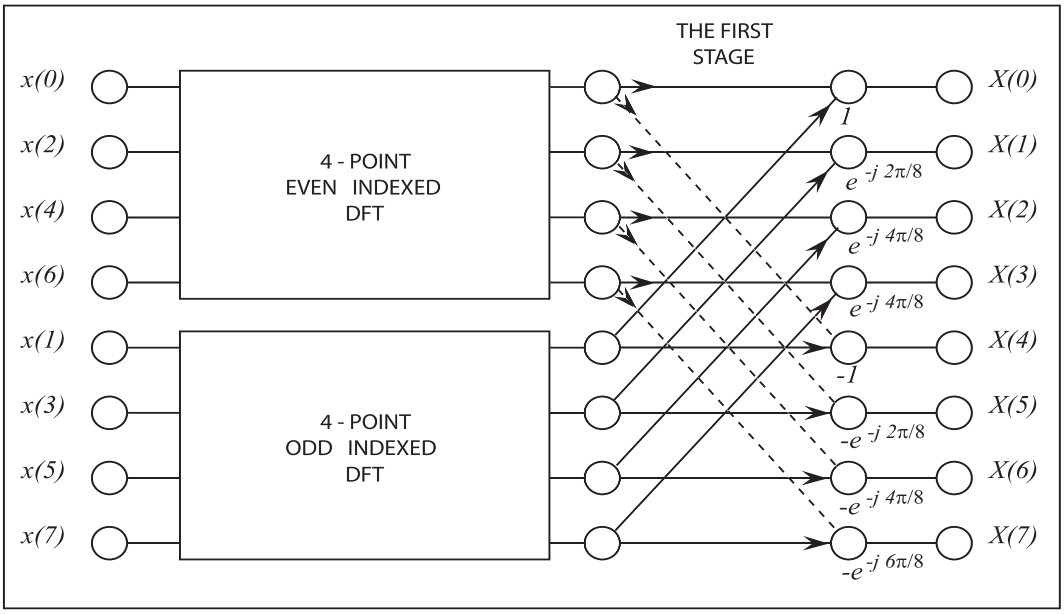

FIGURE 3.9. The first stage of the fast Fourier transform decomposition for $N = 8$

computation to that of evaluating N single point DFTs. The final flow chart for $N = 8$ presented in Fig. 3.10 is based upon the "butterfly" structure of the $N = 2$ point DFT of a sequence $\{s(0), s(1)\}$ evaluating

$$S(0) = s(0) + s(1)e^{-j0\frac{2\pi}{2}} = s(0) + s(1) \quad (3.38)$$
$$S(1) = s(0) + s(1)e^{-j1\frac{2\pi}{2}} = s(0) - s(1) \quad (3.39)$$

It is obvious that for the algorithm presented above it is necessary to shuffle the order of the input data. This data shuffle is usually termed "bit reversal" for reasoning that are clear if the indices of the shuffle data are written in binary as shown in Tab. 3.2.

It can be seen that the process reduces at each stage the computation by half but introduces an extra $N/2$ multiplications to account for the complex exponential term outside the second summation term in the reduction. Thus for the condition of $N = 2^M$ the process can be repeated M times to reduce the computation to that of evaluating N single point DFTs which require no computation. However at each of the M stages of reduction an extra $N/2$ multiplications is introduced so that the total number of arithmetic operations require to evaluate an N-point DFT is $(N/2)log_2(N)$.

binary	000	001	010	011	100	101	110	111
bit reversal	000	100	010	110	001	101	011	111
decimal	0	4	2	6	1	5	3	7

TABLE 3.2. Bit reversal used in the algorithm of the fast Fourier transform.

The FFT algorithm has a further significant advantage over the direct evaluation of the DFT expression in the fact that computation can be performed *on-place*. This has been illustrated in the final flow chart where it can be seen that after two data values have been processed by the *butterfly structure* those data are not required again in the computation and they may be replaced in the computer store with the values at the output of the butterfly structure. Computational algorithm of the fast Fourier transform is used in Algorithm 3.5 presented before.

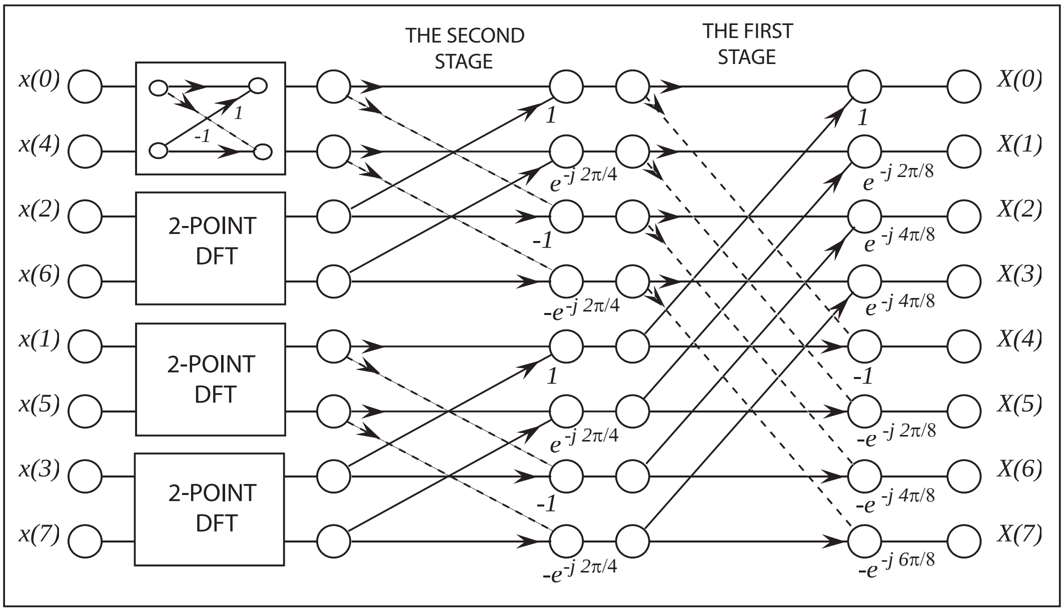

FIGURE 3.10. The first and second stage of the fast Fourier transform decomposition for $N = 8$

It is obvious from the definition of the direct and inverse discrete Fourier transform that the fast algorithm applied to obtain transformed values can be used with slight modifications in both directions.

3.3.3 Signal Decomposition and Reconstruction

Problem of the sampling rate estimation can be simply studied in connection with one harmonic component of the continuous-time periodic signal in the form

$$x_a(t) = cos(\Omega_a t) \tag{3.40}$$

sampled with the *sampling period* T_s to define sequence

$$x(n) = cos(\Omega_a n T_s) \tag{3.41}$$

for $n = 0, 1, \cdots$. Instead of the *analogue* frequency Ω_a [rad/s] we can introduce normalized *digital* frequency $\omega_d = \Omega_a T_s$ [rad] implying

$$x(n) = cos(\omega_d n) \tag{3.42}$$

Let us restrict our attention now to the finite length sequence having N samples and let us apply direct and inverse Fourier transform for its decomposition and reconstruction.

The SIGNAL DECOMPOSITION involves the application of the DFT definition in the form

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-jk\frac{2\pi}{N}n} \tag{3.43}$$

for the unitless frequency index $k = 0, 1, \cdots, N-1$ which can be related according to the previous notes to

- digital frequency in [rad] : $\omega_k = k2\pi/N \in \langle 0, 2\pi)$
- digital frequency in [Hz] : $f_k = \omega_k/(2\pi) = k/N \in \langle 0, 1)$
- analogue frequency in [rad/s] : $\Omega_k = \omega_k/T_s \in \langle 0, 2\pi/T_s)$

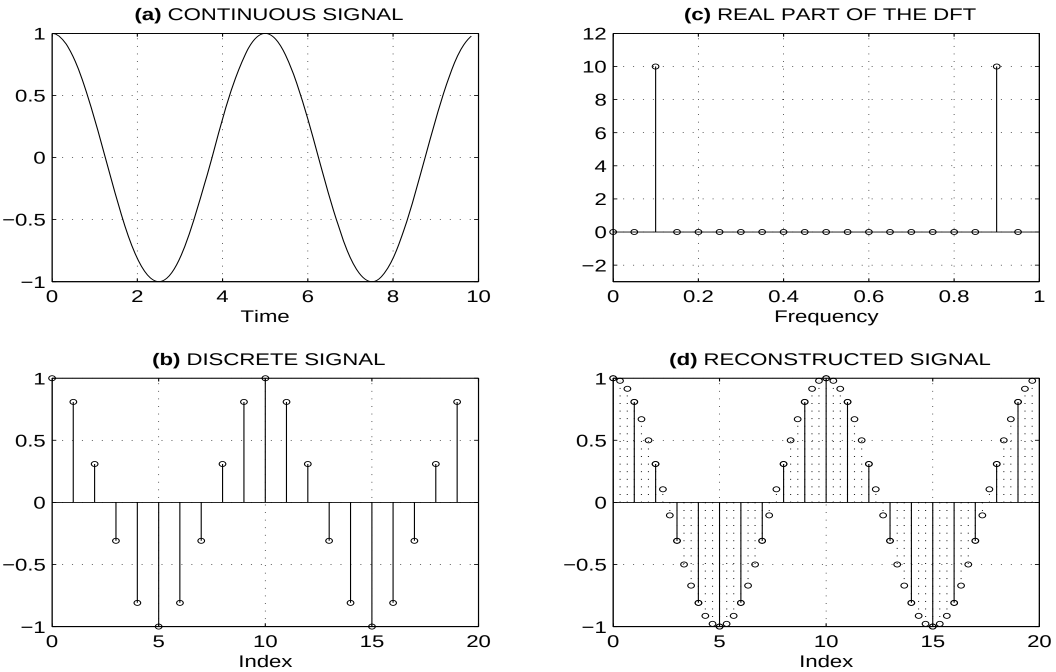

FIGURE 3.11. Signal decomposition and reconstruction: (**a**) Continuous signal $x_a(t) = cos(\Omega_a t)$ for $\Omega_a = 2\ \pi\ f_a \pi$ [rad/s] for $f_a = 0.2[Hz]$ and $t \in < 0, 10)$ [s] (**b**) Discrete signal $x(n) = x_a(n\ T_s)$, $n = 0, 1, \cdots, N-1$ for sampling period $T_s = 0.5$ [s] ($f_s = 1/T_s = 2[Hz]$, $N = 20$ and resulting normalized digital frequency $f_d = f_a/f_s = 0.1$ (**c**) Real part of $X(k)$ defined as a DFT of $\{x(n)\}$ and presented for $k = 0, 1, \cdots, N-1$ (**d**) Result of the inverse DFT of $X(k)$ for signal reconstruction combined with digital interpolation

Using further for simplicity real even sequence $\{x(n)\}_{n=0}^{N-1}$ with the number of its values equal to the multiple of the signal period the evaluation of the DFT results in the real even sequence $\{X(k)\}_{k=0}^{N-1}$ [30, p.93]. The whole process of sampling and analysis for such a harmonic sequence with its digital frequency $\omega_d = 0.2\pi$ ($f_d = 0.1$) and $N = 20$ samples is given in Fig. 3.11 (a), (b) and (c). The result of the DFT is presented for $\omega_k = k2\pi/N, k = 0, 1, \cdots, N/2$ only taking into account that evaluations for indices greater than $k = N/2$ related to frequencies greater then $\omega_k = \pi$ are redundant owing to the periodicity properties of the DFT.

Signal reconstruction is based upon the inverse discrete Fourier transform in the form

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{-jk\frac{2\pi}{N}n} \tag{3.44}$$

for $n = 0, 1, \cdots, N-1$. Using the previous example it is possible to apply this equation to the sequence in Fig. 3.11 (d) given by solid lines. To obtain more values of the reconstructed sequence it is possible to use *digital interpolation* for evaluation of values between these samples given in Fig. 3.11 (d) by dotted lines. The principle of this interpolation [40, p.80] is based on the following statements with their graphic interpretation restricted in Fig. 3.12 for an even sequence with real part of the DFT only

- Real sequence $\{x(n)\}_{n=0}^{N-1}$ derived from a band limited continuous signal $x_a(t)$ with sampling T_s has its DFT $X(k)$ decreasing to zero for $k \to N/2$ and owing to properties of the discrete Fourier transform $X(N-k) = conj(X(k))$ for $k = 0, 1, \cdots, N-1$.

- Real sequence $\{v(m)\}_{m=0}^{M-1}$ where $M = N \cdot N_s$ derived from the same band limited continuous signal $x_a(t)$ with sampling T_s/N_s has its DFT $V(k)$ closely related to the original one with $N \cdot (N_s - 1)$ inserted zero values as no new frequency components are present

$$\{V(k)\}_{k=0}^{M-1} = \frac{M}{N}\{X(0), X(1), \cdots, X(N/2-1), 0, 0, \cdots, 0, X(N/2), \cdots, X(N-1)\} \tag{3.45}$$

Constant M/N introduced in Eq. (3.45) is caused by the different length of sequences $\{x(n)\}$ and $\{v(n)\}$ which affects the multiplication factor in the definition of the inverse DFT. Fig. 3.12 (b) and (e) explain that the analogue resolution frequencies are the same for the DFT of both sequences $\{x(n)\}$ and $\{v(n)\}$. Computer processing of the digital interpolation (for even N) can be based upon the Algorithm 3.6 with all indices shifted by one to have their positive values only. Similar process can be designed for odd N.

We have supposed till now the digital frequency ω_d slow enough enabling signal decomposition and reconstruction as well. It is obvious from Fig. 3.11 that when frequency ω_d is

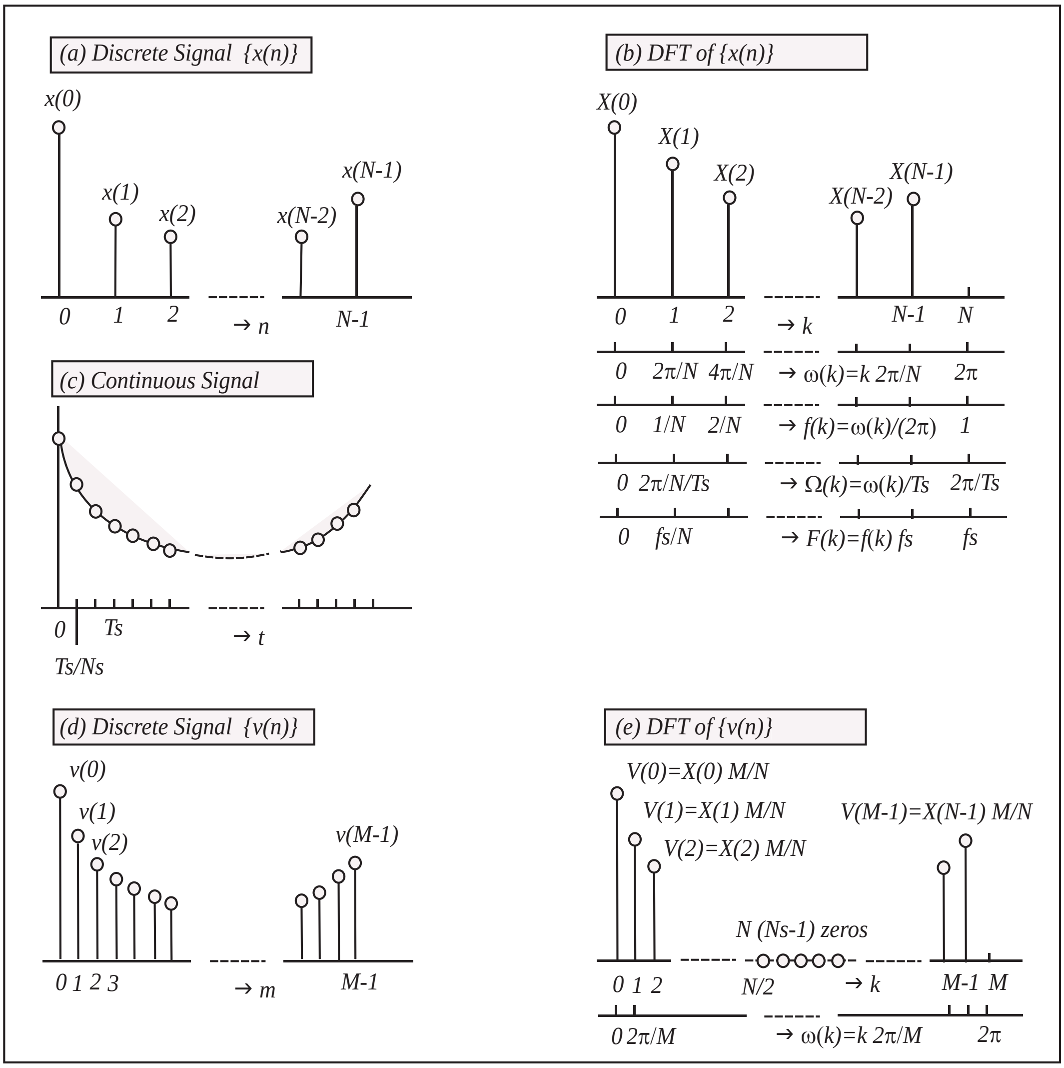

FIGURE 3.12. Principle of the digital interpolation of signal $\{x(n)\}_{n=0}^{N-1}$ =IDFT$[\{X(k)\}_{k=0}^{N-1}]$ by the inverse discrete Fourier transform of $\{V(k)\}_{k=0}^{M-1}$ for $M = N \cdot N_s$ with N_s standing for the number of subsampling intervals.

Algorithm 3.6 *Digital interpolation for signal reconstruction using the inverse DFT*

- definition of vector $\mathbf{x} = [x(1), x(2), \cdots, x(N)]$ and the subsampling index NS
- discrete Fourier transformm evaluation
  $\mathbf{X} = \text{fft}\ (\mathbf{x})$
  defining sequence $\mathbf{X} = [X(1), X(2), \cdots, X(N/2), X(N/2+1) \cdots, X(N)]$
- new sequence definition of the length $M = N \cdot N_s$ with inserted zero values
  $\mathbf{V} = \frac{M}{N}\ [\mathbf{X}(1:N/2), \text{zeros}\ (1, N * (NS-1), \mathbf{X}(N/2+1:N)]$
- inverse discrete Fourier transform
  $\mathbf{y} = \text{ifft}\ (\mathbf{V})$

changing from zero to π the DFT is able to distinguish this frequency component (evaluating its complex conjugate in the range $\langle \pi, 2\pi)$ as well). But when ω_d is grater than π the interpretation is not unique already. This situation is given in Fig. 3.13 for $\omega_d = 1.8\pi$ [rad]. Values of this discrete signal are the same as those in Fig. 3.11 for $\omega_d = 0.2\pi$ [rad] and the reconstruction process provides signal given in Fig. 3.13 with its digital frequency in range $\langle 0, \pi)$ different from the original signal in this case. This frequency reduction is often presented as *aliasing* and it results in the signal reconstruction of the lowest possible frequency component defined by the given sequence as given in Fig. 3.14

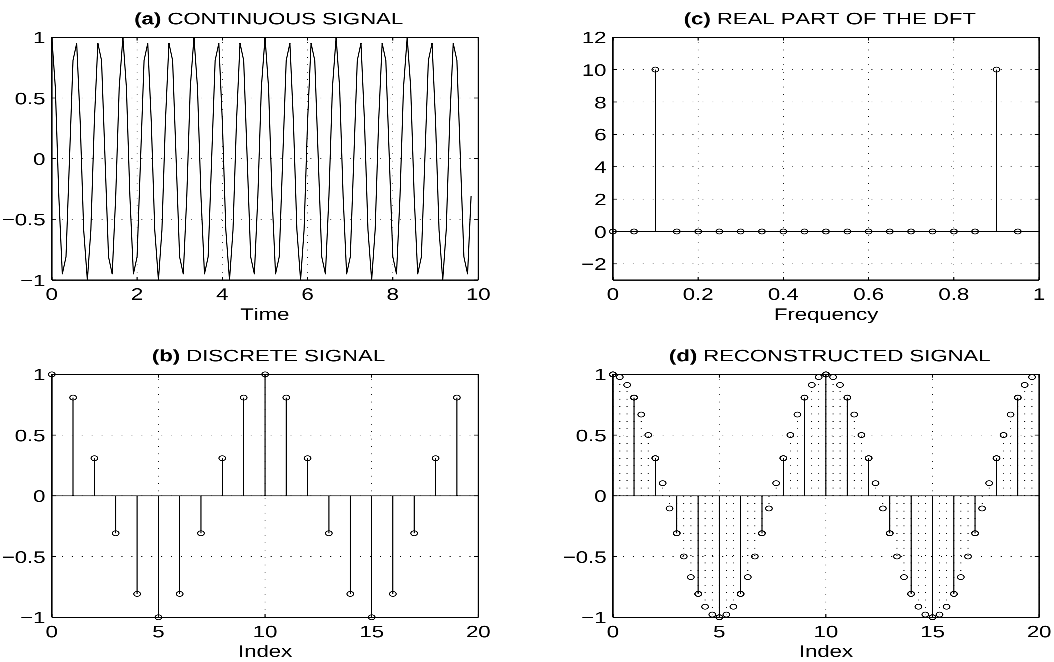

FIGURE 3.13. Signal decomposition and reconstruction: (**a**) Continuous signal $x_a(t) = cos(\Omega_a t)$ for $\Omega_a = 2\ \pi\ f_a \pi$ [rad/s] for $f_a = 1.8[Hz]$ and $t \in < 0, 10)$ [s] (**b**) Discrete signal $x(n) = x_a(n\ T_s)$, $n = 0, 1, \cdots, N-1$ for sampling period $T_s = 0.5$ [s] ($f_s = 1/T_s = 2[Hz]$, $N = 20$ and resulting normalized digital frequency $f_d = f_a/f_s = 0.9$ (**c**) Real part of $X(k)$ defined as a DFT of $\{x(n)\}$ and presented for $k = 0, 1, \cdots, N-1$ (**d**) Result of the inverse DFT of $X(k)$ for signal reconstruction combined with digital interpolation

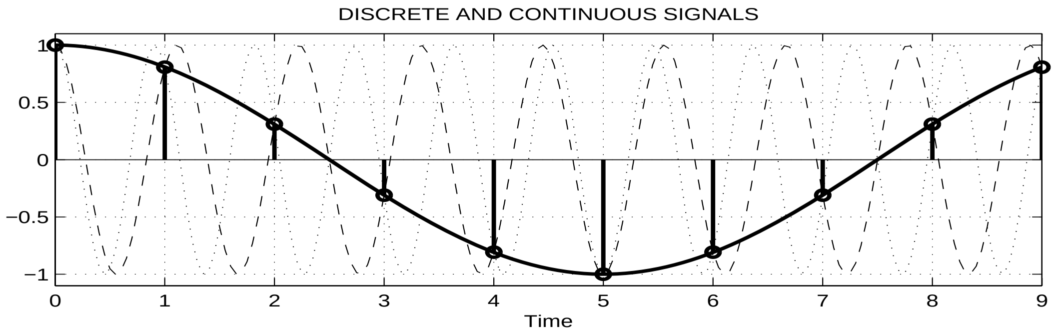

FIGURE 3.14. Continuous signal $x_a(t) = cos(2\pi\, f_a\, t)$ for $f_a = 0.1, 0.9$ and 1.1 [Hz] resulting in the same discrete signal $x(n) = cos(2\pi\, f_d\, n)$ for sampling period $T_s = 1$ and $f_d = f_a$ causing aliasing.

In general any continuous signal of frequency $\Omega_a > \pi/T_s$ is aliased to the frequency range $\langle 0, \pi/T_s)$. To avoid such an aliasing it is necessary to choose the sampling period $T_s \langle \pi/\Omega_a$ or the sampling frequency $f_s > 2f_a$ confirming the sampling theorem presented in the previous chapter. The highest frequency component of a signal implies in this case the necessary sampling rate for further digital signal processing. To reduce the number of the discrete signal values it is sometimes possible to reduce the high frequency components in the analogue signal already and to sample such a prefiltered signal.

3.4 Method of the Least Squares and the Gradient Method

The previous mathematical background was devoted to various methods of signal and system description based on discrete transforms. Further mathematical methods enabling signal and system modelling are based upon the parameters estimation by the least square method. This principle is essential in many engineering applications including signal approximation, prediction and adaptive filtering as well. Its specific modifications will be discussed in further chapters and we shall summarize here basic principles only resulting in finite and iterative methods [43], [40], [12] using nonorthogonal and orthogonal basis functions during the search process and parameters evaluation [25], [20].

3.4.1 General Principle of the Least Square Method

Basic principle of the least square method can be explained on approximation of given values $\{x(n), y(n)\}_{n=0}^{N-1}$ by a linear combination of basis functions $\{g_j(x)\}_{j=0}^{M-1}$ in the form

$$f(x) = \sum_{j=0}^{M-1} w_j g_j(x) \tag{3.46}$$

Main problems of the approximation can be summarized in the following items

- estimation of the general form of function (3.46)
- evaluation of coefficients $w_0, w_1, \cdots, w_{M-1}$ by a chosen method

The first problem can be often solved taking into account physical principle of approximated values and the second one presumes the choice of a proper criterium.

Function $f(x)$ is often looked upon as a continuous function of a variable x but in digital signal processing applications its discrete values are used only defined on a sequence

$\{x(n)\}_{n=0}^{N-1}$. This special case of the *approximation problem* is often referred to as *signal modelling*. In case that we further assume $g_j(x(n)) = x(n-j)$ it is possible to rewrite expression 3.46 to the form

$$f(n) = \sum_{j=0}^{M-1} w_j x(n-j) \tag{3.47}$$

with $f(n)$ standing for $f(x(n))$ in fact. This specific case corresponds to the system representation by the impulse response mentioned before which implies that classical methods of the least square approximation can be applied in both cases. Comparison of a general and specific function defined by Eq. (3.46) and (3.47) for a given set of values $\{x(n)\}$ is presented in Fig. 3.15.

The *method of the least squares* is used for the minimization of the total squared error between given and approximated values visualized in Fig. 3.16 for a chosen example and having generally form

$$J(w_0, w_1, \cdots, w_{M-1}) = \sum_{n=0}^{N-1} \epsilon(n)^2 = \sum_{n=0}^{N-1} (y(n) - f(x(n))^2 = \sum_{n=0}^{N-1} (y(n) - \sum_{j=0}^{M-1} w_j g_j(x(n)))^2 \tag{3.48}$$

As J is a function of variables $w_0, w_1, \cdots, w_{M-1}$ it is possible to find their values minimizing the total sum (3.48) standing for the error-performance surface in Fig. 3.16 and defining coefficients of function (3.46) in this way.

Theorem 3.1 *Let us assume a sequence* $\{x(n), y(n)\}_{n=0}^{N-1}$. *Then the coefficients* $\{w_j\}_{j=0}^{M-1}$ *of the approximation function (3.46) for a given basis* $\{g_j(x)\}_{j=0}^{M-1}$ *are given by the solution of the set of* M *linear algebraic equations*

$$\mathbf{Rw} = \mathbf{p} \tag{3.49}$$

where

$$\mathbf{R} = \begin{bmatrix} \sum g_0(x(n))g_0(x(n)) & \cdots & \sum g_0(x(n))g_{M-1}(x(n)) \\ & \cdots\cdots\cdots & \\ \sum g_{M-1}(x(n))g_0(x(n)) & \cdots & \sum g_{M-1}(x(n))g_{M-1}(x(n)) \end{bmatrix},$$

$$\mathbf{w} = \begin{bmatrix} w_0 \\ \cdots \\ w_{M-1} \end{bmatrix}, \qquad \mathbf{p} = \begin{bmatrix} \sum y(n)g_0(x(n)) \\ \cdots \\ \sum y(n)g_{M-1}(x(n)) \end{bmatrix}$$

with all summations for $n = 0, 1, \cdots, N-1$.

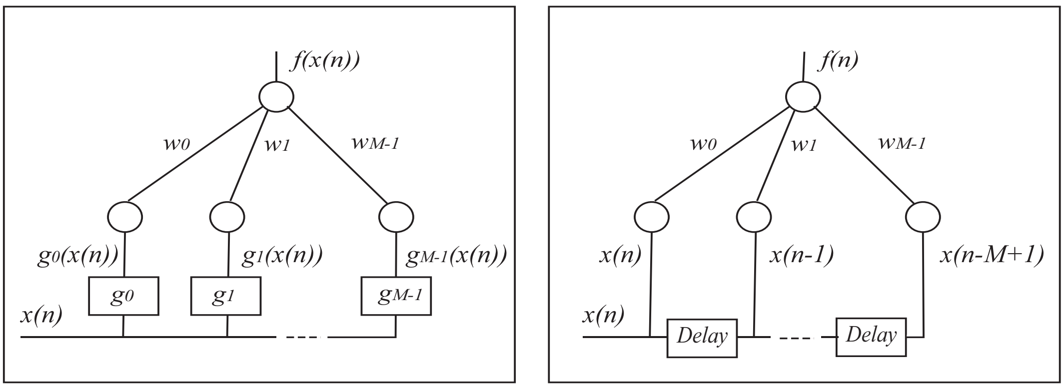

FIGURE 3.15. Comparison between approximation function from the general and signal processing point of view

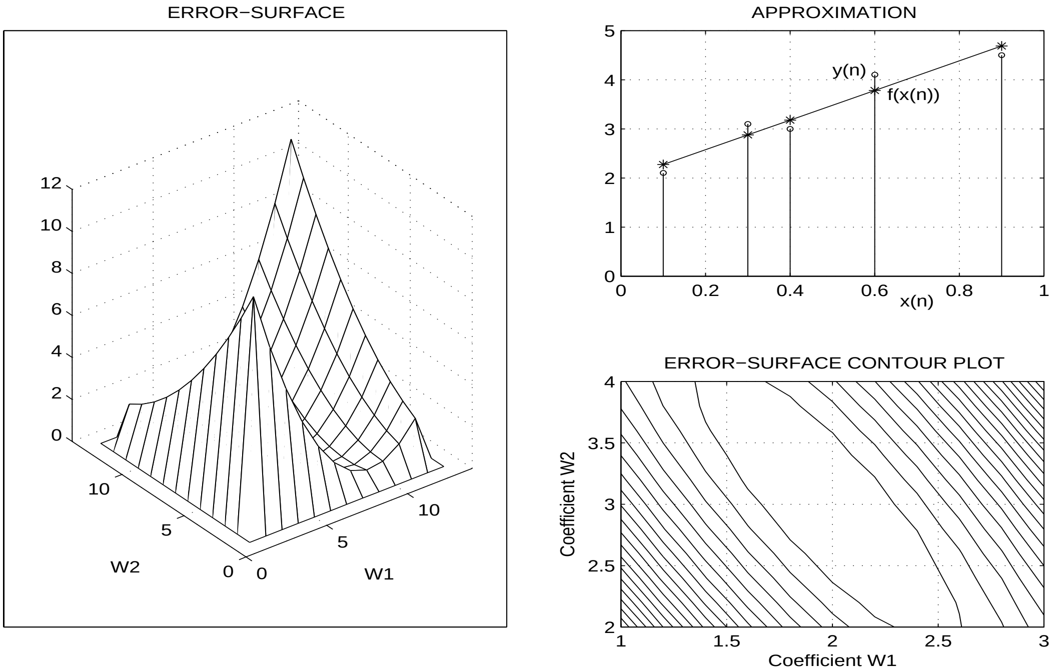

FIGURE 3.16. Error-performance surface for the linear approximation of a given sequence of $N = 5$ values by function $f(x) = w_0 + w_1 x$ by the least squares method and plot of given and evaluated values

Proof: To minimize the sum of squares in form (3.48) it is necessary to evaluate its partial derivatives with respect to coefficients $\{w_j\}_{j=0}^{M-1}$ and to put them equal to zero which means that

$$\frac{\partial J}{\partial w_i} \equiv 2\sum_{n=0}^{N-1}(y(n) - \sum_{j=0}^{M-1} w_j g_j(x(n)) g_i(x(n))) = 0$$

for $i = 0, 1, \cdots, M-1$. Rearranging this equation we shall obtain

$$\sum_{j=0}^{M-1} w_j \sum_{n=0}^{N-1} g_j(x(n)) g_i(x(n)) = \sum_{n=0}^{N-1} y(n) g_i(x(n))$$

The last expression is equivalent to (3.49) and it represents the set of M linear algebraic equations defining coefficients $w_0, w_1, \cdots, w_{M-1}$.

Example 3.9 *Evaluate coefficients of the approximation function in the form*

$$f(x) = w_0 + w_1 x$$

for a given sequence $\{x(n), y(n)\}_{n=0}^{N-1}$.

Solution: Using Eq. (3.48) it is possible to express the sum of squares in the form

$$J(w_0, w_1) = \sum_{n=0}^{N-1}(y(n) - w_0 - w_1 x(n))^2$$

The condition for minimizing this expression can be stated in the form

$$\frac{\partial J}{\partial w_0} \equiv -2\sum_{n=0}^{N-1}(y(n) - w_0 - w_1 x(n)) = 0$$

$$\frac{\partial J}{\partial w_1} \equiv -2\sum_{n=0}^{N-1}(y(n) - w_0 - w_1 x(n)) x(n) = 0$$

which results in the set of equation

$$\begin{aligned} w_0 N + w_1 \sum x(n) &= \sum y(n) \\ w_0 \sum x(n) + w_1 \sum x(n)^2 &= \sum x(n)y(n) \end{aligned}$$

with all summations for $n = 0, 1, \cdots, N-1$ defining coefficients w_0 and w_1. Graphic results of this example for a given sequence of values is presented in Fig. 3.16.

The set (3.49) of linear algebraic equations is not well conditioned which might cause numerical problems in its solution. It is one of reasons why orthogonal basis functions are used as well.

Definition 3.3 *The sequence of functions* $\{p_j(x)\}_{j=0}^{M-1}$ *is said to be orthogonal with respect to a given sequence* $\{x(n)\}_{n=0}^{N-1}$ *in case that*

$$\sum_{n=0}^{N-1} p_i(x(n))p_j(x(n)) \begin{cases} = 0 & \textit{for } i \neq j \\ \neq 0 & \textit{for } i = j \end{cases} \tag{3.50}$$

The sum (3.50) represents scalar multiplication in fact referred to as $(p_i(x), p_j(x))$ very often which substantially simplifies the approximation problem stated in the next theorem.

Theorem 3.2 *Let as assume a sequence* $\{x(n), y(n)\}_{n=0}^{N-1}$. *Then the coefficients* $\{w_j\}_{j=0}^{M-1}$ *of the approximation function*

$$f(x) = \sum_{j=0}^{M-1} w_j p_j(x) \tag{3.51}$$

for given orthogonal basis functions $\{p_j(x\}_{j=0}^{M-1}$ *with respect to the sequence* $\{x(n)\}_{n=0}^{N-1}$ *are defined by relation*

$$w_j = \frac{\sum_{n=0}^{N-1} y(n)p_j(x(n))}{\sum_{n=0}^{N-1} p_j(x(n))p_j(x(n))} \tag{3.52}$$

for $j = 0, 1, \cdots, M-1$.

Proof of this statement is based upon that of Theorem 3.1 with the matrix G reduced to the diagonal matrix with zero nondiagonal elements owing to the definition of orthogonal functions. As no set of equations is solved in this case it is possible very simply to evaluate any further coefficient w_j to improve the approximation with no change of coefficients defined before.

A typical example of the error-performance surface in this case is presented on Fig 3.17 for the linear approximation. The comparison of this sketch with that in Fig. 3.16 for the nonorthogonal bases functions illustrates that orthogonality changes positions of axis only with no affect to a very flat minimum of the surface.

Definition of the set of orthogonal basis functions $\{p_j(x)\}_{j=1}^{M-1}$ can be based upon the *Gramm - Schmidt* process [25] applied to the nonorthogonal set of functions $\{g_j(x)\}_{j=0}^{M-1}$. The whole process can be summarized in the following way using the notation for scalar multiplication mentioned before

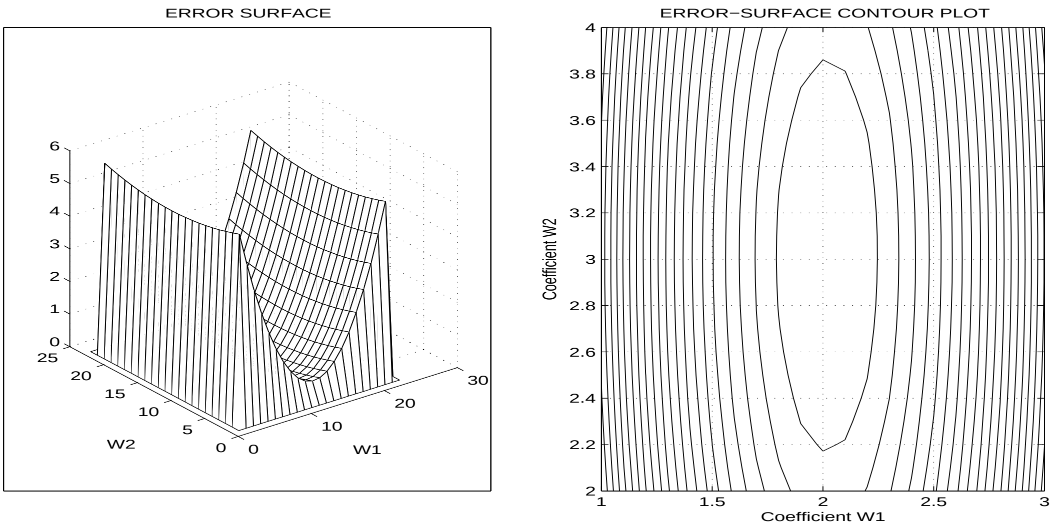

FIGURE 3.17. Error-performance surface for linear approximation of a given sequence of $N = 5$ values $\{x(n), y(n)\}$ by function $f(x(n)) = w_0 + w_1(x(n) - mean(\mathbf{x}))$ involving the set of orthogonal basis functions $\{1, \mathbf{x} - mean(\mathbf{x})\}$ for given values $\{x(n)\}$

- definition of
$$p_0(x) = g_0(x)$$
- estimation of
$$p_1(x) = g_1(x) - \Lambda_{01} p_0(x)$$
orthogonal to $p_1(x)$ implying that
$$(p_1(x), p_0(x)) \equiv (g_1(x), p_0(x)) - \Lambda_{01}(p_0(x), p_0(x)) = 0$$
$$\Lambda_{01} = \frac{(g_1(x), p_0(x))}{(p_0(x), p_0(x))}$$
- estimation of
$$p_2(x) = g_2(x) - \Lambda_{02} p_0(x) - \Lambda_{12} p_1(x)$$
orthogonal to $p_1(x)$ and $p_2(x)$ implying that
$$(p_2(x), p_0(x)) \equiv (g_2(x), p_0(x)) - \Lambda_{02}(p_0(x), p_0(x)) - \Lambda_{12}(p_1(x), p_0(x)) = 0$$
$$\Lambda_{02} = \frac{(}{g_2(x), p_0(x))}(p_0(x), p_0(x))$$
and
$$(p_2(x), p_1(x)) \equiv (g_2(x), p_1(x)) - \Lambda_{02}(p_0(x), p_1(x)) - \Lambda_{12}(p_1(x), p_1(x)) = 0$$
$$\Lambda_{12} = \frac{(}{g_2(x), p_1(x)})(p_1(x), p_1(x))$$

The same process can be applied for further functions in the same way as well.

Example 3.10 *Evaluate coefficients of the approximation function in the form*
$$f(x) = w_0 p_0(x) + w_1 p_1(x)$$
for a given sequence $\{x(n), y(n)\}_{n=0}^{N-1}$ *and orthogonal bases functions* $\{p_0(x), p_1(x)\}$ *defined by nonorthogonal functions* $g_0(x) = 1$ *and* $g_1(x) = x$.

Solution: Using the Gramm-Schmidt process described before it is possible to define
$$\begin{aligned} p_0(x) &= g_0(x) = 1 \\ p_1(x) &= g_1(x) - \Lambda_{01} p_0(x) \end{aligned}$$

where

$$\Lambda_{01} = \frac{(g_1(x), p_0(x))}{(p_0(x), p_0(x))} = \frac{\sum_{n=0}^{N-1} x(n)}{N}$$

The approximation function has therefore the following form

$$f(x) = w_0 + w_1(x - \Lambda_{01})$$

where according to Eq. (3.52)

$$\begin{aligned} w_0 &= \frac{1}{N}\sum_{n=0}^{N-1} y(n) \\ w_1 &= \frac{\sum_{n=0}^{N-1} y(n)(x(n) - \Lambda_{01})}{\sum_{n=0}^{N-1}(x(n) - \Lambda_{01})^2} \end{aligned}$$

Graphic results of this example for a given sequence of values is presented in Fig. 3.16.

Examples 3.9 and 3.10 explain that the same approximation function can be evaluated in two possible ways. Nonorthogonal basis functions results in the set of algebraic equations while orthogonal basis functions enable direct evaluation of coefficients after the previous orthogonalization process.

3.4.2 The Steepest Descent Method

In case of the linear approximation the basic method described in the previous section can be used. In more general case and nonlinear approximation function other methods must be applied. We shall describe now very briefly the *gradient method* used very often in many applications involving optimization problems of various objective functions.

The total squared error given by Eq. (3.48) presented before is a function of M variables $\{w_0, w_1, \cdots, w_{M-1}\}$ in the form

$$J(w_0, w_1, \cdots, w_{M-1}) = \sum_{n=0}^{N-1}(y(n) - \sum_{j=0}^{M-1} w_j g_j(x(n)))^2 \tag{3.53}$$

or in the equivalent matrix notation

$$J(\mathbf{w}) = (\mathbf{y} - \mathbf{G}'\mathbf{w})'(\mathbf{y} - \mathbf{G}'\mathbf{w}) \tag{3.54}$$

where

$$\begin{aligned} \mathbf{w} &= [w_0, w_1, \cdots, w_{M-1}]' \\ \mathbf{y} &= [y(0), y(1), \cdots, y(N-1)]' \\ \mathbf{x} &= [x(0), x(1), \cdots, x(N-1)]' \end{aligned}$$

and

$$\mathbf{G} = \begin{bmatrix} g_0(\mathbf{x}') \\ \cdots \\ g_{M-1}(\mathbf{x}') \end{bmatrix} = \begin{bmatrix} g_0(x(0)) & \cdots & g_0(x(N-1)) \\ & \cdots\cdots\cdots & \\ g_{M-1}(x(0)) & \cdots & g_{M-1}(x(N-1)) \end{bmatrix}$$

with an apostrophe standing for matrix or vector transposition.

To find the optimum vector $\mathbf{w}$ defining the minimum value of function (3.54) assumes the evaluation of the *gradient vector*

$$\frac{\partial J(w_0,\cdots,w_{M-1})}{\partial w_i} = 2\sum_{n=0}^{N-1}(y(n) - \sum_{j=0}^{M-1} w_j g_j(x(n)))g_i(x(n)) \tag{3.55}$$

resulting in the following matrix notation

$$\frac{\partial J(\mathbf{w})}{\partial w_i} = -2\mathbf{G}(i,:)(\mathbf{y} - \mathbf{G}'\mathbf{w}) \tag{3.56}$$

where $\mathbf{G}(i,:)$ stands for the i-th row of matrix $\mathbf{G}$ and $i = 0, 1, \cdots, M-1$ or in the compact form

$$\mathbf{q} = \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = -2\mathbf{G}(\mathbf{y} - \mathbf{G}'\mathbf{w}) = \mathbf{2}(\mathbf{R}\mathbf{w} - \mathbf{p}) \tag{3.57}$$

where

$$\mathbf{R} = \mathbf{G}\mathbf{G}', \qquad \mathbf{p} = \mathbf{G}\mathbf{y} \tag{3.58}$$

The optimum vector $\mathbf{w} = \mathbf{w}^*$ has such values for which its gradient is equal to zero resulting in Eq. (3.49) in the form

$$\mathbf{R}\mathbf{w}^* = \mathbf{p} \tag{3.59}$$

discussed in the previous section already.

Using this notation for the optimum gradient vector it is possible to use it in Eq. (3.54) which provides another expression for the sum of squares based on the weight deviation vector

$$\mathbf{v} = \mathbf{w} - \mathbf{w}^* \tag{3.60}$$

As

$$\begin{aligned} J(\mathbf{w}) &= (\mathbf{y} - \mathbf{G}'\mathbf{w})'(\mathbf{y} - \mathbf{G}'\mathbf{w}) = (\mathbf{y}' - \mathbf{w}'\mathbf{G})(\mathbf{y} - \mathbf{G}'\mathbf{w}) = \\ &= \mathbf{y}'\mathbf{y} - \mathbf{w}'\mathbf{G}\mathbf{y} - \mathbf{y}'\mathbf{G}'\mathbf{w} + \mathbf{w}'\mathbf{G}\mathbf{G}'\mathbf{w} \end{aligned}$$

it is possible to use (3.58) and (3.60) to find

$$\begin{aligned} J(\mathbf{w}) &= \mathbf{y}'\mathbf{y} - (\mathbf{v} + \mathbf{w}^*)'\mathbf{p} - \mathbf{p}'(\mathbf{v} + \mathbf{w}^*) + (\mathbf{v} + \mathbf{w}^*)'\mathbf{R}(\mathbf{v} + \mathbf{w}^*) = \\ &= \mathbf{y}'\mathbf{y} - \mathbf{v}'\mathbf{p} - (\mathbf{w}^*)'\mathbf{p} - \mathbf{p}'\mathbf{v} - \mathbf{p}'\mathbf{w}^* + \mathbf{v}'\mathbf{R}\mathbf{v} + \mathbf{v}'\mathbf{R}\mathbf{w}^* + (\mathbf{w}^*)'\mathbf{R}\mathbf{v} + (\mathbf{w}^*)'\mathbf{R}\mathbf{w}^* \end{aligned}$$

Using (3.59) it follows that $\mathbf{v}'\mathbf{R}\mathbf{w}^* = \mathbf{v}'\mathbf{p}$ and $(\mathbf{w}^*)'\mathbf{R}\mathbf{w}^* = (\mathbf{w}^*)'\mathbf{p}$ and as $\mathbf{R}' = \mathbf{R}$ it is possible to express $(\mathbf{w}^*)'\mathbf{R}\mathbf{v} = \mathbf{p}'\mathbf{v}$ which results in the following relation

$$J(\mathbf{w}) = \mathbf{y}'\mathbf{y} - (\mathbf{w}^*)'\mathbf{R}\mathbf{w}^* + \mathbf{v}'\mathbf{R}\mathbf{v} \tag{3.61}$$

or

$$J(\mathbf{w}) = J_{min}(\mathbf{w}) + \mathbf{v}'\mathbf{R}\mathbf{v} \tag{3.62}$$

The last expression is used very often to evaluate the error-performance surface around its minimum value and to find gradients for further processing as well.

We can visualize the dependence of the squared error on elements of vector $\mathbf{w}$ by a sketch in Fig. 3.18 for $M = 2$ elements only and to design an alternative procedure to the finite least square method referred to as the method of the *steepest descent* summarised in Algorithm 3.7 for the approximation problem.

Example 3.11 *Evaluate the coefficients of the approximation function in the form*

$$f(x) = w_0 + w_1 x \tag{3.65}$$

for a given sequence $\{x(n), y(n)\}_{n=0}^{N-1}$.

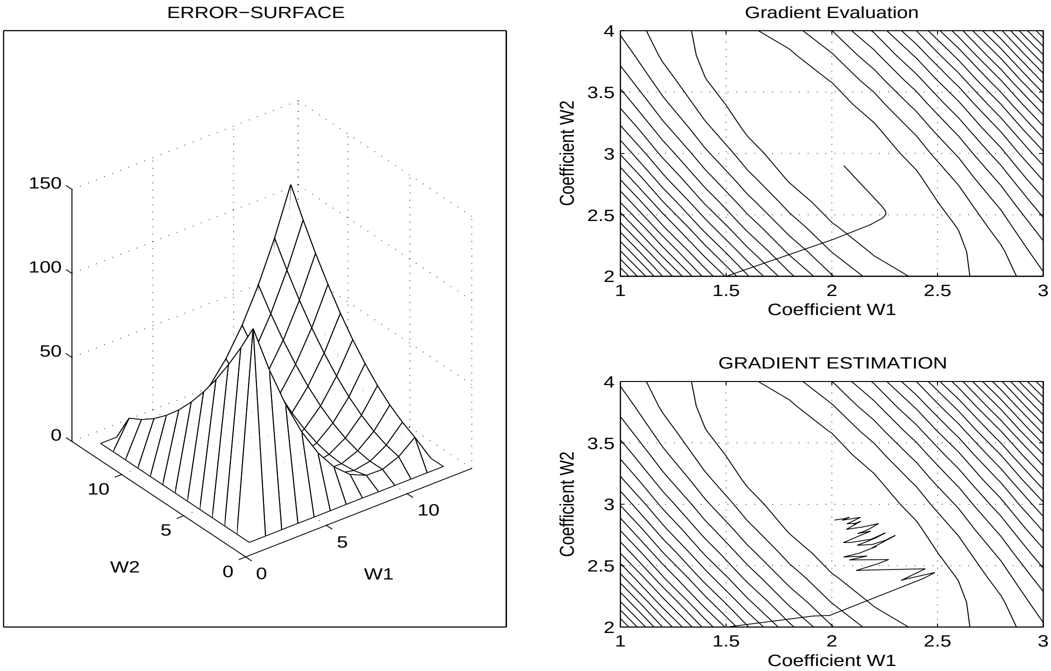

FIGURE 3.18. Error-performance surface for the linear approximation of a given sequence $\mathbf{y} = 2 + 3 * \mathbf{x}$ with the additive random noise and random values of vector $\mathbf{x} = [x(0), x(1), \cdots, x(N-1)]'$ for $N = 50$ by function $f(x) = w_0 + w_1 x$ and results of the steepest descent search with gradient evaluated both for the whole set of given values and estimated separately for each of approximated values with the initial estimate $\mathbf{w} = [1.5, 2]$

.

Algorithm 3.7 *The steepest descent method applied for approximation of N values $\mathbf{y} = y(\mathbf{x})$ for $\mathbf{x} = [x(0), \cdots, x(N-1)]'$ by sequence $\mathbf{f} = \mathbf{w}'\mathbf{G}$ with weights $\mathbf{w} = [w_0, \cdots, w_{M-1}]'$ minimizing the objective function*

$$J(\mathbf{w}) = (\mathbf{y} - \mathbf{G}'\mathbf{w})'(\mathbf{y} - \mathbf{G}'\mathbf{w})$$

for a given set $\{g_0, \cdots, g_{M-1}\}$ of M basis functions defining matrix

$$\mathbf{G} = [g_0(\mathbf{x}'), \cdots, g_{M-1}(\mathbf{x}')]'$$

- definition of vectors $\mathbf{x}$, $\mathbf{y}$ and matrix $\mathbf{G}$ of values of basis functions
- estimation of the initial guess of coefficients $\mathbf{w}$
- iterative evaluation of
  - the gradient vector
  $$\mathbf{q} = -2 * \mathbf{G} * (\mathbf{y} - \mathbf{G}' * \mathbf{w}) \qquad (3.63)$$
  - new estimate of coefficients in direction opposite to that of the gradient vector for a given convergence factor c
  $$\mathbf{w} = \mathbf{w} - c * \mathbf{q} \qquad (3.64)$$

Solution: Matrix $\mathbf{G}$ defined by Eq. (3.54) has values

$$\mathbf{G} = \begin{bmatrix} \mathbf{x}' \cdot \wedge 0 \\ \mathbf{x}' \cdot \wedge 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x(0) & x(1) & \cdots & x(N-1) \end{bmatrix} \tag{3.66}$$

with symbol $\cdot\wedge$ defining that each element of a given vector is raised to the given power and it implies values of gradient vector (3.57) used in the iterative Algorithm **??**. Results for a chosen artificial sequence $\mathbf{y} = 2 + 3 * \mathbf{x}$ with the additive random noise and random values of vector $\mathbf{x} = [x(0), x(1), \cdots, x(N-1)]'$ for $N = 50$ are presented in Fig. 3.18 for gradient evaluated for the whole set of given values and initial estimate of weights $\mathbf{w} = [1.5, 2]$.

The same principle of the gradient search can be applied in various modifications to any objective function and other problems as well. In context of signal processing and adaptive filtering the method of the steepest descent is modified to much simple form but a slower process of convergence, too [43, 12].

Let us consider a single squared error of Eq. (3.53) in the form

$$\varepsilon(n)^2 = (y(n) - \sum_{j=0}^{M-1} w_j g_j(x(n)))^2 \tag{3.67}$$

as an estimate of the mean of squared error used for the gradient evaluation before [43, p.100]. The gradient estimate for each n can be then written in the form

$$\hat{\mathbf{q}}(n) = \begin{bmatrix} \frac{\partial \varepsilon(n)^2}{\partial w_0} \\ \cdots \\ \frac{\partial \varepsilon(n)^2}{\partial w_{M-1}} \end{bmatrix} = 2\varepsilon(n) \begin{bmatrix} \frac{\partial \varepsilon(n)}{\partial w_0} \\ \cdots \\ \frac{\partial \varepsilon(n)}{\partial w_{M-1}} \end{bmatrix} = -2\varepsilon(n) \begin{bmatrix} g_0(x(n)) \\ \cdots \\ g_{M-1}(x(n)) \end{bmatrix} \tag{3.68}$$

It is obvious that

$$\mathbf{q} = \sum_{n=0}^{N-1} \hat{\mathbf{q}}(n) \tag{3.69}$$

enabling to evaluate the gradient from its estimates. The whole process for such a modified gradient method is given in Algorithm 3.8.

The *convergence factor* c has the same meaning as before and it regulates the speed and stability of convergence. As the estimates of the gradient vector are imperfect it is possible to expect noisy adaptive process not following the true line of the steepest descent. Results for the previous example with $M = 2$ elements of vector $\mathbf{w}$ only are given in Fig. 3.18. The choice of orthogonal basis functions can improve the whole process of adaptation.

Similar method of that described before can be applied in case of signal processing applications. Defining the basis functions in $g_j(x(n)) = x(n-j)$ it is possible to find the estimate of values $\{y(n)\}$ in the same way for the objective function in the form

$$J(w_0, w_1, \cdots, w_{M-1}) = \sum_{n=0}^{N-1} \varepsilon(n)^2 = \sum_{n=0}^{N-1} (y(n) - \sum_{j=0}^{N-1} w_j x(n-j))^2 \tag{3.72}$$

As the sequence of values $\{x(n), y(n)\}$ is usually very long the approximate gradient method is used very often. The estimate of the gradient vector given by Eq. (3.72) can be then stated in the following form

$$\hat{\mathbf{q}}(n) = -2\varepsilon(n) \begin{bmatrix} x(n) \\ x(n-1) \\ \cdots \\ x(n-M+1) \end{bmatrix} \tag{3.73}$$

Algorithm 3.8 *The gradient search applied for approximation of N values $\mathbf{y} = y(\mathbf{x})$ for $\mathbf{x} = [x(0), \cdots, x(N-1)]'$ by sequence $\mathbf{f} = \mathbf{w}'\mathbf{G}$ with weights $\mathbf{w} = [w_0, \cdots, w_{M-1}]'$ minimizing the objective function*

$$J(\mathbf{w}) = (\mathbf{y} - \mathbf{G}'\mathbf{w})'(\mathbf{y} - \mathbf{G}'\mathbf{w})$$

for a given set $\{g_0, \cdots, g_{M-1}\}$ of M basis functions defining matrix

$$\mathbf{G} = [g_0(\mathbf{x}'), \cdots, g_{M-1}(\mathbf{x}')]'$$

- definition of vectors $\mathbf{x}$, $\mathbf{y}$ and matrix $\mathbf{G}$ of values of basis functions
- estimation of the initial guess of coefficients $\mathbf{w}$
- iterative evaluation for each n of
  - the estimate of the gradient vector

$$\hat{\mathbf{q}}(n) = -2\mathbf{G}(:, n) * (y(n) - \mathbf{G}(:, n)' * \mathbf{w}) \tag{3.70}$$

  - new estimate of coefficients in direction opposite to that of the gradient vector for a given convergence factor c

$$\mathbf{w} = \mathbf{w} - c * \hat{\mathbf{q}} \tag{3.71}$$

defining the iterative process

$$\mathbf{w}_{new} = \mathbf{w}_{old} - c\hat{\mathbf{q}}(n) \tag{3.74}$$

The sample process of adaptation for $M = 2$ weights is presented in Fig. 3.19.

The mean least square principle and gradient methods are essential in many signal processing applications and they are closely related to the classical Newton method as well. Their more detail discussion will be presented further.

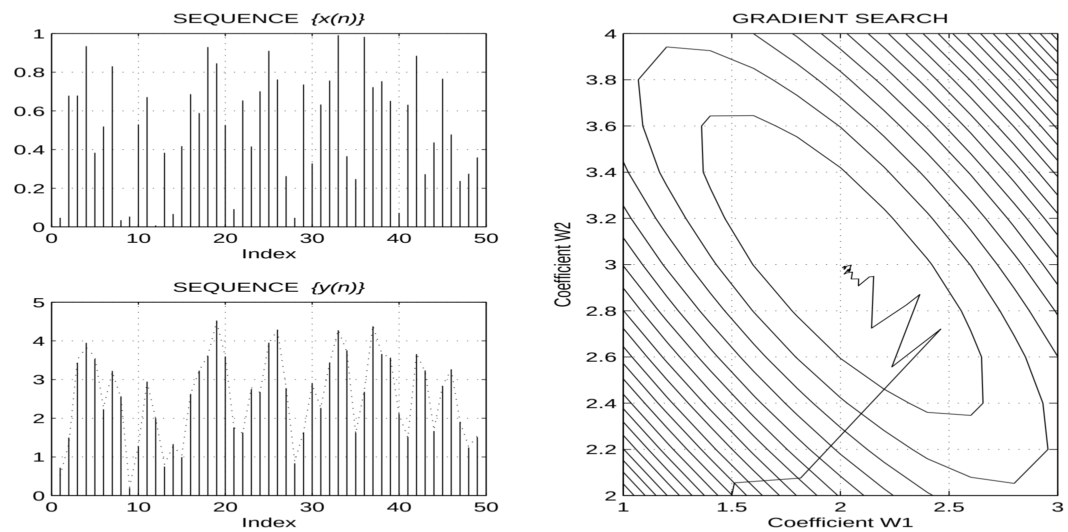

FIGURE 3.19. Signal modelling of a chosen sequence $\mathbf{y}$: $y(n) = 3x(n) + 2x(n-1)$ with the additive random noise for $N = 50$ random values of vector $\mathbf{x}$ by values $\{\mathbf{f} : f(n) = w_0x(n) + w_1x(n-1)\}$ with weights continuously updated using the gradient estimate and initial guess of vector $\mathbf{w} = [1.5, 2]$

3.5 Summary

Z-transform stands for a basic mathematical tool in signal processing methods enabling representation of a signal $\{x(n)\}$ in complex domain by a function $X(z)$ of complex variable z. *Direct transform* is based upon its definition while for the *inverse transform* usually indirect methods are used based upon the partial fraction expansion and polynomial division. These techniques may be simplified by various computer routines including MATLAB functions as well.

Application of Z-transform covers various possibilities of system description including *discrete transfer function* and *frequency response function* using the complex plane representation. Z-transform is often used to simplify the solution of difference equations, too.

Discrete Fourier transform closely related to Z-transform is a basic mathematical tool for signal decomposition and reconstruction. Its applications cover many engineering disciplines as well.

Various learning and adaptive discrete systems are based upon the use of the *least square method* fundamental in many optimization problems. In many cases *gradient methods* are applied.