1

INTRODUCTION TO ADAPTIVE FILTERING

1.1 INTRODUCTION

In this section, we define the kind of signal processing systems that will be treated in this text.

In the last thirty years significant contributions have been made in the signal processing field. The advances in digital circuit design have been the key technological development that sparked a growing interest in the field of digital signal processing. The resulting digital signal processing systems are attractive due to their low cost, reliability, accuracy, small physical sizes, and flexibility.

One example of a digital signal processing system is called *filter*. Filtering is a signal processing operation whose objective is to process a signal in order to manipulate the information contained in the signal. In other words, a filter is a device that maps its input signal to another output signal facilitating the extraction of the desired information contained in the input signal. A digital filter is the one that processes discrete-time signals represented in digital format. For time-invariant filters the internal parameters and the structure of the filter are fixed, and if the filter is linear the output signal is a linear function of the input signal. Once prescribed specifications are given, the design of time-invariant linear filters entails three basic steps, namely: the approximation of the specifications by a rational transfer function, the choice of an appropriate structure defining the algorithm, and the choice of the form of implementation for the algorithm.

An adaptive filter is required when either the fixed specifications are unknown or the specifications cannot be satisfied by time-invariant filters. Strictly speaking an adaptive filter is a nonlinear filter since its characteristics are dependent on the input signal and consequently the homogeneity and additivity conditions are not satisfied. However, if we freeze the filter parameters at a given instant of time, most adaptive filters considered in this text are linear in the sense that their output signals are linear functions of their input signals. The exceptions are the adaptive filters discussed in Chapter 11.

The adaptive filters are time-varying since their parameters are continually changing in order to meet a performance requirement. In this sense, we can interpret an adaptive filter as a filter that performs the approximation step on-line. Usually, the definition of the performance criterion requires the existence of a reference signal that is usually hidden in the approximation step of fixed-filter

P.S.R. Diniz, *Adaptive Filtering*, DOI: 10.1007/978-0-387-68606-6_1,

design. This discussion brings the feeling that in the design of fixed (nonadaptive) filters a complete characterization of the input and reference signals is required in order to design the most appropriate filter that meets a prescribed performance. Unfortunately, this is not the usual situation encountered in practice, where the environment is not well defined. The signals that compose the environment are the input and the reference signals, and in cases where any of them is not well defined, the design procedure is to model the signals and subsequently design the filter. This procedure could be costly and difficult to implement on-line. The solution to this problem is to employ an adaptive filter that performs on-line updating of its parameters through a rather simple algorithm, using only the information available in the environment. In other words, the adaptive filter performs a data-driven approximation step.

The subject of this book is adaptive filtering, which concerns the choice of structures and algorithms for a filter that has its parameters (or coefficients) adapted, in order to improve a prescribed performance criterion. The coefficient updating is performed using the information available at a given time.

The development of digital very large scale integration (VLSI) technology allowed the widespread use of adaptive signal processing techniques in a large number of applications. This is the reason why in this book only discrete-time implementations of adaptive filters are considered. Obviously, we assume that continuous-time signals taken from the real world are properly sampled, i.e., they are represented by discrete-time signals with sampling rate higher than twice their highest frequency. Basically, it is assumed that when generating a discrete-time signal by sampling a continuous-time signal, the Nyquist or sampling theorem is satisfied [1]-[9].

1.2 ADAPTIVE SIGNAL PROCESSING

As previously discussed, the design of digital filters with fixed coefficients requires well defined prescribed specifications. However, there are situations where the specifications are not available, or are time varying. The solution in these cases is to employ a digital filter with adaptive coefficients, known as adaptive filters [10]-[17].

Since no specifications are available, the adaptive algorithm that determines the updating of the filter coefficients, requires extra information that is usually given in the form of a signal. This signal is in general called a desired or reference signal, whose choice is normally a tricky task that depends on the application.

Adaptive filters are considered nonlinear systems, therefore their behavior analysis is more complicated than for fixed filters. On the other hand, because the adaptive filters are self designing filters, from the practitioner's point of view their design can be considered less involved than in the case of digital filters with fixed coefficients.

The general set up of an adaptive-filtering environment is illustrated in Fig. 1.1, where k is the iteration number, $x(k)$ denotes the input signal, $y(k)$ is the adaptive-filter output signal, and $d(k)$ defines the desired signal. The error signal $e(k)$ is calculated as $d(k) - y(k)$. The error signal is then used to form a performance (or objective) function that is required by the adaptation algorithm in order to determine the appropriate updating of the filter coefficients. The minimization of the objective function implies that the adaptive-filter output signal is matching the desired signal in some sense.

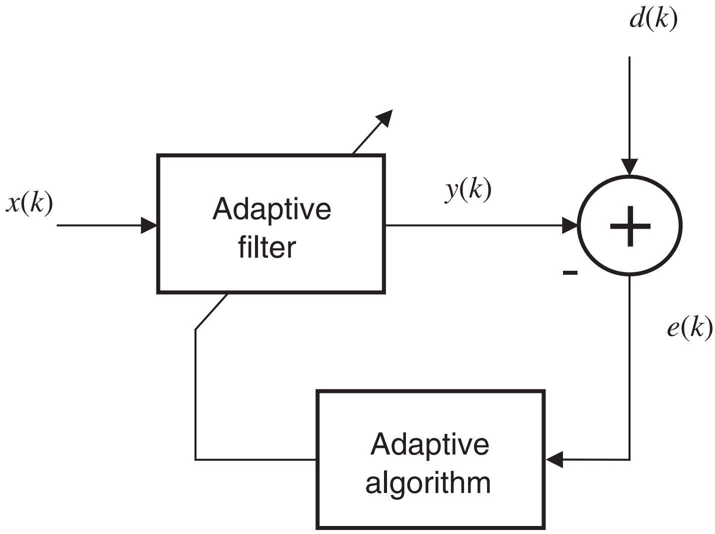

Figure 1.1 General adaptive-filter configuration.

The complete specification of an adaptive system, as shown in Fig. 1.1, consists of three items:

1) **Application**: The type of application is defined by the choice of the signals acquired from the environment to be the input and desired-output signals. The number of different applications in which adaptive techniques are being successfully used has increased enormously during the last two decades. Some examples are echo cancellation, equalization of dispersive channels, system identification, signal enhancement, adaptive beamforming, noise cancelling, and control [14]-[20]. The study of different applications is not the main scope of this book. However, some applications are considered in some detail.

2) **Adaptive-Filter Structure**: The adaptive filter can be implemented in a number of different structures or realizations. The choice of the structure can influence the computational complexity (amount of arithmetic operations per iteration) of the process and also the necessary number of iterations to achieve a desired performance level. Basically, there are two major classes of adaptive digital filter realizations, distinguished by the form of the impulse response, namely the finite-duration impulse response (FIR) filter and the infinite-duration impulse response (IIR) filters. FIR filters are usually implemented with nonrecursive structures, whereas IIR filters utilize recursive realizations.

- Adaptive FIR filter realizations: The most widely used adaptive FIR filter structure is the transversal filter, also called tapped delay line, that implements an all-zero transfer function with a canonic direct form realization without feedback. For this realization, the output signal

$y(k)$ is a linear combination of the filter coefficients, that yields a quadratic mean-square error (MSE $= E[|e(k)|^2]$) function with a unique optimal solution. Other alternative adaptive FIR realizations are also used in order to obtain improvements as compared to the transversal filter structure, in terms of computational complexity, speed of convergence, and finite wordlength properties as will be seen later in the book.

- Adaptive IIR filter realizations: The most widely used realization of adaptive IIR filters is the canonic direct form realization [5], due to its simple implementation and analysis. However, there are some inherent problems related to recursive adaptive filters which are structure dependent, such as pole-stability monitoring requirement and slow speed of convergence. To address these problems, different realizations were proposed attempting to overcome the limitations of the direct form structure. Among these alternative structures, the cascade, the lattice, and the parallel realizations are considered because of their unique features as will be discussed in Chapter 10.

3) **Algorithm**: The algorithm is the procedure used to adjust the adaptive filter coefficients in order to minimize a prescribed criterion. The algorithm is determined by defining the search method (or minimization algorithm), the objective function, and the error signal nature. The choice of the algorithm determines several crucial aspects of the overall adaptive process, such as existence of sub-optimal solutions, biased optimal solution, and computational complexity.

1.3 INTRODUCTION TO ADAPTIVE ALGORITHMS

The basic objective of the adaptive filter is to set its parameters, $\boldsymbol{\theta}(k)$, in such a way that its output tries to minimize a meaningful objective function involving the reference signal. Usually, the objective function F is a function of the input, the reference, and adaptive-filter output signals, i.e., $F = F[x(k), d(k), y(k)]$. A consistent definition of the objective function must satisfy the following properties:

- Non-negativity: $F[x(k), d(k), y(k)] \geq 0, \forall y(k), x(k)$, and $d(k)$;
- Optimality: $F[x(k), d(k), d(k)] = 0$.

One should understand that in an adaptive process, the adaptive algorithm attempts to minimize the function F, in such a way that $y(k)$ approximates $d(k)$, and as a consequence, $\boldsymbol{\theta}(k)$ converges to $\boldsymbol{\theta}_o$, where $\boldsymbol{\theta}_o$ is the optimum set of coefficients that leads to the minimization of the objective function.

Another way to interpret the objective function is to consider it a direct function of a generic error signal $e(k)$, which in turn is a function of the signals $x(k)$, $y(k)$, and $d(k)$, i.e., $F = F[e(k)] = F[e(x(k), y(k), d(k))]$. Using this framework, we can consider that an adaptive algorithm is composed of three basic items: definition of the minimization algorithm, definition of the objective function form, and definition of the error signal.

1) **Definition of the minimization algorithm for the function** F: This item is the main subject of Optimization Theory [22]-[23], and it essentially affects the speed of convergence and computational complexity of the adaptive process.

In practice any continuous function having high-order model of the parameters can be approximated around a given point $\boldsymbol{\theta}(k)$ by a truncated Taylor series as follows

$$F[\boldsymbol{\theta}(k) + \Delta\boldsymbol{\theta}(k)] \approx F[\boldsymbol{\theta}(k)] + \mathbf{g}_{\boldsymbol{\theta}}^T\{F[\boldsymbol{\theta}(k)]\}\Delta\boldsymbol{\theta}(k) + \frac{1}{2}\Delta\boldsymbol{\theta}^T(k)\mathbf{H}_{\boldsymbol{\theta}}\{F[\boldsymbol{\theta}(k)]\}\Delta\boldsymbol{\theta}(k) \quad (1.1)$$

where $\mathbf{H}_{\boldsymbol{\theta}}\{F[\boldsymbol{\theta}(k)]\}$ is the Hessian matrix of the objective function, and $\mathbf{g}_{\boldsymbol{\theta}}\{F[\boldsymbol{\theta}(k)]\}$ is the gradient vector, further details about the Hessian matrix and gradient vector are presented along the text. The aim is to minimize the objective function with respect to the set of parameters by iterating

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + \Delta\boldsymbol{\theta}(k) \quad (1.2)$$

where the step or correction term $\Delta\boldsymbol{\theta}(k)$ is meant to minimize the quadratic approximation of the objective function $F[\boldsymbol{\theta}(k)]$. The so-called Newton method requires the first and second-order derivatives of $F[\boldsymbol{\theta}(k)]$ to be available at any point, as well as the function value. These informations are required in order to evaluate equation (1.1). If $\mathbf{H}_{\boldsymbol{\theta}}(\boldsymbol{\theta}(k))$ is a positive definite matrix, then the quadratic approximation has a unique and well defined minimum point. Such a solution can be found by setting the gradient of the quadratic function with respect to the parameters correction terms, at instant $k+1$, to zero which leads to

$$\mathbf{g}_{\boldsymbol{\theta}}\{F[\boldsymbol{\theta}(k)]\} = -\mathbf{H}_{\boldsymbol{\theta}}\{F[\boldsymbol{\theta}(k)]\}\Delta\boldsymbol{\theta}(k) \quad (1.3)$$

The most commonly used optimization methods in the adaptive signal processing field are:

- Newton's method: This method seeks the minimum of a second-order approximation of the objective function using an iterative updating formula for the parameter vector given by

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \mu\mathbf{H}_{\boldsymbol{\theta}}^{-1}\{F[e(k)]\}\mathbf{g}_{\boldsymbol{\theta}}\{F[e(k)]\} \quad (1.4)$$

  where μ is a factor that controls the step size of the algorithm, i.e., it determines how fast the parameter vector will be changed. The reader should note that the direction of the correction term $\Delta\boldsymbol{\theta}(k)$ is chosen according to equation (1.3). The matrix of second derivatives of $F[e(k)]$, $\mathbf{H}_{\boldsymbol{\theta}}\{F[e(k)]\}$ is the Hessian matrix of the objective function, and $\mathbf{g}_{\boldsymbol{\theta}}\{F[e(k)]\}$ is the gradient of the objective function with respect to the adaptive filter coefficients. It should be noted that the error $e(k)$ depends on the parameters $\boldsymbol{\theta}(k)$. If the function $F[e(k)]$ is originally quadratic, there is no approximation in the model of equation (1.1) and the global minimum of the objective function would be reached in one step if $\mu = 1$. For nonquadratic functions the value of μ should be reduced.

- *Quasi*-Newton methods: This class of algorithms is a simplified version of the method above described, as it attempts to minimize the objective function using a recursively calculated estimate of the inverse of the Hessian matrix, i.e.,

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \mu\mathbf{S}(k)\mathbf{g}_{\boldsymbol{\theta}}\{F[e(k)]\} \quad (1.5)$$

where $\mathbf{S}(k)$ is an estimate of $\mathbf{H}_{\boldsymbol{\theta}}^{-1}\{F[e(k)]\}$, such that

$$\lim_{k\rightarrow\infty} \mathbf{S}(k) = \mathbf{H}_{\boldsymbol{\theta}}^{-1}\{F[e(k)]\}$$

A usual way to calculate the inverse of the Hessian estimate is through the matrix inversion lemma (see, for example [21] and some chapters to come). Also, the gradient vector is usually replaced by a computationally efficient estimate.

- Steepest-descent method: This type of algorithm searches the objective function minimum point following the opposite direction of the gradient vector of this function. Consequently, the updating equation assumes the form

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \mu \mathbf{g}_{\boldsymbol{\theta}}\{F[e(k)]\} \tag{1.6}$$

  Here and in the open literature, the steepest-descent method is often also referred to as gradient method.

In general, gradient methods are easier to implement, but on the other hand, the Newton method usually requires a smaller number of iterations to reach a neighborhood of the minimum point. In many cases, *Quasi*-Newton methods can be considered a good compromise between the computational efficiency of the gradient methods and the fast convergence of the Newton method. However, the *Quasi*-Newton algorithms are susceptible to instability problems due to the recursive form used to generate the estimate of the inverse Hessian matrix. A detailed study of the most widely used minimization algorithms can be found in [22]-[23].

It should be pointed out that with any minimization method, the convergence factor μ controls the stability, speed of convergence, and some characteristics of residual error of the overall adaptive process. Usually, an appropriate choice of this parameter requires a reasonable amount of knowledge of the specific adaptive problem of interest. Consequently, there is no general solution to accomplish this task. In practice, computational simulations play an important role and are, in fact, the most used tool to address the problem.

2) **Definition of the objective function** $F[e(k)]$: There are many ways to define an objective function that satisfies the optimality and non-negativity properties formerly described. This definition affects the complexity of the gradient vector and the Hessian matrix calculation. Using the algorithm's computational complexity as a criterion, we can list the following forms for the objective function as the most commonly used in the derivation of an adaptive algorithm:

- Mean-Square Error (MSE): $F[e(k)] = E[|e(k)|^2]$;
- Least Squares (LS): $F[e(k)] = \frac{1}{k+1}\sum_{i=0}^{k} |e(k-i)|^2$;
- Weighted Least Squares (WLS): $F[e(k)] = \sum_{i=0}^{k} \lambda^i |e(k-i)|^2$, λ is a constant smaller than 1;
- Instantaneous Squared Value (ISV): $F[e(k)] = |e(k)|^2$.

The MSE, in a strict sense, is only of theoretical value, since it requires an infinite amount of information to be measured. In practice, this ideal objective function can be approximated by the other three listed. The LS, WLS, and ISV functions differ in the implementation complexity and in the convergence behavior characteristics; in general, the ISV is easier to implement but presents noisy convergence properties, since it represents a greatly simplified objective function. The LS is convenient to be used in stationary environment, whereas the WLS is useful in applications where the environment is slowly varying.

3) **Definition of the error signal** $e(k)$: The choice of the error signal is crucial for the algorithm definition, since it can affect several characteristics of the overall algorithm including computational complexity, speed of convergence, robustness, and most importantly for the IIR adaptive filtering case, the occurrence of biased and multiple solutions.

The minimization algorithm, the objective function, and the error signal as presented give us a structured and simple way to interpret, analyze, and study an adaptive algorithm. In fact, almost all known adaptive algorithms can be visualized in this form, or in a slight variation of this organization. In the remaining parts of this book, using this framework, we present the principles of adaptive algorithms. It may be observed that the minimization algorithm and the objective function affect the convergence speed of the adaptive process. An important step in the definition of an adaptive algorithm is the choice of the error signal, since this task exercises direct influence in many aspects of the overall convergence process.

1.4 APPLICATIONS

In this section, we discuss some possible choices for the input and desired signals and how these choices are related to the applications. Some of the classical applications of adaptive filtering are system identification, channel equalization, signal enhancement, and prediction.

In the system identification application, the desired signal is the output of the unknown system when excited by a broadband signal, in most cases a white-noise signal. The broadband signal is also used as input for the adaptive filter as illustrated in Fig. 1.2. When the output MSE is minimized, the adaptive filter represents a model for the unknown system.

The channel equalization scheme consists of applying the originally transmitted signal distorted by the channel plus environment noise as the input signal to an adaptive filter, whereas the desired signal is a delayed version of the original signal as depicted in Fig. 1.3. This delayed version of the input signal is in general available at the receiver in a form of standard training signal. In a noiseless case, the minimization of the MSE indicates that the adaptive filter represents an inverse model (equalizer) of the channel.

In the signal enhancement case, a signal $x(k)$ is corrupted by noise $n_1(k)$, and a signal $n_2(k)$ correlated to the noise is available (measurable). If $n_2(k)$ is used as an input to the adaptive filter with the signal corrupted by noise playing the role of the desired signal, after convergence the output

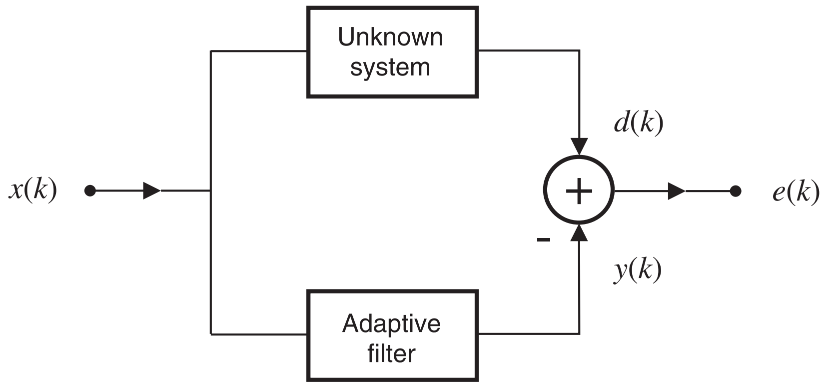

Figure 1.2 System identification.

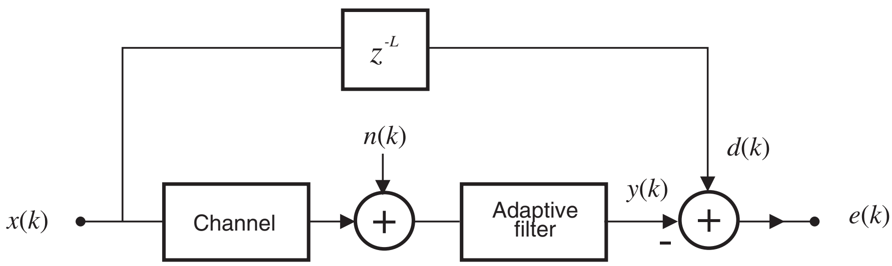

Figure 1.3 Channel equalization.

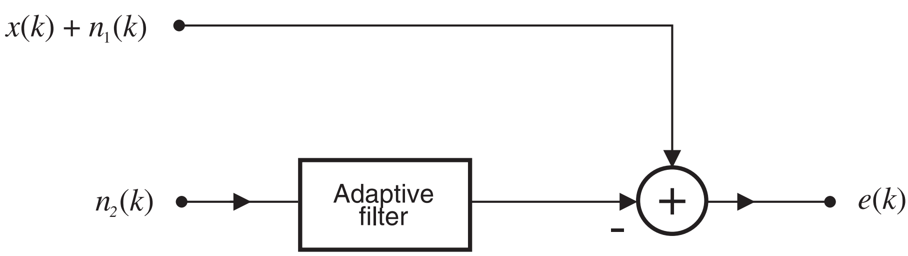

Figure 1.4 Signal enhancement ($n_1(k)$ and $n_2(k)$ are noise signals correlated to each other).

error will be an enhanced version of the signal. Fig. 1.4 illustrates a typical signal enhancement setup.

Finally, in the prediction case the desired signal is a forward (or eventually a backward) version of the adaptive-filter input signal as shown in Fig. 1.5. After convergence, the adaptive filter represents a model for the input signal, and can be used as a predictor model for the input signal.

Further details regarding the applications discussed here will be given in the following chapters.

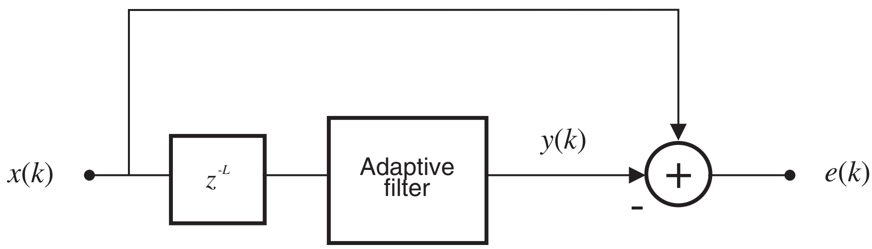

Figure 1.5 Signal prediction.

Example 1.1

Before concluding this chapter, we present a simple example in order to illustrate how an adaptive filter can be useful in solving problems that lie in the general framework represented by Fig. 1.1. We chose the signal enhancement application illustrated in Fig. 1.4.

In this example, the reference (or desired) signal consists of a discrete-time triangular waveform corrupted by a colored noise. Fig. 1.6 shows the desired signal. The adaptive-filter input signal is a white noise correlated with the noise signal that corrupted the triangular waveform, as shown in Fig. 1.7.

The coefficients of the adaptive filter are adjusted in order to keep the squared value of the output error as small as possible. As can be noticed in Fig. 1.8, as the number of iterations increase the error signal resembles the discrete-time triangular waveform shown in the same figure (dashed curve).

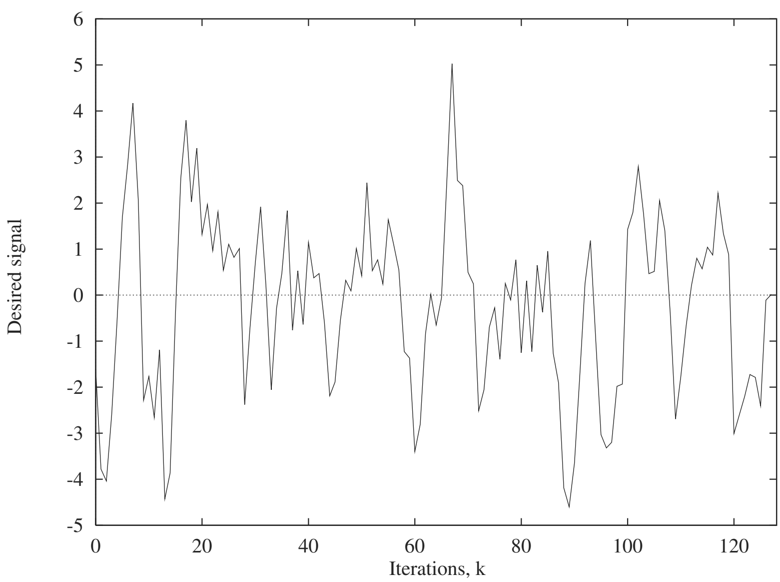

Figure 1.6 Desired signal.

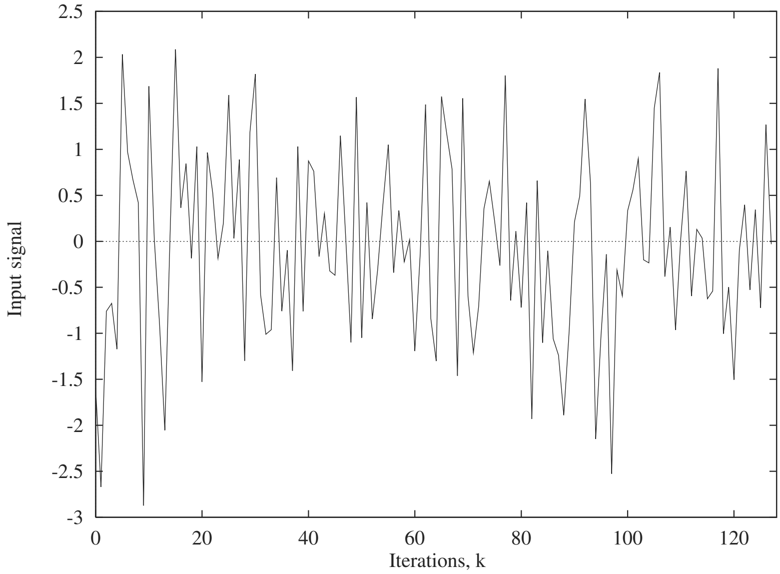

Figure 1.7 Input signal.

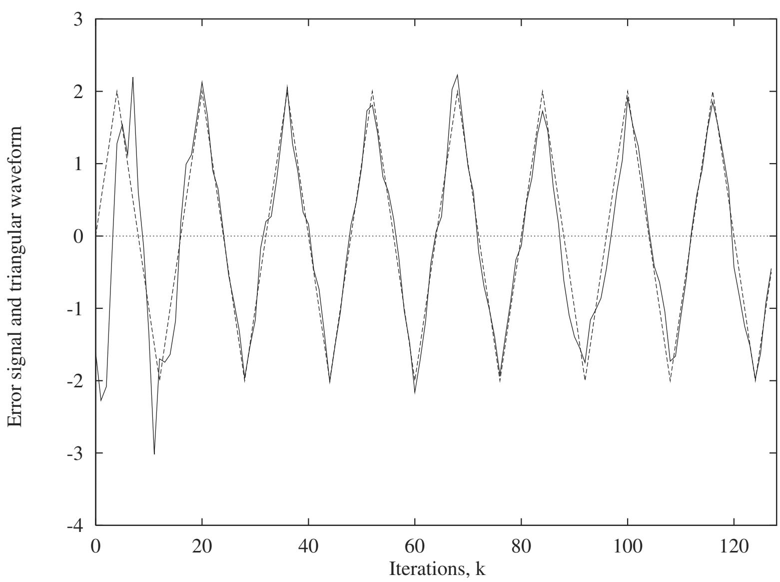

Figure 1.8 Error signal (continuous line) and triangular waveform (dashed line).

□

1.5 REFERENCES

1. P. S. R. Diniz, E. A. B. da Silva, and S. L. Netto, *Digital Signal Processing: System Analysis and Design*, Cambridge University Press, Cambridge, UK, 2002.

2. A. Papoulis, *Signal Analysis*, McGraw Hill, New York, NY, 1977.

3. A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals and Systems*, Prentice Hall, Englewood Cliffs, NJ, 2nd edition, 1997.

4. A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1989.

5. A. Antoniou, *Digital Signal Processing: Signals, Systems, and Filters*, McGraw Hill, New York, NY, 2005.

6. L. B. Jackson, *Digital Filters and Signal Processing*, Kluwer Academic Publishers, Norwell, MA, 3rd edition, 1996.

7. R. A. Roberts and C. T. Mullis, *Digital Signal Processing*, Addison-Wesley, Reading, MA, 1987.

8. J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 4th edition, 2007.

9. T. Bose, *Digital Signal and Image Processing*, John Wiley & Sons, New York, NY, 2004.

10. M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, algorithms, and Applications*, Kluwer Academic Publishers, Boston, MA, 1984.

11. S. T. Alexander, *Adaptive Signal Processing*, Springer Verlag, New York, NY, 1986.

12. M. Bellanger, *Adaptive Digital Filters*, Marcel Dekker, Inc., New York, NY, 2nd edition, 2001.

13. P. Strobach, *Linear Prediction Theory*, Springer Verlag, New York, NY, 1990.

14. B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1985.

15. J. R. Treichler, C. R. Johnson, Jr., and M. G. Larimore, *Theory and Design of Adaptive Filters*, John Wiley & Sons, New York, NY, 1987.

16. B. Farhang-Boroujeny, *Adaptive Filters: Theory and Applications*, John Wiley & Sons, New York, NY, 1998.

17. S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, NJ, 4th edition, 2002.

18. A. H. Sayed, *Fundamentals of Adaptive Filtering*, John Wiley & Sons, Hoboken, NJ, 2003.

19. L. R. Rabiner and R. W. Schaffer, *Digital Processing of Speech Signals*, Prentice Hall, Englewood Cliffs, NJ, 1978.

20. D. H. Johnson and D. E. Dudgeon, *Array Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1993.

21. T. Kailath, *Linear Systems*, Prentice Hall, Englewood Cliffs, NJ, 1980.

22. D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Addison Wesley, Reading, MA, 2nd edition, 1984.

23. A. Antoniou and W.-S. Lu, *Practical Optimization: Algorithms and Engineering Applications*, Springer, New York, NY, 2007.