USC–SIPI REPORT #275

Adaptive Interference Mitigation Techniques

by

Dae C. Shin and Chrysostomos L. Nikias

November 1994

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 404
Los Angeles, CA 90089-2564 U.S.A.

Contents

Chapter 1

Adaptive Filters and Interference Mitigation

1.1 Introduction and Motivation

When the characteristics of the data to be processed are stationary, the linear filter solution is the design of a Wiener filter that is optimal in the minimum mean square error (MMSE) sense [Kailath, 1974; 1981; Mendel, 1987]. The filter design requires *a priori* information about the statistics of the data. The filter is optimal only when the statistical characteristics of the input data match the a priori information on which the design of the filter is based. When this a priori information is not known completely, however, it may not be possible to design the Wiener filter or else the design may not be optimal. In such situations, a straightforward approach is the "estimate and plug" procedure [Haykin, 1991]. This is a two–stage procedure. First, the statistical characteristics of the data to be processed are estimated and then the obtained parameters are plugged into a *nonrecursive* formula for computing the filter parameters. For real time operation, this procedure has the disadvantage of requiring excessively complex and costly hardware.

A more efficient method is to use an *adaptive* filter [Widrow and Stearns, 1985; Haykin, 1991]. The adaptive filter is a *self-designing* scheme based on a *recursive algorithm*, which makes the filter perform satisfactorily in an environment where complete knowledge of the relevant signal characteristics is not available. In a stationary environment, it is found that after successive iterations the filter *converges* to the optimum Wiener solution in some statistical sense. In a nonstationary environment where the statistics of the input data is time–varying, the algorithm provides a *tracking* capability, provided that the variations are sufficiently slow. Usually, an adaptive filter is more complex and difficult to analyze than a non–adaptive filter. It should be noted that the adaptive filter can be considered as either a nonlinear device or a linear device. It is nonlinear in the sense that it does not obey the principle of superposition. Since the adaptive filter parameters are updated at each iteration, the parameters become data dependent. On the other hand, it is linear in the sense that the interesting estimate is obtained adaptively (at the output of the filter) as a linear combination of the filter inputs weighted by the filter coefficients.

Relying upon their "self–adjustable" property and "programmable" property by a training process, the adaptive filters have been successfully used in many fields such as control, communications, radar, sonar, seismology, mechanical design, navigation systems, etc. In this book, we restrict ourselves to the adaptive interference mitigation problem which is one of the most important applications of adaptive filtering. Although interference mitigation problems can be easily found in many books and journals of signal processing and communications, there is still no textbook addressing specifically the adaptive interference mitigation methods. This observation has motivated us to write this monograph on adaptive interference mitigation.

1.2 Historical Evolution

The earliest work on adaptive filter algorithms may be traced back to the 1950s. Widrow and Hoff have devised the least mean square (LMS) algorithm as a simple algorithm for adaptive transversal filters in 1959. Robbins and Monro [1951] have developed stochastic approximation methods. Plackett [1950] and other several researchers have worked on the recursive least squares (RLS) algorithm. Another algorithm related to the LMS algorithm is the gradient adaptive lattice (GAL) algorithm by Griffith [1977; 1978] using lattice filters instead of transversal filters. Godard [1974] used Kalman filter theory to propose a new class of adaptive transversal filtering algorithms. Gentleman and Kung [1981] have introduced another procedure using systolic arrays, the so called recursive QR decomposition least squares algorithm using Givens rotations. The most comprehensive textbook on adaptive filter theory was written by Haykin [1991].

1.3 Adaptive Algorithms

When the information of the statistical characteristics of the input data is not known completely, an efficient method to design the corresponding optimal filter is to use an adaptive algorithm. Although a wide variety of algorithms have been developed for adaptive filters, the choice of one algorithm versus another would be determined by various factors: convergence rate, misadjustment, computational complexity, tracking capability, robustness, algorithmic structure, and so on.

1.3.1 Development of Adaptive Algorithms

Although there is no unique classification for adaptive algorithms, we may identify the following typical four groups of adaptive algorithms depending on the basis of their development.

Algorithms Based on Mean–Squared Error

A tapped delay line or transversal filter is used as the basis for implementing the adaptive filter. The finite impulse response (FIR) of such a filter is defined by the number of tap weights (filter coefficients). To adaptively adjust the filter coefficients, we first modify the system of Wiener–Hopf equations using the method of steepest descent and then use instantaneous values for the autocorrelation of the tab inputs and cross–correlation between the desired response and the tab inputs to derive an estimate of the gradient. The resulting algorithm is known as the least mean square (LMS) algorithm [Widrow and Stearns, 1985; Haykin, 1991; Clarkson, 1993]. The LMS algorithm is simple and capable of achieving satisfactory performance under the right conditions. Its major drawbacks are a relatively slow convergence rate and a sensitivity to the eigenvalue spread defined as the ratio of the maximum to minimum eigenvalues of the correlation matrix of the tap inputs.

Algorithms Based on Kalman Filter

When a linear dynamic system is described by the state–space model consisting of two basic equations: the plant equation and the measurement equation, the Kalman filter solution can be expressed as time–update recursive equations [Kalman, 1960; Kalman and Bucy, 1961; Sorenson, 1970]. With a transversal filter used to provide the structure base, we may identify the filter coefficient vector as the state vector.

By adopting the idealized state model and by making identification of the corresponding parameters, we may use the recursive solution to the Kalman filtering problem to derive a number of different recursive algorithms for updating the filter weights [Haykin, 1991]. The algorithm provides faster convergence rate, is insensitive to the eigenvalue spread, and deals with a stationary or nonstationary environments. Its major limitation is high computational complexity.

Algorithms Based on Least Squares

In algorithms based on least squares, we minimize a performance index that consists of the sum of weighted error squares, where the error is defined as the difference between desired response and filter output. Note that the method of least squares differs from the two previous algorithms in the sense that the former algorithm is deterministic in its formulation and the latter ones are based on statistical concepts.

There are two basic categories for methods of least squares [Haykin, 1991]: block estimation approach and recursive estimation approach. In the block approach, the input data is arranged in several blocks and the filter coefficients are updated on a block–by–block basis. In the recursive approach, the filter coefficients are updated on a sample–by–sample basis.

- Recursive Least Squares (RLS) [Morf *et al.*, 1976; Falconer and Ljung, 1978]: The algorithm uses a transversal filter and its derivation relies on a basic result from the matrix–inversion lemma.

- Least Squares Lattice (LSL) [Friedlander, 1982b; 1982c; Lev–Ari *et al.*, 1984]: The issue of computational complexity is resolved by using a multistage lattice predictor as the structural basis for implementing the adaptive filter. An important property of the

multistage lattice predictor is that its individual stages are decoupled from each other in a time–averaged sense.

- QR Decomposition Least Squares (QRDLS) [Haykin, 1991]: The algorithm consists of an iterative open–loop configuration while the other algorithms use iterative closed–loop configurations. At the first stage, the input data matrix is recursively transformed into upper triangular form by the QR decomposition method. The QR decomposition is a matrix factorization to express a matrix as a product of an orthogonal matrix and an upper triangular matrix. Then the special structure of the data matrix is exploited to compute the least squares weight vector. This algorithm is stable, robust, rapidly convergent, and computationally efficient. It may be implemented using systolic arrays well suited for VLSI implementation.

Algorithms Based on Nonlinear Criteria (Non–MSE)

Mean–square error (MSE) or least–squares criteria used in adaptive filtering lead to second–order statistics of the signals associated with the adaptation process. The limitation of second–order statistics is that they are phase blind and that they cannot extract information related to nonlinear and/or higher–order statistics. When the input data is non–Gaussian, using a nonlinear cost function, the filter is enabled to extract information (particularly phase information) from the input signal in a more efficient manner. An important example of an adaptive filter with non–MSE creteria is blind deconvolution (or equalization). This problem can be found in the literature of [Benveniste and Goursat, 1984], [Godard, 1980], [Chen and Nikias, 1991], [Hatzinakos and Nikias, 1991], etc.

Another method that overcomes the limitation of second–order statistics is to use higher–order statistics (HOS) when the input data has non–zero higher–order cumulants. The use of HOS has drawn a lot of attention due to the fact that they preserve phase information, are blind to any Gaussian process, and detect and characterize nonlinear properties in time series [Nikias and Petropulu, 1993; Nikias and Mendel, 1993].

Recently, many researchers devoted themselves to studying impulsive phenomena. Especially, when the input data is characterized by a stable process [Shao and Nikias, 1993; 1995], the use of second–order or higher–order statistics is not possible because of the high or infinite variance of impulsive–like signals. In such situations, fractional lower order moments (FLOM) may play an important role instead second– and higher–order moments. Also, new criteria based on the FLOM could be considered for stable signal processing.

1.3.2 Adaptive Filter Structures

To implement an adaptive filter, we can use the transversal or finite impulse response (FIR) filter, the infinite impulse response (IIR) filter , and lattice structure filter.

Transversal Filter

The transversal filter is also referred to as a tapped delay line. It consists of three elements: unit delay element, multiplier, and adder. The number of delay elements used in the filter determines the duration of its impulse response [Widrow and Stearns, 1985; Haykin, 1991; Clarkson, 1993].

Lattice Filter

A lattice filter is modular in structure in that it consists of a number of individual stages, each of which has the appearance of a lattice. Each stage of the lattice filter is described by the pair of input–output relations among the corresponding forward prediction error, backward prediction error, and reflection coefficients [Makhoul, 1977; Haykin, 1991; Clarkson, 1993].

IIR Filter

The feature that distinguishes an infinite impulse response (IIR) filter from finite impulse response (FIR) filters that are previously described is the inclusion of feedback paths [Johnson, 1984; Widrow and Stearns, 1985; Clarkson, 1993]. On the one hand, the presence of feedback makes the duration of the impulse response of an IIR filter infinitely long. On the other hand, it may introduce instability of the filter. It is possible for an IIR filter to become unstable, unless the feedback coefficients are chosen very carefully. To avoid the IIR filter from being unstable, a stability test should be performed at each iteration. If unstable, the corresponding parameters need to be projected back into a stable region.

1.4 Applications of Adaptive Filtering

Adaptive filtering has been successfully applied in many fields: communication, radar, sonar, seismology, biomedical engineering, and so on. Depending on the desired response, we may distinguish four basic classes of adaptive filtering applications: identification, inverse modeling, prediction, and interference cancellation [Haykin, 1991]. Among many applications of the adaptive filtering, we describe a few selected topics.

- System Identification: When we have an unknown time invariant (or time varying) dynamic system with sets of discrete time stationary input signal and its corresponding measurements, the requirement is to identify the unknown system [Goodwin and Payne, 1977] in the form of a transversal filter.

- Equalization: The transmission of digital signal through a linear communication channel is limited by the following two factors: intersymbol interference and additive thermal noise. The intersymbol interference is usually the major concern. Since the characteristics of the channel is unknown/random in practice, an adaptive equalizer is desired to adequately reduce intersymbol interference [Lucky, 1965; 1966; Godard, 1980].

- Speech Processing: There are two broad classes to represent speech signals: source coders (vocoders) and waveform coders. An example of source coding is linear predictive coding (LPC) in which linear prediction is used to estimate the speech parameters [Rabiner and Schafer, 1978]. A linear prediction of the present sample of a speech signal is adaptively obtained by using a set of its past samples. An example of waveform coding is adaptive differential pulse code modulation (ADPCM) that is DPCM [Cutler, 1952] utilizing an adaptive quantizer and an adaptive predictor for further reduction in the transmission rate [Cummiskey *et al.*, 1973; Noll, 1975].

- Spectrum Analysis: In parametric spectrum analysis, the power spectrum is estimated by assuming a model for the process. One of practical models is the autoregressive (AR) model excited by a white noise process [Marple, 1987; Kay, 1988]. When the AR parameters are assumed to be nonstationary, an adaptive prediction

error filter can be applied to determine the AR parameters and the instant power spectrum.

- Detection: In several environments, there may be inadequate information on the signal, and noise statistics to design a fixed optimal detector like Bayes or Neyman–Pearson detectors [Helstrom, 1968; Van Trees, 1968]. An adaptive detector exploits some knowledge of general characteristics of the signal and noise, and adjusts itself in response to changes in the received signal. One example of the adaptive detection is the detection of narrowband signal using an adaptive line enhancer in a background of additive broadband noise [Widrow *et al.*, 1975; Zeidler *et al.*, 1978; Nehorai and Malah, 1980].

- Noise Cancellation: The noise canceler subtracts noise from a received signal and improves output signal–to–noise ratio (SNR). An adaptive noise canceler is a dual–input (primary and reference inputs), closed loop adaptive control system. The structure of the adaptive noise cancellation will be described in Chapter 3 in detail. Useful applications of the adaptive noise cancellation include canceling 60 Hz interference in elctrocardiography [Widrow *et al.*, 1975], reducing background (acoustic) noise in speech [Boll and Pulsipher, 1980; Clarkson, 1993], canceling echo in a long distance call made on hybrid telephone connection [Sondhi, 1967; Rosenberger and Thomas, 1971], and etc.

- Beamforming: This is a spatial form of adaptive signal processing that finds applications in radar and sonar. A number of independent sensors, which are placed at different points in space, receive the signal. A beamformer distinguishes between the spatial properties of signal and noise. In order to enable a beamformer to respond

to an unknown interference, it has to be adaptively performed so as to automatically place nulls in the directions of the sources of interference [Widrow *et al.*, 1967; Frost, 1972; Compton, 1988].

Among these several applications, we have restricted ourselves here to the adaptive interference cancellation problem throughout the book[1] as shown in Fig. 1.1. Although there are three categories of interference

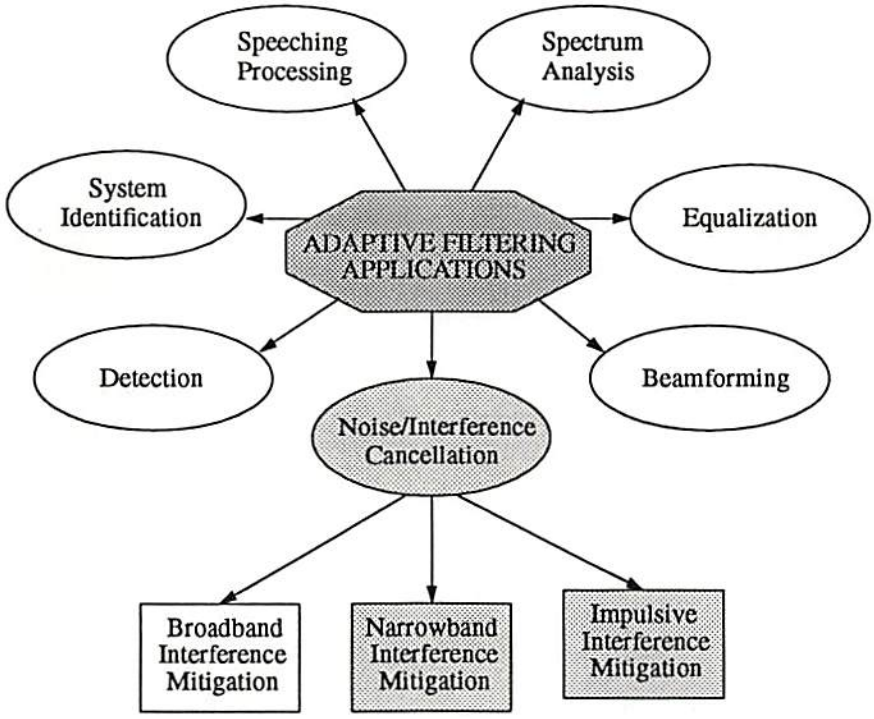

Figure 1.1: Configuration of adaptive interference (noise) mitigation problems among the applications of adaptive filtering.

[1]Traditionally, the terms *adaptive noise cancellation* and *adaptive noise canceler* have been used. Throughout the book, however, we use *adaptive interference mitigation* and *adaptive interference canceler* instead because it is desired to clearly distinguish between interference and background (thermal) uncorrelated noise.

mitigation problems: broadband interference mitigation, narrowband interference mitigation, and impulsive interference mitigation as shown in Fig. 1.1, the book covers narrowband and impulsive interference mitigation problems only.

1.5 Types of Interference Models

To represent a signal a variety of models has been used in the literature. Here, we divide interferences into two large groups: Gaussian interferences and non–Gaussian interferences. Non–Gaussian interferences include any signal that is not Gaussian. Each group contains two members categorized by the bandwidth of the signal: narrowband and wideband (or broadband). The spectrum of a narrowband signal is smaller than the passband of the receiver, i.e., the spectrum of the signal of interest (SOI). The spectrum of a wideband signal is broader than the passband of the receiver. The configuration of the types of interference models is illustrated in Fig. 1.2. Note that the generalized Gaussian and stable interferences are special forms of non–Gaussian signals.

1.5.1 Gaussian Interference

The most popular model for interference or noise is a Gaussian model due to its simple form of the distribution and its justification from the use of the central limit theorem [Davenport and Root, 1987].

Narrowband Gaussian Interferences

One way to model a Gaussian narrowband interference or noise is to use a colored Gaussian process occupying a small bandwidth in the frequency domain. An example for this model is a narrowband Gaussian

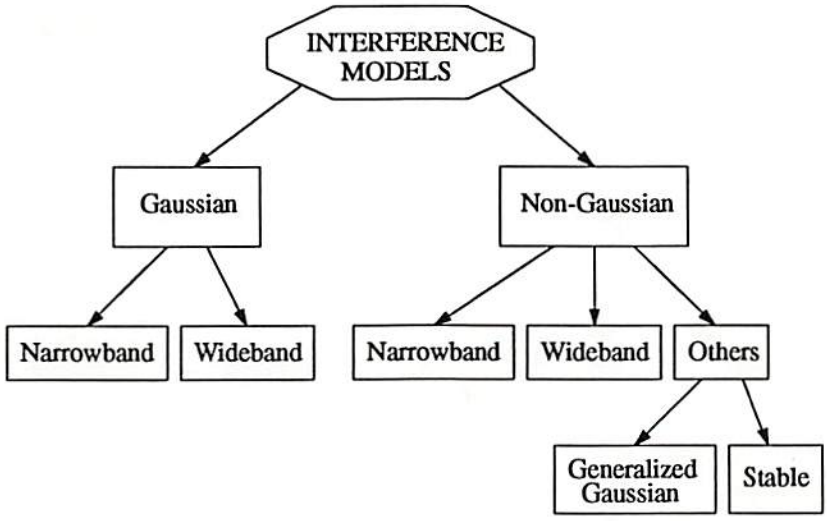

Figure 1.2: Categorization of types of interference models.

interference expressed as [Iltis and Milstein, 1984; Masry and Milstein, 1986]

$$I(k) = I_c(k)\ \cos((\omega_0 + \delta\omega)k) - I_s(k)\ \sin((\omega_0 + \delta\omega)k) \qquad (1.1)$$

where $I_c(k)$ and $I_s(k)$ denote the in–phase and quadrature terms, respectively. In (1.1), $\delta\omega$ denotes an interference offset from the carrier frequency.

Another way to present a wide class of narrowband Gaussian interferences is by using an all–pole (or AR) filter. The narrowband interference is generated by an autoregressive (AR) process of order p excited by a Gaussian white process:

$$I(k) = \sum_{i=1}^{p} \phi_i\, I(k-i) + e(k) \qquad (1.2)$$

where $\{e(k)\}$ is a white Gaussian process. In the expression, $\{\phi_i,\ i = 1, 2, \ldots, p\}$ denotes the AR parameters. The model for a narrowband

interference can be found in [Vijayan and Poor, 1990; Garth and Poor, 1992].

Wideband Gaussian Interferences

The most popular way to model a Gaussian wideband interference or noise is to pass a white Gaussian process through a wideband filter. Another way is to use a colored Gaussian process occupying a broad bandwidth in the frequency domain [Milstein and Das, 1983].

1.5.2 Non–Gaussian Interference

For many signal processing or communication problems, the usual Gaussian assumption is inadequate due to the occurrence of high amplitude "spikes" with low probability. These impulsive components of the interference have been observed in many real world applications. Examples include atmospheric noise, where lightning discharges in the vicinity of the receiver can cause such spikes [Watt and Maxwell, 1957; Ibukun, 1966], and underwater problems such as sonar and submarine communication, where the ambient acoustical noises may include impulses due to noisy aquatic animals such as snapping shrimp or due to ice cracking in arctic regions. There is a great variety of man–made non–Gaussian noise sources [Middleton, 1972; 1973] such as automobile ignitions, neon lights, and other electronic devices.

For modeling these non–Gaussian, impulsive phenomena, the following two approaches can be used:

- Empirical models: They are developed to fit collected data often with little regard for the underlying physical mechanisms [Trunk, 1970; Sorenson and Alspach, 1971; Miller and Thomas, 1976]

- Physical models: They attempt to model these mechanisms directly.

We concentrate here on a number of various physical models for non–Gaussian interferences.

Narrowband Non–Gaussian Interferences

The most popular model for narrowband signal representation is to use a finite sum of sinusoidal signals as follows.

$$I(k) = \sum_{m=1}^{M} a_m \cos(2\pi f_m k + \phi_m) \tag{1.3}$$

where M is a positive integer. In the expression, a_m and f_m denote the amplitude and frequency of each sinusoid, respectively. The phase ϕ_m is a value over the interval of $[-\pi, \pi]$. This model can be found in [Hsu and Giordano, 1978], [Milstein and Das, 1980], [Ketchum and Proakis, 1982], [Proakis, 1982], [Milstein, 1988], etc.

Another way is to use a zero–mean wide–sense stationary process with the following covariance sequence [Masry, 1984; 1985].

$$\rho_I(k) = \sigma_I^2 \, \alpha^{|k|} \tag{1.4}$$

where σ_I^2 is a positive constant and $0 < \alpha < 1$. The spectral density of the interference exists and is given by

$$\begin{aligned} S_I(\omega) &= \frac{(\sigma_I^2/2\pi)\,(1-\alpha^2)}{|\,1-\alpha\, e^{i\omega}\,|^2} \\ &= \frac{(\sigma_I^2/2\pi)\,(1-\alpha^2)}{1+\alpha^2-2\alpha\cos\omega}. \end{aligned} \tag{1.5}$$

Note that as the value of α approaches to unity, the spectral density becomes more sharply peaked.

We can also model a non–Gaussian narrowband interference or noise using the Middleton Class A model [Middleton, 1977; 1979a, 1979b]. The associated probability density function (pdf) is given by

$$f_A(\epsilon) \approx e^{-A} \sum_{m=0}^{\infty} \frac{A^m \, \epsilon \, e^{-\epsilon^2/2\hat{\sigma}_m^2}}{m! \, \hat{\sigma}_m^2} \tag{1.6a}$$

where ϵ is the normalized envelop and

$$\begin{aligned} \hat{\sigma}_m^2 &= \frac{m/A + \Gamma_A}{2(1+\Gamma_A)} \\ \epsilon &= \frac{E}{\sqrt{2\,\Omega_A(1+\Gamma_A)}}. \end{aligned} \tag{1.6b}$$

In the expression, the parameter E denotes the instantaneous envelope and the parameters (A, Γ_A, Ω_A) are called "global" parameters briefly stated as follows:

- A (overlap index) is defined as the average number of source emission "events" impinging on the receiver per second, times the mean duration of a typical interfering source emission. The smaller A, the fewer such events and/or their duration. As A is made large, the statistics or the instantaneous amplitude approaches Gaussian statistics.

- Γ_A (Gaussian factor) is defined as the ratio of the intensity of the independent Gaussian component of the input interference to the intensity of the non–Gaussian component.

- Ω_A is defined as the intensity of the impulsive non–Gaussian component.

Wideband Non–Gaussian Interferences

The Middleton Class B model [Middleton, 1977; 1979a; 1979b] can be used for wideband non–Gaussian signals and requires two characteristic

functions. One is for small and intermediate values of the envelope and the other is for the large values. The associated pdf's are given by, for an appropriate threshold value of the envelope ϵ_B,

$$f_B(\epsilon) \approx \begin{cases} 2\epsilon \sum_{m=0}^{\infty} \frac{(-\hat{A}_\alpha)^m}{m!} \Gamma\left(\frac{2+m\alpha}{2}\right) {}_1F_1\left(\frac{2+m\alpha}{2}; 1; -\epsilon^2\right) & 0 \le \epsilon \le \epsilon_B \\ \frac{e^{-A}}{4G^2} \sum_{m=0}^{\infty} \frac{A^m}{m!} \epsilon \frac{e^{-\epsilon^2/2\hat{\sigma}_m^2}}{\hat{\sigma}_m^2} & \epsilon_B \le \epsilon < \infty \end{cases} \tag{1.7a}$$

where $0 < \alpha < 2$, ${}_1F_1$ is a confluent hypergeometric function [Middleton, 1960; 1977], and

$$\begin{aligned} \hat{A}_\alpha &= A\left(\frac{b_{1\alpha}\, a_B}{G}\right)^\alpha, \qquad a_B^2 = \frac{1}{2\Omega_B(1+\Gamma_B)} \\ \hat{\sigma}_m^2 &= \frac{m/\hat{A}_B + \Gamma_B}{2(1+\Gamma_B)}, \qquad \hat{A}_B = \left(\frac{2-\alpha}{4-\alpha}\right) A \\ G^2 &= \frac{\frac{4-\alpha}{2-\alpha} + \Gamma_B}{4(1+\Gamma_B)}. \end{aligned} \tag{1.7b}$$

There are six "global" parameters $(A_\alpha, \alpha, A, \Gamma_B, \Omega_B, N_I)$ briefly stated as follows:

- The subset (A, Γ_B, Ω_B) are just as for Class A narrowband interferences.

- $b_{1\alpha}$ is the structure factor given by

$$b_{1\alpha} = \frac{\Gamma(1-\alpha/2)}{2^{\alpha/2-1}\,\Gamma(1+\alpha/2)} < (\hat{B}_0/\sqrt{2})^\alpha > \tag{1.8}$$

  where $< \hat{B}_0^\alpha >$ is the α–moment (fractional moment) of the envelop $\hat{B}_0$ of the typical emission at the output of the front–end stages of the receiver. The structure factor influences the shape of the pdf.

- α is the spatial density–propagation parameter given by

$$\alpha = \begin{cases} \frac{2-\mu}{\gamma}\Big|_{\text{surface}} \\ \\ \frac{3-\mu}{\gamma}\Big|_{\text{vol}} \end{cases} \tag{1.9}$$

where μ and γ are the power–law exponents associated with the range dependence of the density distribution of the possibly emitting sources and their propagation, respectively. It provides an effective measure of the average source density with range.

- N_I is a scaling factor on ϵ.

Note that there is the Middleton Class C interference which is the sum of the Class A and the Class B interferences. But, it can be shown to be reducible to an equivalent Class B form [Middleton, 1979a].

Other Non–Gaussian Interferences

A non–Gaussian family of symmetric probability distributions is the generalized Gaussian distribution defined as [Gray, 1979]

$$f(x,\alpha,\beta) = \frac{\alpha}{2\beta\Gamma(\frac{1}{\alpha})}\, e^{-\left(\frac{|x|}{\beta}\right)^{\alpha}} \tag{1.10a}$$

where

$$\begin{array}{ll} -\infty < x < \infty & \\ \Gamma(\cdot) & \text{is the gamma function;} \\ \alpha > 0 & \text{is the shape parameter;} \\ \beta > 0 & \text{is the scale parameter.} \end{array} \tag{1.10b}$$

The generalized Gaussian family covers a wide range of distributions including those having heavier tails than the Gaussian distribution. The double exponential distribution (Laplace distribution), Gaussian distribution, and uniform distribution are special cases of the generalized Gaussian distribution. The generalized Gaussian distribution will be revisited in Chapter 6.

Some impulsive interferences can be characterized by stable processes [Shao and Nikias, 1993; 1995] whose characteristic function is given by

$$\phi(t) = \exp\{\, jat - \gamma\,|t|^{\alpha}\ [\,1 + j\beta\,\mathrm{sign}(t)\,w(t,\alpha)\,]\,\} \tag{1.11a}$$

where

$$\mathrm{sign}(t) = \begin{cases} 1 & \text{for } t > 0 \\ 0 & \text{for } t = 0 \\ -1 & \text{for } t < 0 \end{cases} \tag{1.11b}$$

$$w(t,\alpha) = \begin{cases} \tan\left(\frac{\alpha\pi}{2}\right) & \text{for } \alpha \neq 0 \\ \frac{2}{\pi}\log|t| & \text{for } \alpha = 1. \end{cases} \tag{1.11c}$$

The stable distribution has four parameters: location parameter (a), index of skewness (β), characteristic exponent (α), and dispersion parameter (γ). The stable distributions are called symmetric α–stable (SαS) distributions when the distribution with the characteristic exponent α is symmetric. The value of α is constrained to be $0 < \alpha \leq 2$. They are smooth, unimodal, and bell shaped. When $\alpha = 2$, the distribution becomes Gaussian. The smaller α is, the heavier the tails. One of their important characteristics is their infinite variance unless $\alpha = 2$. The important definitions and useful properties of the SαS distributions are described in Chapter 7.

1.6 Organization of the Book

It is assumed that the reader is familiar with fundamental knowledge and tools of signal processing such as discrete time sequences, linear system, convolution, Fourier transforms, linear algebra, etc. It is also assumed that the reader has knowledge of basic probability and random variables.

This book deals with a number of techniques of interference mitigation. As shown in Fig. 1.1, the book covers narrowband and impulsive interference mitigation using digital filters. The digital filters consist of fixed and adaptive filters. After introducing fixed digital filter techniques, we concentrate, however, on the use of adaptive filters illustrated

in Fig. 1.3. The adaptive digital filters may be divided into three groups as shown in Fig. 1.4, depending on the availability of reference signals, filter structures, and criteria. The reference signal availability group consists of single sensor and multiple sensor problems. The filter structure group may be divided into transversal, IIR, and lattice filters. The criterion may be based on linear, nonlinear, second–order statistics, and higher–order statistics (HOS) criteria. For example, the adaptive noise canceler is using the LMS algorithm (second–order statistics criterion), a transversal filter realization, and multiple sensors.

Chapter 2 introduces the narrowband interference mitigation problem using conventional *fixed notch filters*. When the additive interference is stationary sinusoidal with *a priori* known characteristics (e.g., amplitude and frequency), digital fixed notch filters are described. The realization structures and the corresponding critical design parameters of a second–order and a sixth–order notch filters are considered. In addition, their

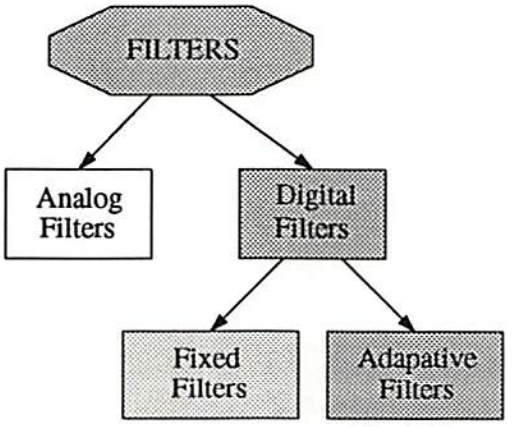

Figure 1.3: Configuration of types of filters.

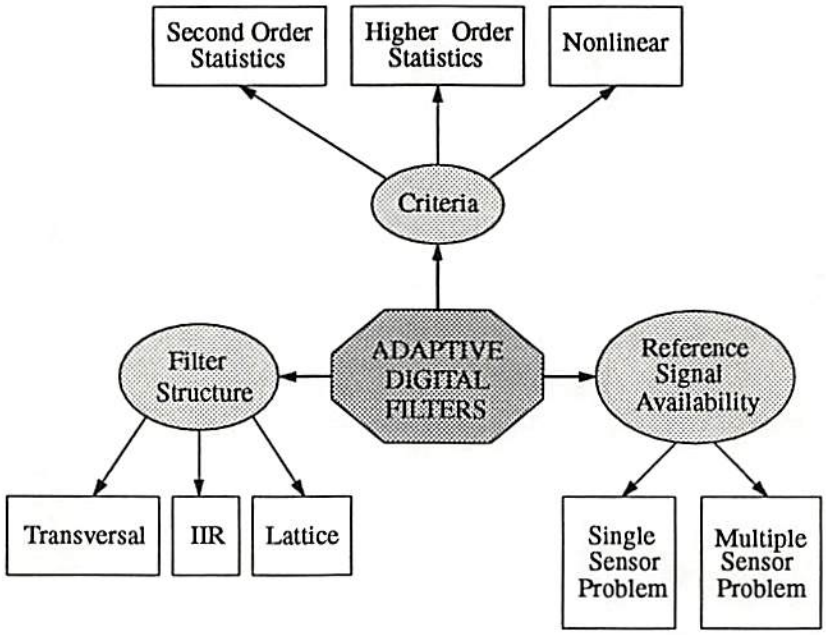

Figure 1.4: Configuration of adaptive digital filters.

performance with different sets of the critical design parameters are presented through computer simulations when the interference is stationary (time invariant). We include an example of nonstationary (time varying) interference mitigation problem and justify the need of *adaptive interference mitigation* (AIM) methods in practice.

Chapter 3 considers AIM methods based on second–order statistics. The AIM methods are categorized into two different classes. One class assumes availability of two inputs: primary and reference inputs. The other class is the AIM problem when only a single sensor receives the signal of interest (SOI) corrupted by additive interference and background

noise. The structures of these AIM methods are transversal, IIR, or lattice filters. Note that all the AIM algorithms described in Chapter 3 are linear and are based on second–order statistics. We present the description of each algorithm, its computational complexity, and its corresponding advantages and disadvantages. In addition, we demonstrate the performance of each AIM algorithm through computer simulations.

Chapter 4 describes AIM methods based on nonlinear cost functions (criteria). These are AIM methods based on Taylor series expansion, nonlinear error, least mean fourth, median filter, constant modulus algorithm (CMA), and criterion with memory nonlinearity (CRIMNO). The advantages and disadvantages of each method are discussed as well as the corresponding computational complexity. In addition, their performance of narrowband interference mitigation is presented through computer simulations.

Chapter 5 deals with AIM methods based on higher–order statistics (HOS). When the interference can be considered as a non–Gaussian process whose nth–order ($n > 2$) statistics are not identically zero, the HOS–based AIM algorithm is derived from a "criterion of goodness" in nth–order cumulant domains. The criterion consists of three components: the cross–cumulants between the primary and reference input measurements, the auto–cummulants of the reference input measurement, and the adaptive filter coefficients. To minimize the criterion, the gradient type algorithm is employed. Although using HOS requires more computational complexity, the HOS–based AIM techniques have the following advantages over the second–order statistics–based ones: (1) the filter coefficient update equation is independent from uncorrelated noise sources that are assumed to be Gaussian, (2) the performance is less sensitive to the statistics of the reference signal, and (3) the convergence rate is faster and excess error is smaller. We develop the AIM method

based on fourth–order statistics and present its required computational complexity. In addition, we demonstrate and compare the performance of the AIM method based on fourth–order statistics with second–order statistics–based AIM methods.

Chapter 6 covers single sensor AIM techniques when a narrowband interference corrupts the broadband DSSS SOI transmitted over a noisy channel. The reference signal for the interference is not available and the transmitted SOI is received by a single sensor. The noise channel distributions are Gaussian or generalized Gaussian. Since the received signal is not Gaussian, linear AIM methods cannot be optimal. Thus, nonlinear AIM methods are introduced to effectively mitigate the interference. The nonlinear AIM methods are derived from the approximate conditional mean (ACM) filter. We present the nonlinear AIM method for generalized Gaussian noise channels as well as those for Gaussian and mixed Gaussian noise channels. Their performance is compared to that of linear AIM methods in terms of the improvement of signal to interference plus noise ratios and the error curves between the restored SOI and the original SOI, when the narrowband interference is a single sinusoidal signal or a signal generated by an AR system.

Chapter 7 considers AIM methods for mitigating impulsive interferences characterized by symmetric α–stable (SαS) distributions. First, the definitions and useful properties of SαS processes are briefly described. When the interference is characterized by SαS distributions, the usual criteria and analyses for AIM methods described in the previous chapters cannot be applied because a SαS process does not have a finite variance. To mitigate the impulsive interference we then describe AIM methods based on pth order criterion, where $p < \alpha$, as well as a variety of AIM methods based on LMS algorithm. Note that the AIM methods based on LMS–typed algorithm can be used because of an instant gradient es-

timate. The performance of each AIM method is presented when the impulsive interference is characterized with different value of the characteristic exponent α.

Chapter 8 is devoted to the concluding remarks and future works on adaptive interference mitigation problems.

Chapter 2

Conventional Notch Filters

2.1 Introduction

In many signal processing applications we frequently encounter the problem of removing additive sinusoidal interference without seriously distorting the signal of interest (SOI). When the characteristics of a sinusoidal interference (e.g. amplitude and frequency) are stationary and their estimation is precisely available, we can use a fixed digital notch filter to effectively mitigate the interference. In this chapter we introduce the problem definition and representation of digital notch filters as well as their design techniques and properties. In particular, second– and sixth–order digital notch filters are described, and several examples that demonstrate their performance are given.

2.2 Problem Definition and Preliminaries

Let us assume that the SOI, $\{s(t)\}$, is corrupted by an additive sinusoidal interference, $\{I(t)\}$,

$$x(t) = s(t) + I(t) \tag{2.1}$$

where the interference $\{I(t)\}$ can be represented by

$$I(t) = A \sin(\omega_0 t). \tag{2.2}$$

Note that the amplitude (A) and frequency (ω_0) of the sinusoidal interference are assumed to be constants. The ideal interference cancellation filter should have a response $\{s(t)\}$ to the input $\{x(t)\}$. In the frequency domain, the required ideal linear filter, $H_{ideal}(j\omega)$, should have unity gain for all frequencies except at ω_0 where its gain is zero as illustrated in Fig. 2.1. This ideal filter processor is called a *notch filter* with notch at ω_0. Unfortunately, the ideal notch filter is not physically realizable and must be approximated in practice as illustrated in Fig. 2.2. If one would attempt to implement a notch filter approximation using analog devices (resistors, capacitors, and inductors), the futility of the approach would be quickly realized because of component tolerances, component drift, accuracy, physical size, and reliability. On the other hand, one may easily design a digital filter whose frequency response closely resembles that of the ideal filter.

Assuming that the sampling period T is small enough so that little distortion results from the analog–to–digital (A/D) conversion of the desired signal $\{s(t)\}$, we use the sample sequence of the signal, $\{x(k)\}$, where $t = Tk$ as the input to a digital filter governed by

$$y(k) = \sum_{i=0}^{m} b_i\, x(k-i) - \sum_{j=1}^{n} a_j\, y(k-j) \tag{2.3}$$

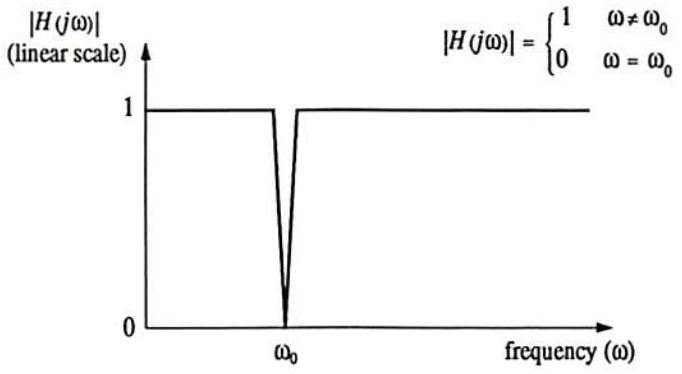

Figure 2.1: Frequency characteristics of an ideal notch filter.

where $\{y(k)\}$ denotes the filter output. The problem is to select the coefficients $\{a_i\}$ and $\{b_i\}$ so that the frequency response of the resulting digital filter will be similar to that of the ideal notch filter. Figure 2.2 illustrates the frequency response of a practical notch filter with a 3–dB rejection bandwidth b and notch frequency ω_0. As we see later in this chapter, both b and ω_0 are critical design parameters.

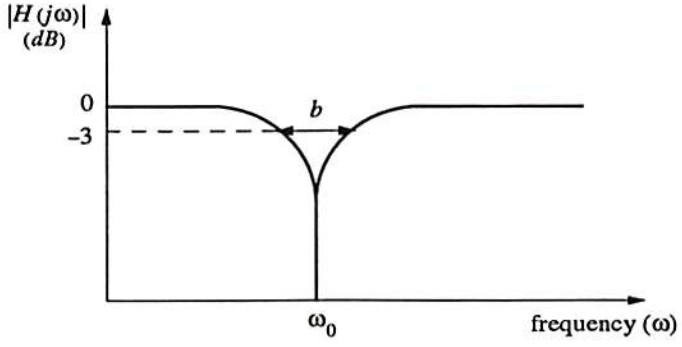

Figure 2.2: Frequency characteristics of a digital notch filter with a 3–dB rejection bandwidth b and a notch frequency at ω_0.

2.3 Gain of a Digital Notch Filter

Let us consider the transfer function, $H(z)$, of the digital filter described in (2.3), then it becomes

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}}. \tag{2.4}$$

After multiplying the numerator and denominator of the transfer function by z^n and factoring, we obtain

$$H(z) = \frac{b_0 (z - z_1)(z - z_2) \cdots (z - z_k)}{(z - p_1)(z - p_2) \cdots (z - p_\ell)} \tag{2.5}$$

where k and ℓ denote the number of zeros and poles of the transfer function, respectively. The steady–state system gain factor of the digital filter $H(z)$ is given by

$$\left|H(e^{j\omega T})\right| = \frac{|b_0| \cdot |e^{j\omega T} - z_1| \cdot |e^{j\omega T} - z_2| \cdots |e^{j\omega T} - z_k|}{|e^{j\omega T} - p_1| \cdot |e^{j\omega T} - p_2| \cdots |e^{j\omega T} - p_\ell|}. \tag{2.6}$$

As shown in (2.6), the system gain at frequency ω depends on the distances from the system transfer function's zeros and poles to the point $e^{j\omega T}$ which is a vector of unit length and angle ωT. Thus, if we wish to completely remove the sinusoid $\sin(k\omega_0 T)$, it is required that the transfer function should have a zero at $z = e^{j\omega_0 T}$. Note that the transfer function must have zeros at $z = e^{j\omega_0 T}$ and $z = e^{-j\omega_0 T}$ for the difference equation (2.3) to have real coefficients.

2.4 Location of the Poles and Zeros of a Notch Filter

It is apparent from the previous section that when the transfer function of a digital filter has zeros located at $e^{j\omega_0 T}$ and $e^{-j\omega_0 T}$, the filter having real coefficients will have a notch at ω_0. In addition, one more

requirement of an ideal notch filter is that the value of its frequency response is near unity for frequencies sufficiently close to the notch frequency ω_0. To satisfy this requirement, we must properly locate poles that tend to cancel the effects of the zeros. Although it is desired for a notch filter that the poles be located as close as possible to the zeros at ω_0 and the unit circle, we have to take into account practical design considerations. Since all digital systems are based on finite word lengths, the desired filter coefficients according to the ideal location of the zeros and poles will be truncated and thus the resulting truncated filter coefficients may lead to an unstable notch filter. Therefore, it is important to decide the location of the zeros and poles so that the resulting digital notch filter not only has the frequency response close to that of the ideal notch filter but also becomes a stable filter.

2.5 Design of Digital Notch Filters

Various design methods to obtain a stable digital notch filter with suitable notch characteristics have been considered quite extensively in the literature by Abu-El-Haija and Perterson [1979], Antoniou [1983], Cadzow [1973, 1974], Hirano, Nishimura, and Mitra [1974], Mitra and Sherwood [1973], and Yan [1984]. In this section, we describe the two most popular design methods that yield a stable digital notch filter.

2.5.1 Second-Order Digital Notch Filter

The transfer function of an analog notch filter is given as

$$H(s) = \frac{s^2 + \lambda^2}{s^2 + bs + \lambda^2} \tag{2.7}$$

where λ and b denote, respectively, a notch frequency and the 3-dB rejection bandwidth: i.e., within the frequency band of width b centered

at λ, all signals are attenuated by more than 3 dB. Note that the DC gain is 0 dB (see Fig. 2.2). To develop a transfer function for the digital notch filter, we apply the bilinear transformation [Lam, 1979; Oppenheim and Schafer, 1975]

$$s = \frac{z-1}{z+1} \tag{2.8}$$

to $H(s)$. This yields

$$\begin{aligned} H(z) &= H(s)\Big|_{s=\frac{z-1}{z+1}} && \text{(2.9a)} \\ &= \frac{(1+\lambda^2) - 2(1-\lambda^2)z^{-1} + (1+\lambda^2)z^{-2}}{(1+\lambda^2+b) - 2(1-\lambda^2)z^{-1} + (1+\lambda^2-b)z^{-2}} && \text{(2.9b)} \\ &= \frac{1}{2} \cdot \frac{(1+a_2) - 2a_1 z^{-1} + (1+a_2)z^{-2}}{1 - a_1 z^{-1} + a_2 z^{-2}} && \text{(2.9c)} \end{aligned}$$

where

$$\begin{aligned} a_1 &= \frac{2(1-\lambda^2)}{1+\lambda^2+b} && \text{(2.10a)} \\ a_2 &= \frac{1+\lambda^2-b}{1+\lambda^2+b}. && \text{(2.10b)} \end{aligned}$$

Neglecting the constant 1/2, which is equivalent to that of adding a flat 6 dB gain to the frequency response, we rewrite (2.9c) in the desired form as

$$H(z) = \frac{(z^{-2}+1) - 2a_1 z^{-1} + a_2(z^{-2}+1)}{1 - a_1 z^{-1} + a_2 z^{-2}}. \tag{2.11}$$

Note that this notch filter transfer function is characterized by two distinct parameters a_1 and a_2. To get the relationships among the constants a_1 and a_2, the notch frequency ω_0, and the 3–dB rejection bandwidth Ω, we need to find the relationships among the parameters λ, b, ω_0, and Ω. Let us first consider the magnitude of transfer function $H(z)$ described in (2.11), then it becomes

$$\left|H(e^{j\omega T})\right|^2 = \frac{\{(1+\lambda^2)\cos(\omega T) - (1-\lambda^2)\}^2}{\{(1+\lambda^2)\cos(\omega T) - (1-\lambda^2)\}^2 + b^2 \sin^2(\omega T)}. \tag{2.12}$$

By setting the numerator of (2.12) equal to zero, the notch frequency ω_0 is given as

$$\cos(\omega_0 T) = \frac{1-\lambda^2}{1+\lambda^2}. \tag{2.13}$$

For the frequencies (ω_1 and ω_2) where the magnitude is down 3 dB from its DC value, we solve

$$\left\{\left(1+\lambda^2\right)\cos(\omega T) - \left(1-\lambda^2\right)\right\}^2 = b^2\sin^2(\omega T). \tag{2.14}$$

That yields

$$\cos(\Omega T) = \cos(\omega_2 - \omega_1)T = \frac{(1+\lambda^2)^2 - b^2}{(1+\lambda^2)^2 + b^2}. \tag{2.15}$$

After solving (2.13) and (2.15) for λ^2 and b, we can easily get the following relationships:

$$\lambda^2 = \tan^2(\omega_0 T/2) \tag{2.16a}$$

$$b = \left(1+\lambda^2\right)\tan(\Omega T/2). \tag{2.16b}$$

Using these relationships and the expressions of a_1 and a_2 given by (2.10a) and (2.10b), we can obtain the following relationship among the parameters a_1, a_2, ω_0, and Ω [Hirano *et al.*, 1974]:

$$a_1 = \frac{2\cos(\omega_0 T)}{1+\tan(\Omega T/2)} \tag{2.17a}$$

$$a_2 = \frac{1-\tan(\Omega T/2)}{1+\tan(\Omega T/2)}. \tag{2.17b}$$

It is observed that the notch frequency can be changed while keeping the 3–dB rejection bandwidth and DC gain constants by varying a_1; and that the rejection bandwidth can be changed by varying only a_2. Various realizations of the transfer function of (2.11) have been described by Hirano, Nishimura, and Mitra [1974] by taking into account multiplication roundoff errors and hardware requirements.

Example 2.1

Assuming that the sampling frequency is $1/T = 10$ kHz, let us design two second–order digital notch filters having a notch at 1,000 Hz having different 3–dB rejection bandwidths: 100 Hz and 250 Hz. Note that the normalized notch frequency is 0.1.

3–dB rejection bandwidth: 100 Hz

Assuming the desired 3–dB rejection bandwidth of 100 Hz and using the specifications of the given problem, we obtain

$$\omega_0 T = 2\pi \times 1000 \times \frac{1}{10000} = 0.2\pi \text{ (notch frequency)}$$
$$\Omega T = 2\pi \times 100 \times \frac{1}{10000} = 0.02\pi \text{ (3–dB rejection bandwidth).}$$

Then, using (2.17a) and (2.17b), we obtain the parameters, a_1 and a_2 as

$$a_1 = \frac{2\cos(0.2\pi)}{1+\tan(0.01\pi)} = 1.56873452036162$$
$$a_2 = \frac{1-\tan(0.01\pi)}{1+\tan(0.01\pi)} = 0.93906250581749.$$

Hence the notch filter transfer function with 3–dB rejection bandwidth of 100 Hz, $H_1^{(2)}(z)$, becomes

$$H_1^{(2)}(z) = \frac{(z^{-2}+1) - 3.13746904072324\, z^{-1} + 0.93906250581749\,(z^{-2}+1)}{1 - 1.56873452036162\, z^{-1} + 0.93906250581749\, z^{-2}}.$$

Its magnitude response shown in Fig. 2.3 (a) illustrates a notch at the normalized frequency 0.1.

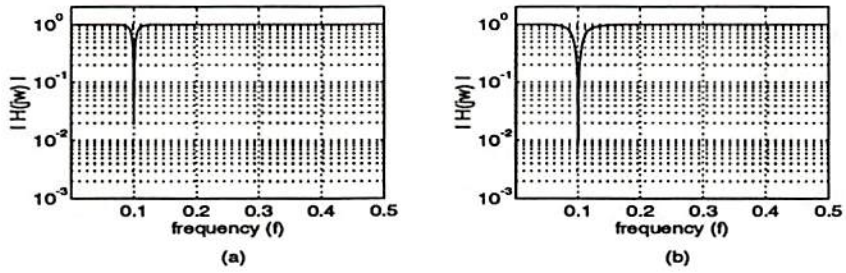

Figure 2.3: Magnitude response of the transfer functions of the second–order digital notch filters using the bilinear transformation with (a) 100 Hz rejection bandwidth and (b) 250 Hz rejection bandwidth.

3–dB rejection bandwidth: 250 Hz

When the desired 3–dB rejection bandwidth is 250 Hz, using the problem specifications, we obtain

$$\begin{aligned} \omega_0 T &= 2\pi \times 1000 \times \frac{1}{10000} = 0.2\pi \\ \Omega T &= 2\pi \times 250 \times \frac{1}{10000} = 0.05\pi. \end{aligned}$$

and the parameters, a_1 and a_2 as

$$\begin{aligned} a_1 &= \frac{2\cos(0.2\pi)}{1+\tan(0.025\pi)} = 1.49998278348230 \\ a_2 &= \frac{1-\tan(0.025\pi)}{1+\tan(0.025\pi)} = 0.85408068546347. \end{aligned}$$

Hence the notch filter transfer function with 3–dB rejection bandwidth of 250 Hz, $H_2^{(2)}(z)$, becomes

$$H_2^{(2)}(z) = \frac{(z^{-2}+1) - 2.99996556696459\, z^{-1} + 0.85408068546347\,(z^{-2}+1)}{1 - 1.49998278348230\, z^{-1} + 0.85408068546347\, z^{-2}}.$$

The magnitude response of the transfer function is shown in Fig. 2.3 (b). From Fig. 2.3 (a) and (b) we observe that the transfer function $H_2^{(2)}(z)$

has a wider 3–dB rejection bandwidth, but a deeper notch than $H_1^{(2)}(z)$.

Example 2.2

Let us consider the following six sinusoidal signals with different frequencies

$$x_i(k) = \sqrt{2}\,\sin(2\pi f_i\, kT + \pi/3) \quad \text{for } k = 0, 1, 2, \ldots, K-1$$

where $K = 200$, $T = 1/10,000$ is the sampling interval, and f_i for $i = 1, 2, 3, 4, 5, 6$, are different frequencies given by

$$f_i = 1000 + (i-1) \cdot \Delta f$$

with $\Delta f = 5$ Hz. Each sinusoidal signal is assumed to be the input to a second–order digital notch filter with either 100 Hz or 250 Hz as 3–dB rejection bandwidths (designs obtained in Example 2.1).

Let $\{y_1^{(2)}(k)\}$ and $\{y_2^{(2)}(k)\}$ be the corresponding outputs obtained by the previously designed digital notch filters $H_1^{(2)}(z)$ and $H_2^{(2)}(z)$, respectively. Each input–output pair using the two designed second–order digital notch filters is illustrated in Fig. 2.4. Although the sinusoidal input with the exact notch frequency is almost completely eliminated, the output obtained by the digital notch filter $H_2^{(2)}(z)$ with 250 Hz rejection bandwidth vanishes faster than that obtained by the filter $H_1^{(2)}(z)$ with 100 Hz rejection bandwidth. As the difference between the input signal frequency and the notch frequency increases, both filters allow more signal to pass through. Since the transfer function $H_2^{(2)}(z)$ has deeper notch and wider rejection bandwidth than $H_1^{(2)}(z)$, the transfer function $H_2^{(2)}(z)$ mitigates more frequency components close to the notch frequency. That means that a digital notch filter with a wide rejection

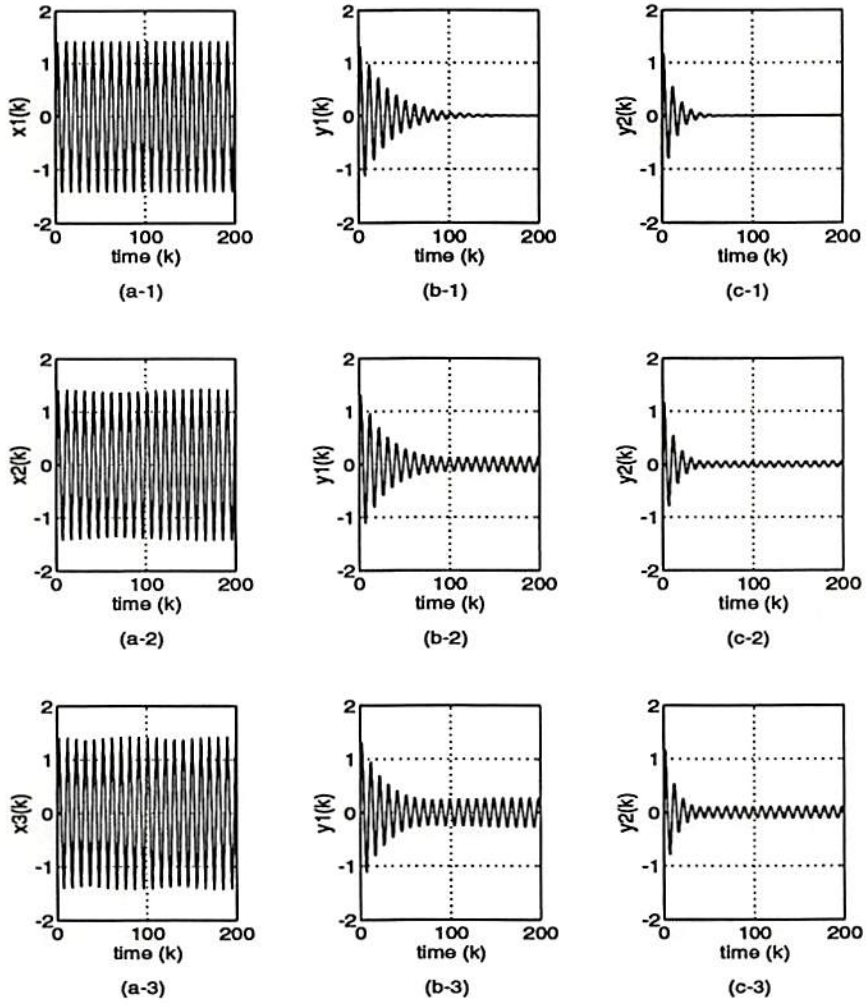

Figure 2.4: Six different sinusoidal inputs and corresponding outputs obtained by $H_1^{(2)}(z)$ and $H_2^{(2)}(z)$. The first column shows the sinusoidal input $\{x(k)\}$ whose frequency is (a–1) 1,000 Hz, (a–2) 1,005 Hz, and (a–3) 1,010 Hz; the second and third columns show the corresponding outputs $\{y_1^{(2)}(k)\}$ and $\{y_2^{(2)}(k)\}$, respectively.

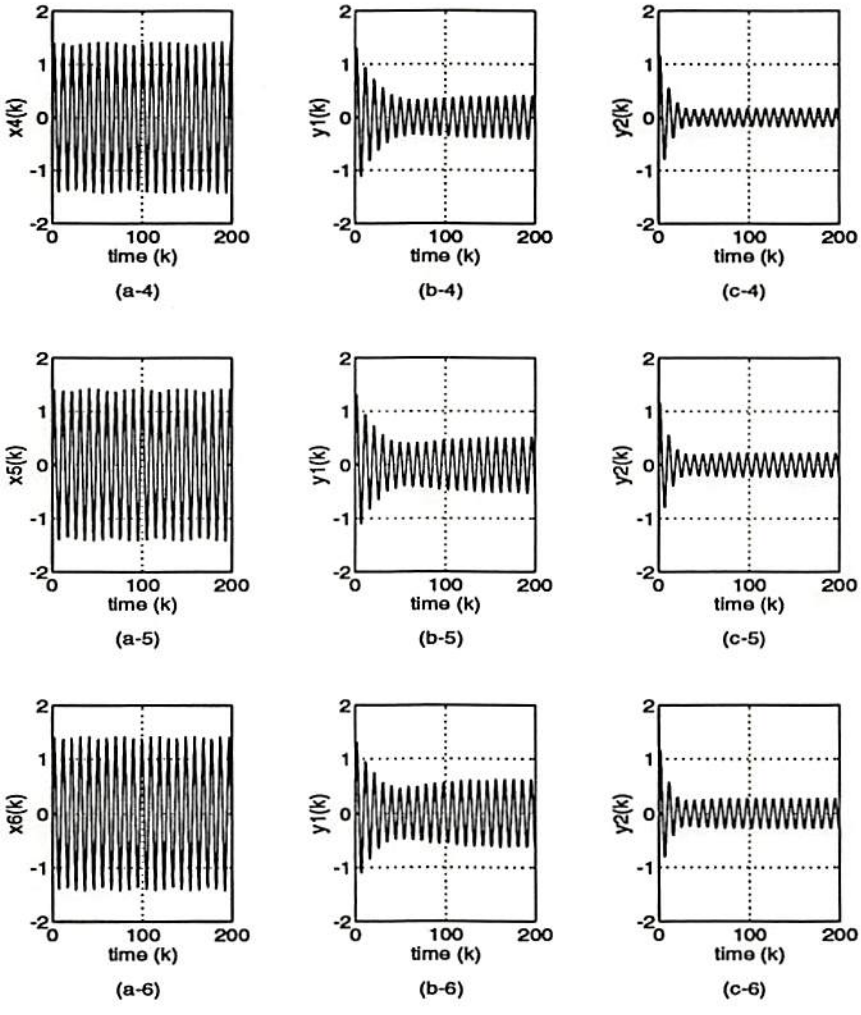

Figure 2.4: (Continued) Six different sinusoidal inputs and corresponding outputs obtained by $H_1^{(2)}(z)$ and $H_2^{(2)}(z)$. The first column shows the sinusoidal input $\{x(k)\}$ whose frequency is (a–4) 1,015 Hz, (a–5) 1,020 Hz, and (a–6) 1,025 Hz; the second and third columns show the corresponding outputs $\{y_1^{(2)}(k)\}$ and $\{y_2^{(2)}(k)\}$, respectively.

bandwidth removes not only the sinusoidal interference, which we want to mitigate, but also other valuable signal components whose frequencies are near the notch frequency.

2.5.2 Sixth–Order Digital Notch Filter

Let us consider a sixth–order notch filter with notch at ω_0. To avoid the possibility of an unstable digital notch filter due to finite word lengths, the required sixth–order notch filter is factored by cascading three second–order digital filters whose individual transfer functions are given by [Cadzow, 1974]

$$H_i(z) = \frac{(z - e^{j\omega_0 T})(z - e^{-j\omega_0 T})}{(z - p_i)(z - p_i^*)} \qquad \text{for} \quad i = 1, 2, 3 \tag{2.18}$$

where the poles, p_i, are located as shown in Fig. 2.5. Using the geometrical pole representation, the pole locations are described by

$$\begin{aligned} p_1 &= e^{j\omega_0 T} - \Delta\, e^{j(\omega_0 T - \theta_1)} \\ p_2 &= (1 - \Delta) e^{j\omega_0 T} \\ p_3 &= e^{j\omega_0 T} - \Delta\, e^{j(\omega_0 T + \theta_2)}. \end{aligned} \tag{2.19}$$

The relationship between the input and output of the i^{th} filter is

$$\begin{aligned} y_i(k) &= u_i(k) - 2\cos(\omega_0 T)\, u_i(k-1) + u_i(k-2) \\ &\quad + (p_i + p_i^*)\, y_i(k-1) - p_i p_i^*\, y_i(k-2) \end{aligned} \tag{2.20}$$

where $\{u_i(k)\}$ and $\{y_i(k)\}$ denote the input and output of the i^{th} filter, respectively, for $i = 1, 2, 3$ (see Fig 2.6). The coefficients of the difference equations for $i = 1, 2, 3$ are determined by

$$p_i + p_i^* = 2\left[\cos(\omega_0 T) - \Delta\, \cos(\omega_0 T + \phi_i)\right] \tag{2.21a}$$

$$p_i \cdot p_i^* = 1 + \Delta^2 - 2\Delta\, \cos\phi_i \tag{2.21b}$$

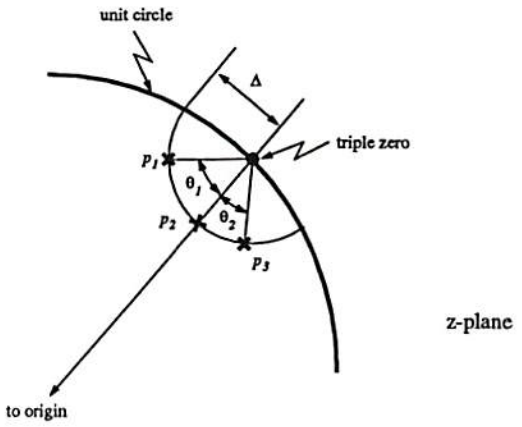

Figure 2.5: Pole and zero locations of the transfer function of a sixth–order digital notch filter.

where $\phi_1 = -\theta_1$, $\phi_2 = 0$, and $\phi_3 = \theta_2$.

When we implement the difference equation (2.20), the coefficients $2\cos(\omega_0 T)$, $p_i + p_i^*$, and $p_i p_i^*$ will be stored with corresponding truncation errors. When the error due to truncation is ϵ_0, the coefficient $2\cos(\omega_0 T)$ is stored as

$$2\cos(\omega_0 T) + \epsilon_0. \tag{2.22}$$

This truncation error causes a shift in the desired zero locations. Let us assume that instead of the ideal zeros at $e^{j\omega_0 T}$ and $e^{-j\omega_0 T}$, we have the

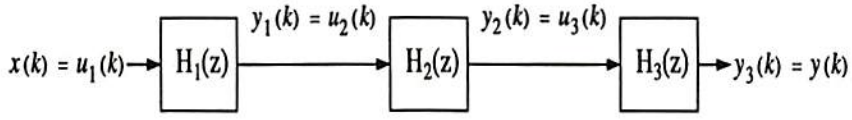

Figure 2.6: The cascade realization of a sixth–order digital notch filter using three second–order digital filters.

zeros located at $e^{j(\omega_0 T+\delta)}$ and $e^{-j(\omega_0 T+\delta)}$ due to ϵ_0, i.e.,

$$2\cos(\omega_0 T+\delta) = 2\cos(\omega_0 T) + \epsilon_0. \tag{2.23}$$

When both the errors, ϵ_0 and δ, are very small, the approximation for δ can be obtained [Cadzow, 1974]

$$\delta \approx -\frac{\epsilon_0}{2\sin(\omega_0 T)}. \tag{2.24}$$

Since the magnitude of this rotation must be significantly smaller than Δ, a design criterion for Δ is

$$\Delta \gg |\,\delta\,| = \left|\frac{\epsilon_0}{2\sin(\omega_0 T)}\right|. \tag{2.25}$$

The parameters Δ, θ_1 and θ_2 can be selected to ensure that the truncation errors of the coefficients $p_i p_i^*$ for $i = 1,2,3$ are zero. Note that the remaining coefficients $p_i + p_i^*$ for $i = 1,2,3$ specifying the poles also have truncation errors and one may similarly argue that the deviation from the ideal pole locations becomes a rotation of $-\epsilon_i/[2\sin(\omega_0 T)]$ where ϵ_i is the corresponding truncation error of coefficient $p_i + p_i^*$. Then the gain factor at the ideal notch frequency ω_0 for the actual filter becomes approximately

$$\text{gain at } \omega_0 = \left|\frac{\epsilon_0}{2\Delta\sin(\omega_0 T)}\right|^3. \tag{2.26}$$

It is important to note that the parameter Δ specifies both the width of the filter's notch and the speed of response characteristic. As Δ becomes smaller, the width of the notch decreases while the filter's response time increases because the poles of the notch filter move closer to the unit circle.

Example 2.3

Considering the same problem specifications as in Example 2.1, we'll design four sixth–order notch digital filters using different values of the design parameter Δ. Note that we assume $\theta_1 = \theta_2 = \frac{2}{5}\pi$ for all the following design cases. Let us first consider $\Delta = 0.0075\,\pi$. Then, using the specifications of the given problem, (2.21a), and (2.21b), we obtain

$$2\cos(\omega_0 T) = 1.61803398874989$$

$$\begin{aligned} p_1 + p_1^* &= 1.57990996105753 & p_1 \cdot p_1^* &= 0.98599308245712 \\ p_2 + p_2^* &= 1.57990996105753 & p_2 \cdot p_2^* &= 0.95343127544371 \\ p_3 + p_3^* &= 1.63259607154034 & p_3 \cdot p_3^* &= 0.98599308245712 \end{aligned}$$

where $\omega_0 T = 0.2\pi$. When $\{x(k)\}$ is the input to the sixth–order digital notch filter, the difference equations given by (2.20) become

$$\begin{aligned} y_1(k) &= x(k) - b_1 x(k-1) + x(k-2) + a_1 y_1(k-1) - a_2 y_1(k-2) \\ y_2(k) &= y_1(k) - b_1 y_1(k-1) + y_1(k-2) + a_3 y_2(k-1) - a_4 y_2(k-2) \\ y_3(k) &= y_2(k) - b_1 y_2(k-1) + y_2(k-2) + a_5 y_3(k-1) - a_6 y_3(k-2) \end{aligned}$$

where $y_3(k)$ is the final filter output and the coefficients b_1 and a_i's are given by

$$b_1 = 1.61803398874989$$

$$\begin{aligned} a_1 &= 1.57990996105753 & a_2 &= 0.98599308245712 \\ a_3 &= 1.57990996105753 & a_4 &= 0.95343127544371 \\ a_5 &= 1.63259607154034 & a_6 &= 0.98599308245712. \end{aligned}$$

The magnitude response of the overall transfer function of the sixth–order digital notch filter is shown in Fig. 2.7 (a). When the values of the parameter Δ are 0.009, 0.0125, and 0.02, we can obtain the required parameters through the same procedure. When $\Delta = 0.009$, the coefficients

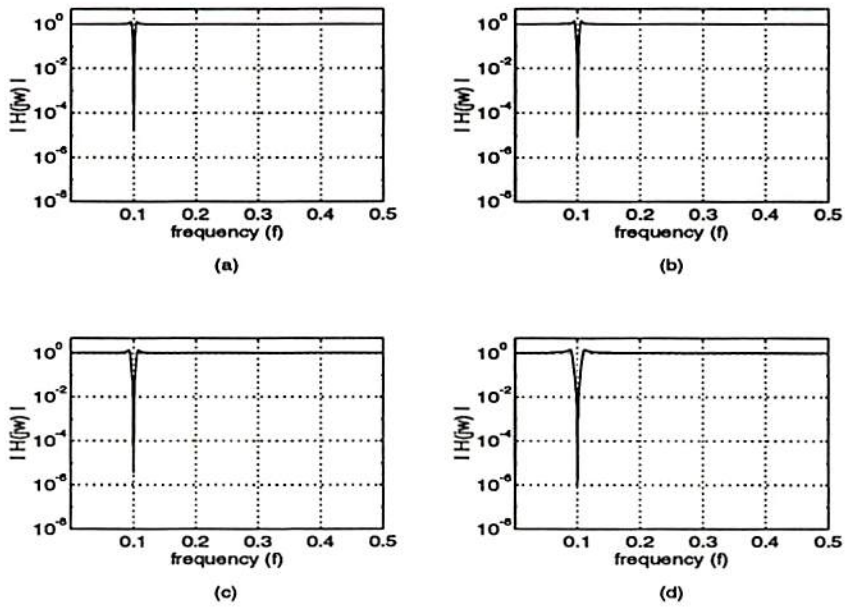

Figure 2.7: Magnitude response of the transfer functions $H^{(6)}(z)$ of the sixth–order digital notch filters with (a) $\Delta = 0.0075$, (b) $\Delta = 0.009$, (c) $\Delta = 0.0125$, and (d) $\Delta = 0.02$.

a_i's are given by

$$a_1 = 1.57228515551906 \qquad a_2 = 0.98332493860796$$
$$a_3 = 1.57228515551906 \qquad a_4 = 0.94425077019187$$
$$a_5 = 1.63550848809842 \qquad a_6 = 0.98332493860796,$$

when $\Delta = 0.0125$, the coefficients a_i's are given by

$$a_1 = 1.55449394259595 \qquad a_2 = 0.97727198770360$$
$$a_3 = 1.55449394259595 \qquad a_4 = 0.92300230934793$$
$$a_5 = 1.64230412673396 \qquad a_6 = 0.97727198770360,$$

and when $\Delta = 0.02$, the coefficients a_i's are given by

$$a_1 = 1.51636991490359 \qquad a_2 = 0.96511562098593$$
$$a_3 = 1.51636991490359 \qquad a_4 = 0.87828413561684$$
$$a_5 = 1.65686620952440 \qquad a_6 = 0.96511562098593.$$

Note that $b_1 = 1.61803398874989$ in all the design cases because it depends on the notch frequency only. Their corresponding magnitude response of transfer functions are also shown in Fig. 2.7. As we can see from this figure, as the value of Δ increases, the depth of the notch increases and the rejection bandwidth increases. With a very small value of Δ we can obtain a digital notch filter whose transfer function has the gain–frequency behavior of an ideal notch filter.

Example 2.4

Let us compare the performance of the three sixth–order digital notch filters designed in the previous example: $H_2^{(6)}(z)$ with $\Delta = 0.009$, $H_3^{(6)}(z)$ with $\Delta = 0.0125$, and $H_4^{(6)}(z)$ with $\Delta = 0.02$, using the six sinusoidal input signals with different frequencies given in Example 2.2.

Let $\{y_2^{(6)}(k)\}$, $\{y_3^{(6)}(k)\}$, and $\{y_4^{(6)}(k)\}$ be the corresponding outputs obtained by the designed digital sixth–order notch filters $H_2^{(6)}(z)$, $H_3^{(6)}(z)$, and $H_4^{(6)}(z)$, respectively. Each output obtained by the three sixth–order digital notch filters is illustrated in Fig. 2.8. Note that the six different sinusoidal inputs are shown in Fig. 2.4. Since the transfer function $H_2^{(6)}(z)$ has the narrowest rejection bandwidth, it passes more input signal components than the other transfer functions as the bias between the notch frequency and the input frequency increases; however, its output produces quite large oscillations that vanish slowly. As the value of Δ increases, these oscillations vanish faster. For fast mitigation of the interference, it is required to design the sixth–order digital notch filter using a large value of Δ at the expense of a wider rejection bandwidth which may cause larger distortion on different signal components from the interference which we want to mitigate.

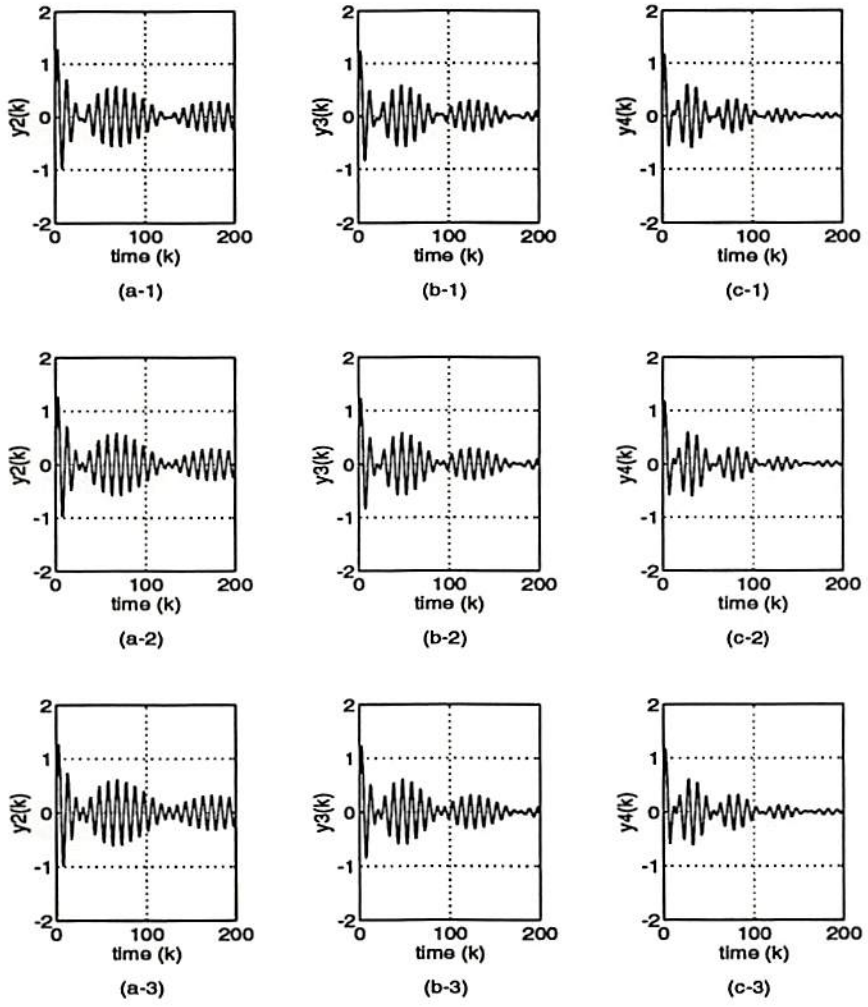

Figure 2.8: Using the six different sinusoidal inputs, the corresponding outputs obtained by $H_2^{(6)}(z)$, $H_3^{(6)}(z)$, and $H_4^{(6)}(z)$. The first column show the outputs $\{y_2^{(6)}(k)\}$ when the frequency of the sinusoidal input is (a–1) 1,000 Hz, (a–2) 1,005 Hz, and (a–3) 1,010 Hz; the second and third columns show the corresponding outputs $\{y_3^{(6)}(k)\}$ and $\{y_4^{(6)}(k)\}$, respectively.

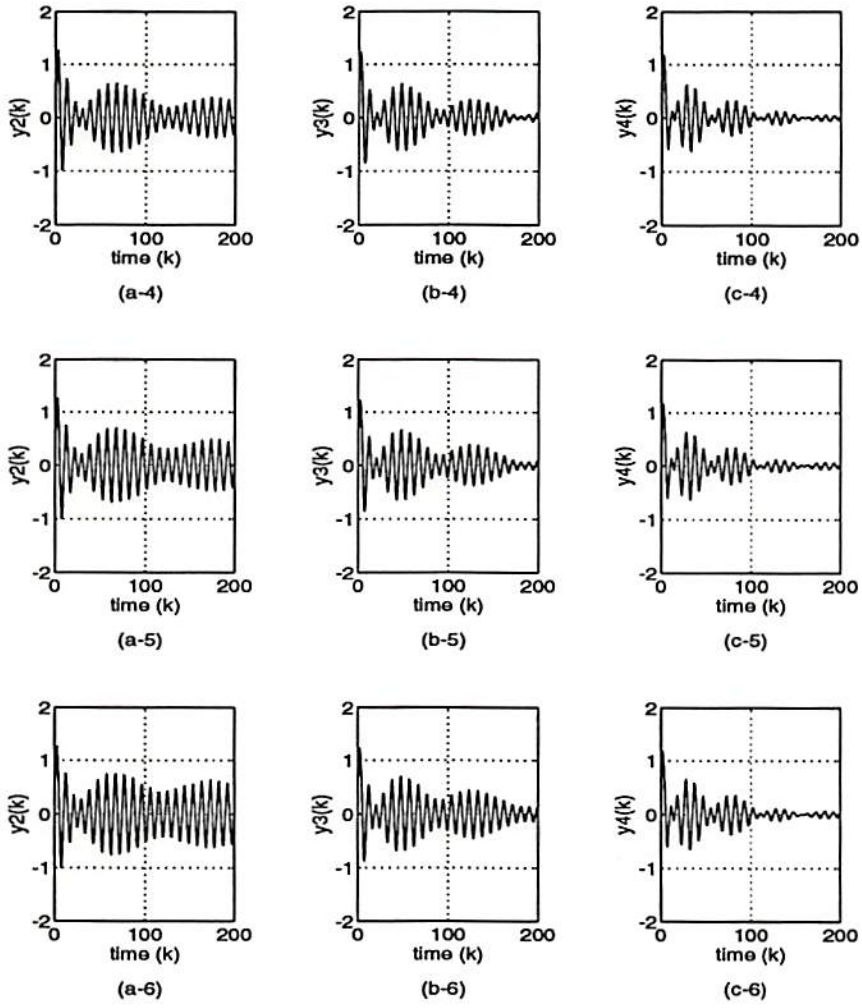

Figure 2.8: (Continued) Using the six different sinusoidal inputs, the corresponding outputs obtained by $H_2^{(6)}(z)$, $H_3^{(6)}(z)$, and $H_4^{(6)}(z)$. The first column show the outputs $\{y_2^{(6)}(k)\}$ when the frequency of the sinusoidal input is (a–4) 1,015 Hz, (a–5) 1,020 Hz, and (a–6) 1,025 Hz; the second and third columns show the corresponding outputs $\{y_3^{(6)}(k)\}$ and $\{y_4^{(6)}(k)\}$, respectively.

Example 2.5

Assume that the SOI $\{s(k)\}$ is a BPSK signal satisfying $\Pr\{s_0 = -1\} = \Pr\{s_1 = 1\} = 0.5$ and that the sinusoidal interference frequency is 1,000 Hz. Then, the input $\{x(k)\}$ can be represented as the sum of the SOI and the interference signal such as

$$x(k) = s(k) + I(k)$$

where

$$I(k) = 5\,\sin(2\pi f_I\, kT + \pi/6)$$

with $f_I = 1,000$ Hz and $T = 1/10,000$. The SOI $\{s(k)\}$ and the input sequence $\{x(k)\}$ are illustrated in Fig. 2.9.

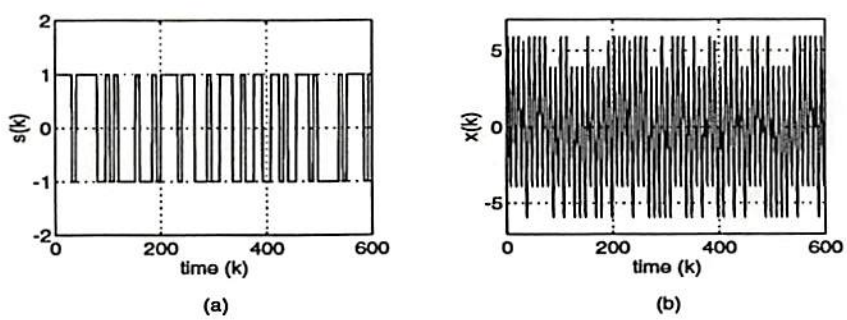

Figure 2.9: (a) Interference–free BPSK SOI $\{s(k)\}$ and (b) input sequence $\{x(k)\}$ consisting of the SOI and the additive interference.

To mitigate the sinusoidal interference using a digital notch filter, let us consider the second–order digital notch filters $H_1^{(2)}(z)$ with 3–dB rejection bandwidth of 100 Hz and $H_2^{(2)}(z)$ with the 3–dB rejection bandwidth of 250 Hz which are designed in Example 2.1. Then, their corresponding outputs with $\{x(k)\}$ as input are shown in Fig. 2.10 (a) and (b), respectively. Comparing Fig. 2.10 (a) and (b), it is apparent that the

output obtained by a digital notch filter with the deeper notch and wider rejection bandwidth shows faster interference–mitigation performance at the expense of a larger distortion on the BPSK signal of interest.

Now, to mitigate the interference let us also use the sixth–order digital notch filters $H_1^{(6)}(z)$ with $\Delta = 0.0075$ and $H_3^{(6)}(z)$ with $\Delta = 0.0125$ which are designed in Example 2.3. Their corresponding outputs with $\{x(k)\}$ as input sequence are shown in Fig. 2.11 (a) and (b), respectively.

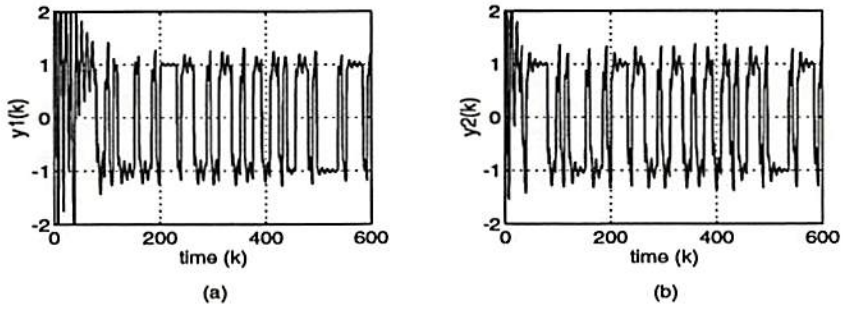

Figure 2.10: With the BPSK SOI and the sinusoidal interference, the output sequences obtained by the second–order digital notch filters (a) $H_1^{(2)}(z)$ with 100 Hz rejection bandwidth and (a–2) $H_2^{(2)}(z)$ with 250 Hz rejection bandwidth.

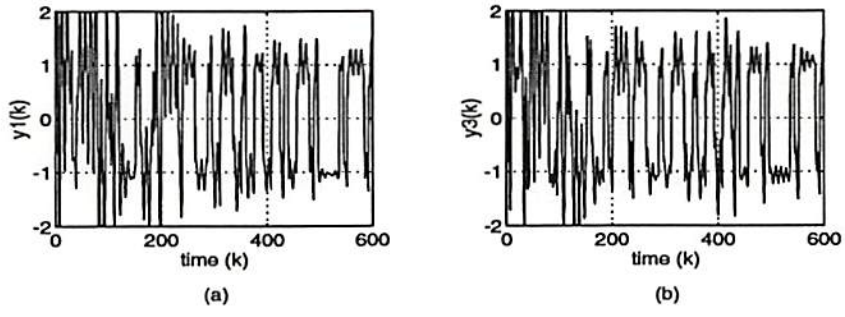

Figure 2.11: With the BPSK SOI and the sinusoidal interference, the output sequences obtained by the sixth–order digital notch filters (a) $H_1^{(6)}(z)$ with $\Delta = 0.0075$ and (b) $H_3^{(6)}(z)$ with $\Delta = 0.0125$.

Again, we can observe the similar phenomena which is that the output obtained by a digital notch filter with the deeper notch and wider rejection bandwidth shows faster interference–mitigation performance at the expense of a larger distortion on the SOI.

Example 2.6

Let us assume that the sinusoidal interference has a slowly time–varying frequency. The purpose of the example is to demonstrate the performance of the second–order and sixth–order digital notch filters in the case of a nonstationary interference. Let us assume that the input $\{x(k)\}$ having time–varying frequency is given by

$$x(k) = \sqrt{2}\,\sin(2\pi f_x(k)\,kT + \pi/3)$$

where

$$f_x(k) = 1000 + 10\,\sin(0.0175\,k + \pi/12).$$

This particular frequency modulating function and the corresponding input sequence are illustrated Fig. 2.12 (a) and (b), respectively.

Let us consider the second–order digital notch filter $H_2^{(2)}(z)$ with 3–dB rejection bandwidth of 250 Hz and the sixth–order notch filter $H_4^{(6)}(z)$ with $\Delta = 0.02$ designed in the previous examples. When we apply the input sequence $\{x(k)\}$ with the time–varying frequency to these two digital notch filters, their corresponding outputs are illustrated in Fig. 2.13.

Let us compare the performance of the second–order digital notch filter $H_2^{(2)}(z)$ shown in Fig. 2.13 to that shown in Fig. 2.4. When the interference is stationary, the maximum value of the outputs in Fig. 2.4 is much less than unity even when the bias between the filter's notch

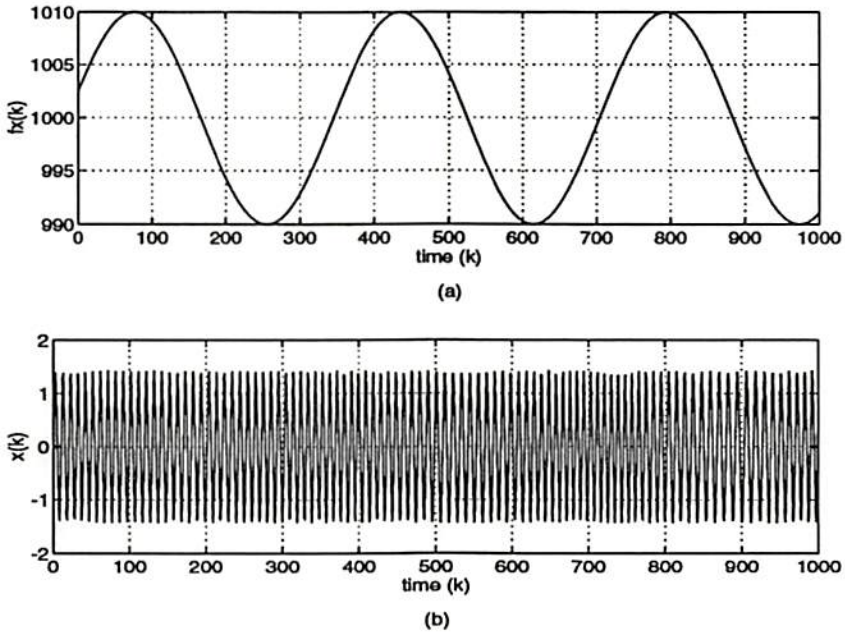

Figure 2.12: (a) Time–varying frequency of the sinusoidal input and (b) input sequence.

frequency and the interference frequency is as large as 25 Hz. On the other hand, when the interference is nonstationary, the maximum value of the output in Fig. 2.13 is larger than unity. Note that the maximum frequency bias is 10 Hz in the nonstationary interference case. When we compare the performance of the sixth–order digital notch filter $H_4^{(6)}(z)$ shown in Fig. 2.13 to that shown in Fig. 2.8, it is apparent that the digital notch filter $H_4^{(6)}(z)$ can mitigate a stationary sinusoidal component more effectively than a nonstationary one. Note that some values of the output obtained by $H_4^{(6)}(z)$ are larger than the maximum value of the input which is $\sqrt{2}$ in Fig. 2.13. This example illustrates that we can not use a digital notch filter with a fixed notch, when the interference frequency is slowly time–varying or non–stationary.

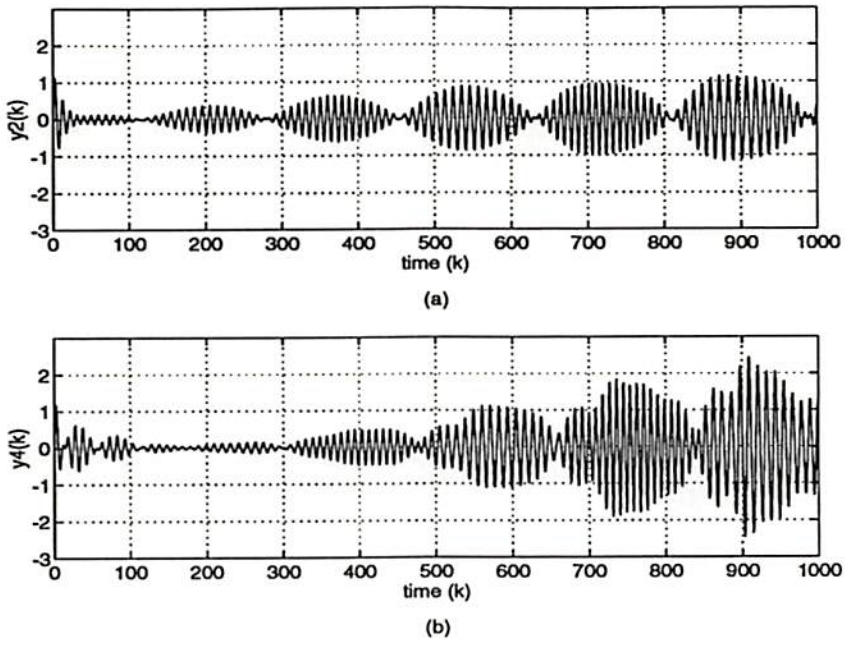

Figure 2.13: When the input is a sinusoidal signal whose frequency is slowly time–varying, the erformance of the digital notch filters (a) $H_2^{(2)}(z)$ with 3–dB rejection bandwidth of 250 Hz and (b) $H_4^{(6)}(z)$ with $\Delta = 0.02$.

2.6 Summary

In this chapter, we introduced the two most popular design techniques for digital notch filters based on either second– or sixth–order transfer functions. We demonstrated that the performance of a designed notch filter depends on many parameters: desired notch depth, desired 3 dB rejection bandwidth, finite word length and roundoff errors, and realization structure. The major disadvantage of fixed digital notch filters is that their performance is very poor when the characteristics of the interference is not stationary, i.e., time–varying. Consequently, the use of fixed digital notch filters is very limited due to the fact that it is not always

possible in practice to know in advance or estimate the time–varying characteristics of the interference. These limitations naturally lead us to develop and use adaptive digital notch filters that can be applied on both stationary and non–stationary environments.

Chapter 3

Adaptive Notch Filters

3.1 Introduction

When the characteristics of narrowband interference are unknown or nonstationary, we cannot use a conventional fixed notch filter described in Chapter 2 to cancel the interference. In such situations, an adaptive filter design technique should be employed. In this chapter, we are concerned primarily with the description, performance assessment, and computational complexity of adaptive interference mitigation (AIM) filter algorithms based on different realization structures and constraints.

3.2 Preliminaries

We can divide the AIM filter algorithms into two categories. The *first category* consists of adaptive AIM algorithms using two inputs: primary and reference inputs shown in Fig. 3.1. The primary input, $\{x(k)\}$, receives the signal of interest (SOI) and the interference. The reference input, $\{z(k)\}$, receives a signal which is highly correlated with the inter-

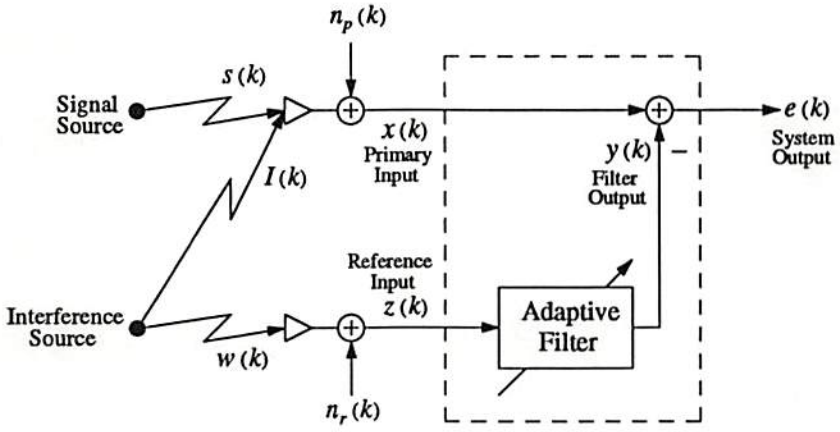

Figure 3.1: Block diagram of the adaptive interference mitigation filter with the reference signal belonging to the first category.

ference only, viz:

$$x(k) = s(k) + I(k) + n_p(k) \tag{3.1a}$$

$$z(k) = w(k) + n_r(k) \tag{3.1b}$$

where $\{s(k)\}$ and $\{I(k)\}$ denote the SOI and the interference, The reference signal $\{w(k)\}$ should be uncorrelated with the SOI. We assume that the characteristics of the interference source are unknown or non-stationary; or that the relationship between the reference signal and the interference is assumed to be partially known, fully unknown, or time–varying. The uncorrelated noise sources $\{n_p(k)\}$ and $\{n_r(k)\}$ are assumed to be uncorrelated with the SOI, the interference, the reference signal, and with each other. It is important to note that the uncorrelated noise sources could not be removed. Since our major concern is to mitigate the strong interference, we would assume that the powers of the uncorrelated noise sources are negligible or very small. The filter in the first category will be adaptively designed to be able to change the reference

input into the best fit of the interference. The filter output is then subtracted from the primary input and the system output would be the SOI only. The AIM filter using the reference input usually has a finite impulse response (FIR). The block diagram of an AIM problem belonging to the first category is illustrated in Fig. 3.1.

In the *second category*, the reference signal is not available or not accessible. What we have is only the primary input $\{x(k)\}$ which is the SOI corrupted by the interference. To mitigate the interference, we have to use the characteristics of both the SOI and the interference and adaptively adjust the filter coefficients. The AIM filter belonging to the second category usually has an infinite impulse response (IIR).

3.3 Adaptive LMS Filter as a Notch Filter

When the primary input consists of the SOI with an additive undesired sinusoidal interference, an AIM filter can be realized by an adaptive noise canceler using the least mean squares (LMS) algorithm [Widrow *et al.*, 1975; Widrow *et al.*, 1976; Widrow and Stearns, 1985]. In LMS–based noise canceling systems, the system output is fed back to the adaptive filter and the filter is adjusted through an adaptive algorithm to minimize the total system output power. Note that the LMS–based AIM filter belongs to the first category.

When the reference input is a single sinusoidal wave $C\cos(\Omega_0 t + \phi)$ and the primary and reference inputs are sampled at intervals of T seconds, we can use a two–weight notch filter based on the LMS algorithm and the delay between the two reference input components is equivalent to a 90° phase shift at the reference frequency shown in Fig. 3.2. Then, the transfer function from the primary input to the system output can

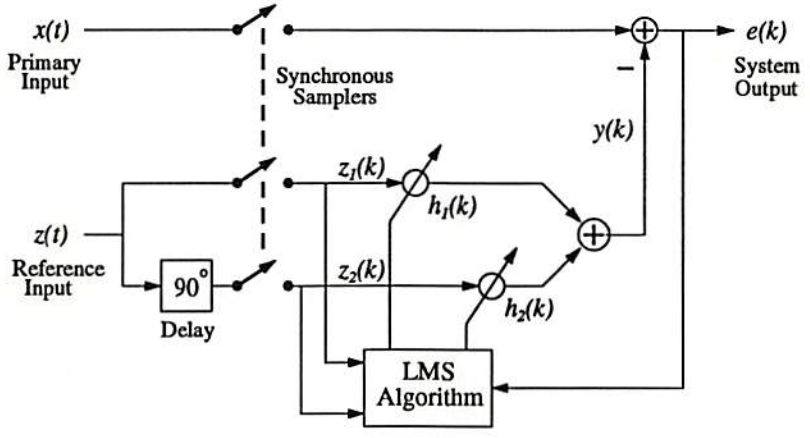

Figure 3.2: Two–weight configuration of adaptive notch filter using the LMS algorithm to mitigate a single–frequency interference.

be obtained [Widrow and Stearns, 1985]

$$H(z) = \frac{z^2 - 2z\cos\omega_0 + 1}{z^2 - 2(1-\mu C^2)z\cos\omega_0 + 1 - 2\mu C^2} \tag{3.2}$$

where $\omega_0 = 2\pi f_0 T$ and μ is the step size of the LMS algorithm. The zeros of the transfer function locate at

$$z = e^{\pm j\omega_0} \tag{3.3}$$

and the poles locate inside the unit circle at a distance

$$\left(1 - 2\mu C^2\right)^{\frac{1}{2}} \tag{3.4a}$$

from the origin and at angles of

$$\pm\ \cos^{-1}\left[(1-\mu C^2)\,(1-2\mu C^2)^{-1/2}\cos\omega_0\right]. \tag{3.4b}$$

The 3–dB rejection bandwidth of the notch filter is seen to be

$$2\mu C^2 \text{ rad } = \frac{\mu C^2}{\pi T} \text{ Hz.} \tag{3.5}$$

Note that this two–weight configuration is preferred only when separate reference signals can be obtained for each of multiple interfering sinusoids. Glover [1977] has presented and analyzed the LMS–based notch filter when the reference input consists of more than one sinusoidal signals whose frequencies are not separable. The linear and time–invariant notch filter response has been derived from the primary input $\{x(k)\}$ to the system output $\{e(k)\}$ when the reference signal is pure sinusoid without noise. Shensa [1980] has discussed the problem when there is a white noise in the reference input and showed that a time–invariant transfer function, in general, does not exist and the output spectrum depends explicitly on the primary input, not just its power spectrum.

The standard LMS algorithm is summarized in Table 3.1. Although we describe here only the standard LMS algorithm, note that a number of different algorithms have been developed which can be classified as LMS variants. They include the normalized LMS, leaky LMS, pilot LMS, clipped LMS, zero-forcing LMS, averaged LMS, momentum LMS, median LMS, and block LMS algorithms [Clarkson, 1993].

When the number of filter taps is M, the required number of multiplications per iteration for the standard LMS algorithm is given by

$$2\,M + 1. \tag{3.6}$$

The advantages of the LMS algorithm as an adaptive notch filter are as follows:

- The LMS algorithm is simple and easily understandable.
- The computational complexity of the LMS algorithm is very low.
- The LMS algorithm is easy and simple in implementation.

The disadvantages of the LMS algorithm as an adaptive notch filter are as follows:

Table 3.1: Summary of the Adaptive Notch Filter Based on the Least Mean Square (LMS) Algorithm

Parameters	$\mathbf{H} = [h_1, h_2, \ldots, h_M]^T$	filter coefficient vector
	M	number of filter taps
	μ	step size
Initial Condition	$\mathbf{H}(0) = \mathbf{0}$	
Data at k	$x(k)$	primary input
	$\mathbf{z}(k) = [z_1(k), z_2(k) \ldots, z_M(k)]^T$	reference vector
Computation	for $k = 0, 1, 2, 3, \ldots$	
	$e(k) = x(k) - \mathbf{H}^T(k)\,\mathbf{z}(k)$	
	$\mathbf{H}(k+1) = \mathbf{H}(k) + \mu\,\mathbf{z}(k)\,e(k)$	

- The average time constant for the LMS algorithm is inversely proportional to the step size parameter.

- The misadjustment is directly proportional to the step size parameter.

- The misadjustment increases linearly with the number of filter taps for a fixed average time constant.

- When the eigenvalues of the correlation matrix of the reference input are widely spread, the excess mean–squared error is determined by the largest eigenvalues and the convergence time for the average filter coefficient vector is limited by the smallest eigenvalues.

- When the eigenvalue spread is large, a large number of iterations are need for the convergence of the LMS algorithm.

Example 3.1

Let us assume that the primary input $\{x(k)\}$ consists of the sum of the sinusoidal SOI and an additive sinusoidal interference as follows.

$$x(k) = s(k) + I(k)$$

where $\{s(k)\}$ and $\{I(k)\}$ denote the SOI and the interference, respectively, given by

$$\begin{aligned} s(k) &= \sqrt{2}\,\sin(2\pi(0.4)\,k + \pi/3) \\ I(k) &= \sqrt{2}\,\sin(2\pi(0.3)\,k + \pi/4). \end{aligned}$$

Assume that the reference input is given by

$$z(k) = \sin(2\pi(0.3)\,k + \phi)$$

where ϕ is a random phase uniformly distributed over $[-\pi, \pi]$. When the number of filter taps M is 8 and the step size parameter μ is 0.0125, Fig. 3.3 shows the averaged magnitude response over 32 transfer functions, the averaged power spectrum $|E(j\omega)|$ over 32 estimated power spectra of the system outputs $\{e(k)\}$ after convergence, and a typical error curve between the system output and the SOI, $\{\epsilon(k)\}$, defined as

$$\epsilon(k) = e(k) - s(k).$$

Note that we do not use the special two–weight configuration shown in Fig. 3.2. Note that we use the last 256 system output data samples to estimate the power spectra.

As we can observe in the figure, the AIM filter based on the LMS algorithm has a notch at the frequency of the reference input sinusoid and mitigates the sinusoidal components related to the reference input from the primary input.

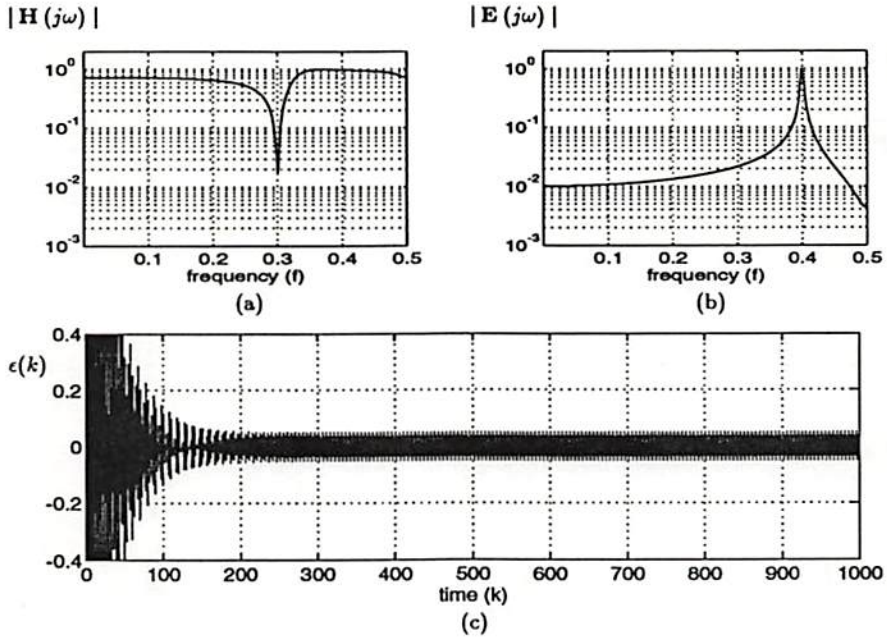

Figure 3.3: (a) Transfer function of the AIM filter based on the LMS algorithm, (b) the average of estimated and normalized power spectra of the system outputs, $\{e(k)\}$, obtained by the LMS–based AIM filter, and (c) an error curve $\{\epsilon(k) = e(k) - s(k)\}$.

3.4 Adaptive Notch Filter Using Lattice Structure

It is well known that the mean–convergence of the filter coefficients of an adaptive LMS filter described in the previous section is dependent on the eigenvalue spread of the filter input signal autocorrelation matrix [Clarkson, 1993; Haykin, 1991; Widrow and Stearns, 1985] To overcome this dependence the lattice structure adaptive filter has been proposed by a lot of researchers: Ding and Yu [1986], Griffiths [1977; 1978], Makhoul [1978], and Satorius *et al.* [1979]. The basic structure of the lattice form implementation of an AIM filter is illustrated in Fig. 3.4. Note that the

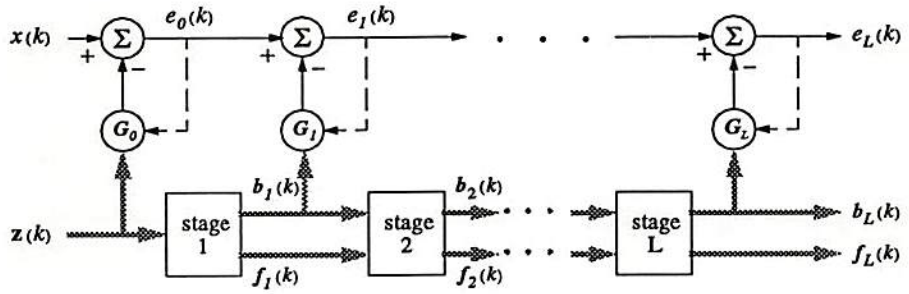

Figure 3.4: Basic structure of the lattice form implementation of adaptive interference mitigation filter using the reference input.

lattice structure AIM filter belongs to the first category. The forward and backward error sequences $\mathbf{f}_\ell(k)$ and $\mathbf{b}_\ell(k)$ at the ℓ stage have the following relationships.

$$\begin{aligned} \mathbf{f}_0(k) &= \mathbf{b}_0(k) = \mathbf{z}(k) \\ \mathbf{f}_{\ell+1}(k) &= \mathbf{f}_\ell(k) - \boldsymbol{\Gamma}_\ell(k)\, \mathbf{b}_\ell(k-1) \\ \mathbf{b}_{\ell+1}(k) &= -\boldsymbol{\Gamma}_\ell(k)\, \mathbf{f}_\ell(k) + \mathbf{b}_\ell(k-1) \end{aligned} \tag{3.7}$$

where $\mathbf{z}(k) = [z_1(k), z_2(k), \ldots, z_M(k)]^T$ and $\boldsymbol{\Gamma}_\ell(k)$ is the reflection coefficient or partial correlation coefficient matrix. The reflection coefficient matrix update equation is given by

$$\boldsymbol{\Gamma}_\ell(k+1) = \boldsymbol{\Gamma}_\ell(k) + \frac{\alpha_1}{\sigma_\ell^2(k)} \left[\mathbf{f}_\ell(k)\mathbf{b}_{\ell-1}^T(k-1) + \mathbf{b}_\ell(k)\mathbf{f}_{\ell-1}^T(k)\right] \tag{3.8}$$

where α_1 is a normalized adaptive step size parameter and $\sigma_\ell^2(k)$ is a measure of the power of the reference input at the ℓ^{th} stage updated by

$$\sigma_\ell^2(k) = \beta_1\, \sigma_\ell^2(k-1) + \frac{1-\beta_1}{M} \left[\mathbf{f}_\ell^T(k)\mathbf{f}_\ell(k) + \mathbf{b}_\ell^T(k-1)\mathbf{b}_\ell(k-1)\right] \tag{3.9}$$

where M is the dimension of $\mathbf{f}_\ell(k)$ and $\mathbf{b}_\ell(k-1)$. In (3.9) the parameter β_1 satisfies $0 < \beta_1 < 1$ and controls the resulting power averaging time. The power estimate is required at each stage in the lattice due to the fact that the powers of the forward and backward error sequences decrease with increased stage numbers. Due to orthogonalization of the entire interference canceling structure, each coefficient vector $\boldsymbol{G}_\ell(k)$, which provides the interference canceling subtraction path in Fig. 3.4, can be determined independently of $\boldsymbol{G}_m(k)$ for $m > \ell$. Thus, the optimal $\boldsymbol{G}_\ell^*$ that minimizes $\mathrm{E}\{e_\ell^2(k)\}$ is given by

$$\boldsymbol{G}_\ell^* = \Big(\mathrm{E}\{\mathbf{b}_\ell(k)\mathbf{b}_\ell^T(k)\}\Big)^{-1} \mathrm{E}\{e_{\ell-1}(k)\mathbf{b}_\ell(k)\}. \tag{3.10}$$

The appropriate LMS algorithm and associated power measurement $\gamma_\ell^2(k)$ are given by

$$\boldsymbol{G}_\ell(k+1) = \boldsymbol{G}_\ell(k) + \frac{\alpha_2}{\gamma_\ell^2(k)}\, e_\ell(k)\mathbf{b}_\ell(k) \tag{3.11a}$$

$$\gamma_\ell^2(k+1) = \beta_2\, \gamma_\ell^2(k) + \frac{1-\beta_2}{M}\, \mathbf{b}_\ell^T(k)\mathbf{b}_\ell(k) \tag{3.11b}$$

where α_2 and β_2 are step sizes. Due to the orthogonality of successive stages, increasing the time dimension from L to $L+1$ does not result in a change of the coefficients associated with the first L stages. The time constant of any one stage is determined only by the correlation statistics of that particular stage. The frequency tracking behavior and the narrowband interference mitigation performance from direct sequence spread spectrum (DSSS) signals of the adaptive stochastic gradient (SG) lattice filter have been discussed in [Zeidler *et al.*, 1991]. The performance of the interference mitigation performance from recursive least squares lattice (RLSL) and the normalized step–size SG lattice algorithms have been compared to the LMS algorithm for sinusoidal interference cancellation in [North *et al.*, 1992]. The adaptive lattice filters usually shows

little harmonic distortion and their performance is not influenced by the reference power.

The adaptive notch filter algorithm based on the lattice structure is summarized in Table 3.2. When the number of stages is L and the number of filter taps at each stage is M, the required number of multiplications per iteration of the lattice structure AIM filter algorithm is given by

$$L\left(5M^2+4M+9\right). \tag{3.12}$$

The advantages of the lattice structure adaptive notch filter are as follows.

- More fast convergence speeds are achievable than the LMS algorithm.
- The time constant of convergence of the mean of the filter coefficients is independent of the eigenvalue spread of the input autocorrelation matrix.
- For approximately the same amount of primary signal distortion, the lattice structure adaptive notch filter produces less harmonic distortion than the LMS algorithm.

The disadvantages of the lattice structure adaptive notch filter are as follows.

- The computational load increases linearly with the number of dimensions in the reference channel.
- Additional computations are needed for power estimate at each stage in the lattice due to the fact that the forward and backward error sequence have decreased power with increased stage number.
- The step size must be normalized by the power level to maintain the same adaptive time constant and misadjustment at each stage in the lattice.

- The performance is depending on a lot of parameters: α_1, α_2, β_1, β_2, and initial conditions.

Table 3.2: Summary of the Lattice Structure Adaptive Notch Filter Algorithm

Parameters	$\mathbf{G}_\ell$	filter coefficient vector at stage ℓ
	Γ_ℓ	reflection coefficients at stage ℓ
	M	number of filter taps
	L	number of stages
	α_1, α_2, β_1 and β_2	step sizes
Initial Conditions	for $\ell = 1, 2, \ldots, L$	
	$\mathbf{f}_\ell(0) = \mathbf{b}_\ell(0) = \mathbf{0}$	
	$\mathbf{G}_\ell(0) = \Gamma_\ell(1) = 0$	
	$\sigma_\ell^2(0) = \gamma_\ell^2(0) = c$	where c is a small positive constant
Data at k	$x(k)$	primary input
	$\mathbf{z}(k) = [\, z_1(k), z_2(k), \ldots, z_M(k) \,]^T$	reference vector
Computation	for $k = 1, 2, 3, \ldots$	
	$\mathbf{f}_0(k) = \mathbf{b}_0(k) = \mathbf{z}(k)$	
	$e_0(k) = x(k)$	
	for $\ell = 1, 2, \ldots, L$	
	$\mathbf{f}_\ell(k)$	$= \mathbf{f}_{\ell-1}(k) - \Gamma_\ell^*(k)\, \mathbf{b}_{\ell-1}(k-1)$
	$\mathbf{b}_\ell(k)$	$= -\Gamma_\ell(k)\, \mathbf{f}_{\ell-1}(k) + \mathbf{b}_{\ell-1}(k-1)$
	$\sigma_\ell^2(k)$	$= \beta_1\, \sigma_\ell^2(k-1) + \frac{1-\beta_1}{M} \cdot \left[\mathbf{f}_\ell^T(k)\mathbf{f}_\ell(k) + \mathbf{b}_\ell^T(k-1)\mathbf{b}_\ell(k-1)\right]$
	$\Gamma_\ell(k+1)$	$= \Gamma_\ell(k) + \frac{\alpha_1}{\sigma_\ell^2(k)} \cdot \left[\mathbf{f}_{\ell-1}(k)\mathbf{b}_\ell^T(k) + \mathbf{b}_{\ell-1}(k-1)\mathbf{f}_\ell^T(k)\right]$
	$\gamma_\ell^2(k)$	$= \beta_2\, \gamma_\ell^2(k-1) + \frac{1-\beta_2}{M} \mathbf{b}_\ell^T(k-1)\mathbf{b}_\ell(k-1)$
	$\mathbf{G}_\ell(k)$	$= \mathbf{G}_\ell(k-1) + \frac{\alpha_2}{\gamma_\ell^2(k)} e_\ell(k-1)\, \mathbf{b}_\ell(k-1)$
	$e_\ell(k)$	$= e_{\ell-1}(k) - \mathbf{G}_\ell^T(k)\, \mathbf{b}_\ell(k)$

Example 3.2

Let us consider the same SOI $\{s(k)\}$, interference $\{I(k)\}$, and reference input $\{z(k)\}$ with those in Example 3.1. The number of filter taps M is 4 and the number of stages L is 2. The step size parameters are $\alpha_1 = 0.011$, $\alpha_2 = 0.0105$, $\beta_1 = 0.995$, and $\beta_2 = 0.995$. The transfer function of the lattice structure adaptive notch filter, the averaged power spectrum $|E(j\omega)|$ over 32 estimated power spectra of the system output $\{e(k)\}$ after convergence, and an error curve between the SOI and the system output, $\{\epsilon(k)\}$, are illustrated in Fig. 3.5. Note again that we use the

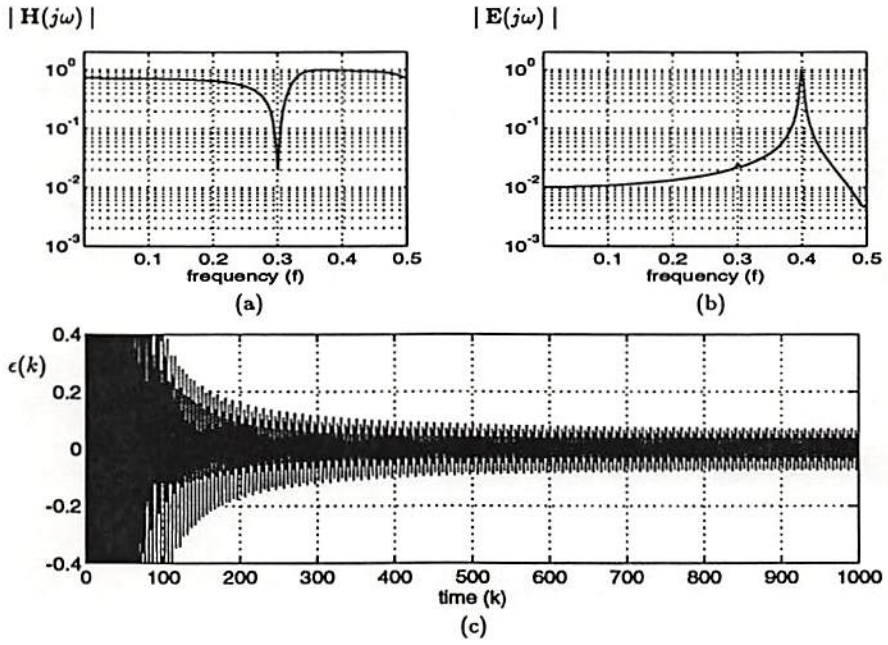

Figure 3.5: (a) Transfer function of the AIM filter using lattice structure with $M = 4$ and $L = 2$, (b) the average of estimated and normalized power spectra of the system outputs, $\{e(k)\}$, obtained by the AIM filter using lattice structure, and (c) an error curve $\{\epsilon(k) = e(k) - s(k)\}$.

last 256 system output data to estimate their power spectra. Although Satorius, Smith, and Reeves [1979] have discussed and implemented the sinusoidal interference algorithm with two–tap configurations, we consider here more general configuration that can be used in non–separable multiple interference sinusoids. Fig. 3.6 shows the transfer function of the adaptive notch filter and the average of 32 estimated power spectra of the system output after convergence when $M = 4$, $L = 3$, $\alpha_1 = 0.008$, $\alpha_2 = 0.00775$, $\beta_1 = 0.975$, and $\beta_2 = 0.985$.

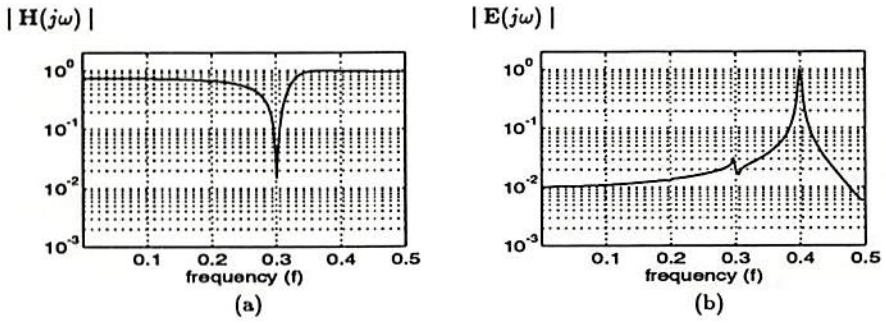

Figure 3.6: (a) Transfer function of the AIM filter using lattice structure with $M = 4$ and $L = 3$ and (b) the average of estimated and normalized power spectra of the system outputs, $\{e(k)\}$, obtained by the AIM filter using lattice structure.

Through computer simulations, we notice that the convergence speed of the algorithm depends not only α_1 and α_2, but also β_1 and β_2. Note that the values of α_1 and α_2 are usually much less than unity and that the values of β_1 and β_2 are close to unity. We observe that the larger the values of α's are, the faster the convergence speed is at the expense of larger excess error; and also that the algorithm might diverge when a value of β's is too close to unity. Sometimes, it is not easy to find proper values of these parameters in order to achieve fast convergence and small excess error of the algorithm.

3.5 Adaptive Notch Filter via Sequential Regression

We present in this section an adaptive notch filter which is derived via a sequential regression (SER) formulation [Soldan and Ahmed, 1978]. Let us consider an adaptive digital filter whose transfer function is given by

$$H_k(z) = h_0(k) + h_1(k)z^{-1} + \cdots + h_M(k)z^{-M} \tag{3.13}$$

where k is the time index and $h_i(k)$ denotes the i^{th} filter coefficient. When the input to the filter is $\{z(k), z(k-1), \ldots, z(k-M)\}$, the output $\{y(k)\}$ can be expressed as

$$y(k) = \mathbf{H}^T(k)\, \mathbf{z}(k) \tag{3.14}$$

where

$$\mathbf{H}(k) = [h_0(k), h_1(k), \ldots, h_M(k)]^T \tag{3.15a}$$

$$\mathbf{z}(k) = [z(k), z(k-1), \ldots, z(k-M)]^T . \tag{3.15b}$$

Note that the SER–based AIM filter belongs to the first category. When $\{x(k)\}$ denotes the primary input at time k, the cost function is defined as

$$V\left(\mathbf{H}(k+1)\right) = q \sum_{i=1}^{k} \left[x(i) - \mathbf{H}^T(k+1)\, \mathbf{z}(i)\right]^2 + \mathbf{H}^T(k+1)\mathbf{H}(k+1) \tag{3.16}$$

where q is a scalar. We want to minimize the cost function with respect to the filter coefficient vector $\mathbf{H}(k+1)$. When we assume $\nabla_{\mathbf{H}(k+1)} x(i) = 0$, the relation $\nabla_{\mathbf{H}(k+1)} V\left(\mathbf{H}(k+1)\right) = 0$ results in

$$\mathbf{H}(k+1) = \mathbf{P}_k^{-1} \sum_{i=1}^{k} x(i)\mathbf{z}(i) \tag{3.17}$$

where

$$\mathbf{P}_k = \mathbf{I} + q \sum_{i=1}^{k} \mathbf{z}(i)\mathbf{z}^T(i). \tag{3.18}$$

The matrix $\mathbf{P}_k$ is called a weighted input data correlation matrix. After applying some manipulations to (3.17), we obtain the recursive equation for the filter coefficient vector $\mathbf{H}(k+1)$ [Parikh and Ahmed, 1980] given by

$$\mathbf{H}(k+1) = \mathbf{H}(k) + q\,\mathbf{P}_k^{-1}\,\mathbf{z}(k)\,e(k) \tag{3.19}$$

where $e(k) = x(k) - y(k)$ is the error (system output) at the k^{th} iteration. The matrix $\mathbf{P}_k^{-1}$ can also be computed recursively using the matrix inversion lemma

$$\mathbf{P}_k^{-1} = \mathbf{P}_{k-1}^{-1} - \frac{1}{\gamma}\,\mathbf{P}_{k-1}^{-1}\mathbf{z}(k)\mathbf{z}^T(k)\mathbf{P}_{k-1}^{-1} \tag{3.20}$$

where $\gamma = 1/q + \mathbf{z}^T(k)\mathbf{P}_{k-1}^{-1}\mathbf{z}(k)$ and $\mathbf{P}_0^{-1} = \mathbf{I}$. The equation (3.19) is referred to as the SER algorithm. When the reference signal is

$$z(k) = C\,\cos(\omega_0 kT + \theta_0) \tag{3.21a}$$

where T denotes the sampling interval, the i^{th} component of $\mathbf{z}(k)$ may be expressed as

$$z_i(k) = z(k-i) = C\,\cos(\omega_0 kT + \theta_i) \quad \text{for } i = 0, 1, \ldots, M. \tag{3.21b}$$

In this case, Parikh and Ahmed [1980] have showed that the output of the SER–based AIM filter contains the corresponding time–invariant and time–variant terms. When the time–varying terms can be small and discarded by increasing the number of filter input signal M, the overall transfer function is represented by [Parikh and Ahmed, 1980]

$$H(z) = \frac{z^2 - 2z\cos(\omega_0 T) + 1}{z^2 - 2\left(1 - \frac{qc^2\xi}{4}\right)z\cos(\omega_0 T) + \left(1 - \frac{qc^2\xi}{2}\right)} \tag{3.22}$$

where

$$\xi = \sum_{i=0}^{M} \sum_{m=i}^{M} b_{i,m} P_{i,m} \cos((i-m)\omega_0 T) \tag{3.23a}$$

$$b_{i,m} = \begin{cases} 1, & i = m \\ 2, & i \neq m. \end{cases} \tag{3.23b}$$

In this expression, $P_{i,m}$ denotes the $(i,m)^{\text{th}}$ element of $\mathbf{P}_k^{-1}$. Note that the overall transfer function is of a second–order notch filter whose notch frequency is ω_0 and 3–dB rejection bandwidth is

$$\frac{q\xi C^2}{2T}. \tag{3.24}$$

The adaptive notch filter based on the SER algorithm is summarized in Table 3.3. When the number of filter taps is M, the required number of multiplications per iteration for the adaptive notch filter using SER algorithm is given by

$$3M^2 + 3M + 1. \tag{3.25}$$

The advantages of the adaptive notch filter based on the SER algorithm are as follows.

- The convergence rate of adaptation is substantially faster than the LMS algorithm with the same number of filter tabs.
- The SER–based notch filter can realize a sharper notch over a large bandwidth than the LMS–based notch filter with the same number of filter tabs.

The disadvantages of the adaptive notch filter based on the SER algorithm are as follows.

- The AIM filter based on the SER algorithm needs the high computational complexity per iteration and large storage requirements.

Table 3.3: Summary of the Adaptive Notch Filter Using Sequential Regression (SER) Algorithm

Parameters	$\mathbf{H} = [h_1, h_2, \ldots, h_M]^T$	filter coefficient vector
	M	number of filter taps
	q	constant
Initial Conditions	$\mathbf{H}(0) = \mathbf{0}$	
	$\mathbf{P}_0^{-1} = \mathbf{I}$	
Data at k	$x(k)$	primary input
	$\mathbf{z}(k) = [z_1(k), z_2(k), \ldots, z_M(k)]^T$	reference vector
Computation	for $k = 0, 1, 2, 3, \ldots$	
	$e(k) = x(k) - \mathbf{H}^T(k)\,\mathbf{z}(k)$	
	$\Gamma(k) = \mathbf{P}_k^{-1}\,\mathbf{z}(k)$	
	$\mathbf{H}(k+1) = \mathbf{H}(k) + q\,\Gamma(k)\,e(k)$	
	$\gamma = \frac{1}{q} + \mathbf{z}^T(k)\,\Gamma(k)$	
	$\mathbf{P}_k^{-1} = \mathbf{P}_{k-1}^{-1} - \frac{1}{\gamma}\,\Gamma(k)\,\Gamma^T(k)$	

Example 3.3

Let us consider the same SOI $\{s(k)\}$, interference $\{I(k)\}$, and reference input $\{z(k)\}$ with those in Examples 3.1 and 3.2. When the number of filter taps M is 8 and the constant parameter $q = 1$, Fig. 3.7 shows the transfer function of the notch filter based on SER algorithm, the averaged power spectrum over 32 estimated power spectra of the system output $\{e(k)\}$ after convergence, and an error curve $\{\epsilon(k) = e(k) - s(k)\}$.

From this figure, we can observe that the SER–based AIM filter mitigates the sinusoidal interference. Compared to Fig. 3.3, the SER–based AIM filter converges faster and has smaller excess error than the LMS–based AIM filter at the expense, however, of larger computational complexity and storage requirements.

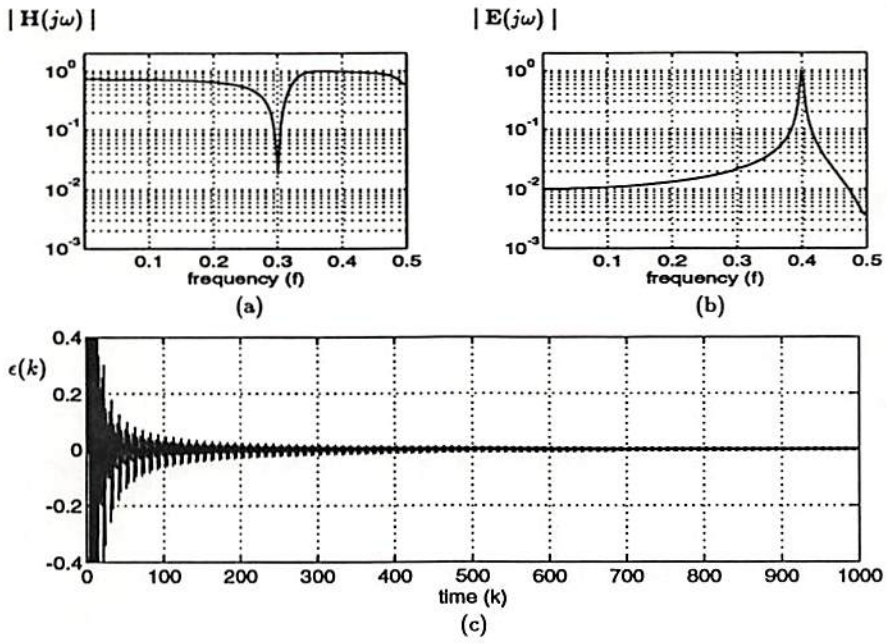

Figure 3.7: (a) Transfer function of the AIM filter based on the SER algorithm, (b) the average of estimated and normalized power spectra of the system outputs, $\{e(k)\}$, obtained by the SER–based AIM filter, and (c) an error curve $\{\epsilon(k) = e(k) - s(k)\}$.

3.6 Adaptive IIR Notch Filter

Adaptive infinite impulse response (IIR) filters have also been studied as an AIM filter in the second category by many researchers: Bhaskar Rao and Kung [1984], Friedlander and Smith [1984], Hush *et al.* [1986], Kwan and Martin [1989], Kung and Bhaskar Rao [1982], Nehorai [1985], Nishimura *et al.* [1989], and Petraglia *et al.* [1990a]. An adaptive IIR notch filter exhibits a performance similar to that of the adaptive FIR notch filter, but it requires significantly fewer coefficients and less computational complexity. A tutorial–style discussion including current results and open issues on adaptive IIR filters has been presented

in [Johnson, 1984]. We summarize in this section the most important AIM filter techniques based on IIR realization structures.

3.6.1 Constrained Adaptive IIR Notch Filter

Bhaskar Rao and Kung [1982; 1984] have proposed a constrained IIR filter employing a frequency domain and time domain for the enhancement and tracking of sinusoids in additive noise. The proposed transfer function of the notch filter is

$$H(z) = \frac{\prod_{i=1}^{2p} \left(1 - z_i z^{-1}\right)}{\prod_{i=1}^{2p} \left(1 - p_i z^{-1}\right)} \tag{3.26}$$

where p is the number of narrowband signals. The poles and zeros of the transfer function are constrained to lie on the same radial line with the poles $\{p_i\}$ lying in between the zeros $\{z_i\}$ and the origin. This constraints on the poles and zeros can be represented as

$$p_i = \alpha z_i \quad \text{for } i = 1, 2, \ldots, 2p \tag{3.27}$$

where $0 \leq \alpha < 1$ is called the debiasing parameter whose role has been discussed in [Bhaskar Rao and Kung, 1984]. Using the relationship between the zeros and the poles, we can rewrite (3.26) as

$$H(z) = \frac{W(z)}{W(\alpha\, z)} \tag{3.28}$$

where

$$W(z) = 1 - h_1 z^{-1} - h_2 z^{-2} - \cdots - h_{2p} z^{-2p} \tag{3.29a}$$

$$W(\alpha\, z) = 1 - \alpha\, h_1 z^{-1} - \alpha^2\, h_2 z^{-2} - \cdots - \alpha^{2p}\, h_{2p} z^{-2p}. \tag{3.29b}$$

Note that when $\alpha = 0$ for a special case, the transfer function becomes the popular linear prediction filter. Defining the following intermediate

term

$$\begin{aligned} \tilde{x}(k) &\triangleq \frac{x(k)}{W(\alpha\, z)} \\ &= x(k) + \sum_{j=1}^{2p} \alpha^j\, h_j \tilde{x}(k-j), \end{aligned} \tag{3.30}$$

we can obtain the system output $\{e(k)\}$ as

$$\begin{aligned} e(k) &= W(z)\tilde{x}(k) \\ &= \tilde{x}(k) - h_1\tilde{x}(k-1) - h_2\tilde{x}(k-2) - \cdots - h_{2p}\tilde{x}(k-2p). \end{aligned} \tag{3.31}$$

The stochastic Gauss–Newton method has been used to adaptively adjust the filter coefficients $\{h_i\}$. Using an approximate Hessian matrix and the matrix inversion lemma [Soderstrom *et al.*, 1978], we obtain

$$\begin{aligned} \mathbf{H}(k) &= \mathbf{H}(k-1) + \mathbf{K}(k)\, e(k) \\ \mathbf{K}(k) &= \frac{\mathbf{P}(k-1)\psi(k)}{\lambda(k) + \psi^T(k)\mathbf{P}(k-1)\psi(k)} \\ \mathbf{P}(k) &= \frac{1}{\lambda(k)}\left[\mathbf{P}(k-1) - \frac{\mathbf{P}(k-1)\psi(k)\psi^T(k)\mathbf{P}(k-1)}{\lambda(k) + \psi^T(k)\mathbf{P}(k-1)\psi(k)}\right] \\ \lambda(k) &= \beta\,\lambda(k-1) + (1-\beta) \\ \psi(k) &= -\frac{de(k)}{d\mathbf{H}} = \tilde{\mathbf{x}}(k) - \Lambda\,\tilde{\mathbf{e}}(k) \end{aligned} \tag{3.32}$$

where

$$\mathbf{H}(k) = [\, h_1(k), h_2(k), \ldots, h_{2p}(k)\,]^T \tag{3.33a}$$

$$\tilde{\mathbf{x}}(k) = [\, \tilde{x}(k-1), \tilde{x}(k-2), \ldots, \tilde{x}(k-2p)\,]^T \tag{3.33b}$$

$$\tilde{\mathbf{e}}(k) = [\, \tilde{e}(k-1), \tilde{e}(k-2), \ldots, \tilde{e}(k-2p)\,]^T \tag{3.33c}$$

$$\Lambda = \text{diag}\left(\alpha, \alpha^2, \ldots, \alpha^{2p}\right). \tag{3.33d}$$

In the expression of (3.33c), $\tilde{e}(k)$ is given by

$$\tilde{e}(k) = \frac{e(k)}{W(\alpha\, z)} = e(k) + \mathbf{H}^T\,\Lambda\,\tilde{\mathbf{e}}(k). \tag{3.34}$$

Although the gradient $\psi(k)$ in (3.32) is a complete one, a simplified gradient has been proposed

$$\psi(k) = \tilde{\mathbf{x}}(k). \tag{3.35}$$

Bhaskar Rao and Kung [1984] have also discussed several related problems: the complete and simplified gradients, convergence analysis, effect of filter length, existence of local minima, and cascade and parallel implementation forms. Although the complete gradient need a heavy computational complexity, it leads the IIR filter to the Wiener–Hopf equation for $\alpha = 0$ and to the optimum notch filter for $\alpha \simeq 1$. Although gradient truncation often simplifies computations and improves stability, the IIR filter may not always converge to the optimal estimates. The simplified gradient in (3.35) is more appropriate for the white noise case. When the filter length exceeds the required length, the excess pole–zero pairs tend to go inside the unit circle till they become ineffective. On the other hand, when the order of the IIR notch filter is lower than required, the filter removes sinusoidal interferences as half as the number of filter order (existence of local minima). The cascade and parallel implementations using second–order filters have been proposed with several advantages: alleviating the polynomial deflation problem, using multiprocessors for real–time processing, and easily incorporating a priori information and constraints on the location of the roots.

The constrained adaptive IIR notch filter algorithm is summarized in Table 3.4. When the number of filter taps is M, the required number of multiplications per iteration for the constrained adaptive IIR notch filter algorithm is given by

$$5\,M^2 + 8\,M + 2. \tag{3.36}$$

The advantages of the constrained adaptive IIR notch filter algorithm are as follows.

Table 3.4: Summary of the Constrained Adaptive IIR Notch Filter Algorithm

Parameters	$\mathbf{H} = [\, h_1, h_2, \ldots, h_{2p} \,]^T$	filter coefficient vector
	p	number of poles
	α	debiasing parameter
	β	step size
Initial Conditions	$\mathbf{H}(0) = \mathbf{0}$	
	$\Lambda = \text{diag}\ (\alpha, \alpha^2, \ldots, \alpha^{2p})$	
	$\mathbf{P}(0) = c_1 \mathbf{I}$	where c is a constant
	$\lambda(0) = c_2$	where c_2 is a constant
Computation	for $k = 0, 1, 2, 3, \ldots$	
	$\tilde{\mathbf{x}}(k) = [\, \tilde{x}(k-1), \tilde{x}(k-2), \ldots, \tilde{x}(k-2p) \,]^T$	
	$\tilde{x}(k) = x(k) + \mathbf{H}^T(k) \Lambda \tilde{\mathbf{x}}(k)$	
	$e(k) = \tilde{x}(k) - \mathbf{H}^T(k) \tilde{\mathbf{x}}(k)$	
	$\tilde{\mathbf{e}}(k) = [\, \tilde{e}(k-1), \tilde{e}(k-2), \ldots, \tilde{e}(k-2p) \,]^T$	
	$\tilde{e}(k) = e(k) + \mathbf{H}^T \Lambda \tilde{\mathbf{e}}(k)$	
	$\psi(k) = \tilde{\mathbf{x}}(k) - \Lambda\, \tilde{\mathbf{e}}(k)$	
	$\lambda(k+1) = \beta\, \lambda(k) + (1 - \beta)$	
	$\mathbf{K}(k+1) = \frac{\mathbf{P}(k)\psi(k)}{\lambda(k+1) + \psi^T(k)\mathbf{P}(k)\psi(k)}$	
	$\mathbf{P}(k+1) = \frac{1}{\lambda(k+1)} \left[\mathbf{P}(k) - \frac{\mathbf{P}(k)\psi(k)\psi^T(k)\mathbf{P}(k)}{\lambda(k+1) + \psi^T(k)\mathbf{P}(k)\psi(k)} \right]$	
	$\mathbf{H}(k+1) = \mathbf{H}(k) + \mathbf{K}(k)\, e(k)$	

- The constrained adaptive IIR notch filter requires few parameters.

- The closer the debiasing parameter α is to unity, the flatter the notch filter (ideal notch filter) response will be.

- When the filter length exceeds the required length (overestimation cases), the excess pole–zero pairs tend to go inside the unit circle till they become ineffective.

- When the order of a filter is lower than required (underestimation cases), the notch filter of order $2r$ will remove r of the p sinusoidal interferences for $r \leq p$.

The disadvantages of the constrained adaptive IIR notch filter algorithm are as follows.

- Although the constrained adaptive IIR notch filter becomes an ideal notch filter as the debiasing parameter α approaches to unity, due to the nonlinearity of the gradient, the resulting notch position has a large bias and variance if α is too close to unity.
- As the value of α increases, the DC gain decreases.

Example 3.4

Let us assume that the input consists of a wideband SOI and an additive sinusoidal interference as follows:

$$x(k) = s(k) + I(k)$$

where $\{s(k)\}$ and $\{I(k)\}$ denote the SOI and the interference, respectively. We assume that the SOI is a zero–mean white Gaussian random process with unity variance and that the interference is the sinusoidal signal given by

$$I(k) = A_I \sin(2\pi(0.3)\, k + \phi)$$

where A_I is the amplitude and ϕ is a random initial phase uniformly distributed over $[-\pi, \pi]$. Then, the signal–to–interference ratio (SIR) is defined as

$$\text{SIR (dB)} = 10 \log_{10} \frac{1}{A_I^2/2}.$$

When $p = 1$; $\beta = 0.99$; $\mathbf{P}(0) = 0.01\,\mathbf{I}$; $\lambda(0) = 0.95$; the number of data samples is 1,000; and the SIR is -3 dB, Fig. 3.8 shows the averaged magnitude response over 32 transfer functions of the constrained adaptive IIR notch filters with the debiasing parameter $\alpha = 0.5, 0.75, 0.9$, and 0.95. The mean and standard deviation values of the deepest notch positions for each α are given in Table 3.5.

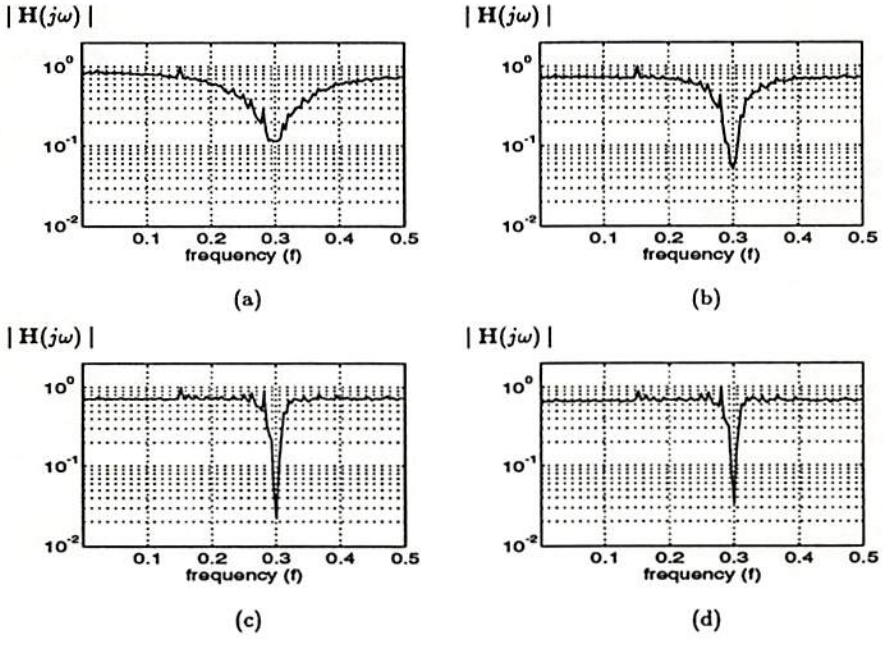

Figure 3.8: Averaged transfer functions of the constrained adaptive IIR notch filter when (a) $\alpha = 0.5$, (b) $\alpha = 0.75$, (c) $\alpha = 0.9$, and (d) $\alpha = 0.95$.

From the Fig. 3.8 and Table 3.5, we can observe that as the debiasing parameter α approaches to unity, the transfer function of the constrained adaptive IIR notch filter approaches that of an ideal notch filter (depth of notch increases and rejection bandwidth decreases); also, the adaptive notch filter has a notch located at the required position with small variances. Note, however, that as the value of α increases, the DC gain

Table 3.5: The Mean and Standard Deviation Values of the Deepest Notch Positions Obtained by the Constrained Adaptive IIR Notch Filter

value of α	mean	standard deviation
0.5	0.2865	0.0553
0.75	0.2991	0.0067
0.9	0.2998	0.0017
0.95	0.3010	0.0024

decreases and that if the value of α is too close to unity, the depth of notch reduces and the bias and variance of the obtained notch position increase.

3.6.2 Recursive Maximum Likelihood Adaptive IIR Notch Filter

Many adaptive IIR notch filters have a rational transfer function

$$H(z) = \frac{A(z)}{C(z)} = \frac{1 + a_1 z^{-1} + \cdots + a_m z^{-m}}{c_0 + c_1 z^{-1} + \cdots + c_n z^{-n}} \tag{3.37}$$

where $\{a_i\}$ and $\{c_i\}$ are the adjustable parameters. This filter is an optimal predictor for the input time series $\{x(k)\}$, when the time series is assumed to be an autoregressive moving average (ARMA) process of order (m, n). However, the filter will not have the desired characteristics of a notch filter. To obtain the desired notch response, Friedlander and Smith [1984] have developed the following transfer function

$$H(z) = 1 + \rho\, z^{-\nu}\, \frac{\tilde{C}(z)}{C(z)} \tag{3.38}$$

where $0 \leq \rho \leq 1$, ν is a nonnegative integer, and

$$C(z) = 1 + c_1 z^{-1} + c_2 z^{-2} + \cdots + c_n z^{-n} \tag{3.39a}$$

$$\tilde{C}(z) = c_n + c_{n-1} z^{-1} + \cdots + z^{-n} = z^{-n}\, C(z^{-1}). \tag{3.39b}$$

The three lemmas providing the necessary information of the zero locations of $H(z)$ have been presented in [Friedlander and Smith, 1984]. When the time delay parameter ν equals to unity, the filter becomes the one–step ahead prediction error filter. For $\nu = 0$, the adaptive algorithm may converge to the trivial solution. And, the case $\nu > 1$ is useful to notch filtering of signals in correlated noise, but introduces superfluous notches. The parameter ρ is introduced to control the depth of the notch. Setting ρ to a value slightly less than unity moves the zeros slightly inside the unit circle and let $H(z)$ have a stable inverse $1/H(z)$. The parameter n is related to the number of notches. When $\nu = 1$, there will be a fixed notch at $\omega = \pi$, in addition to the $\frac{n}{2}$ adaptive notches related to $C(z)$. To derive the recursive maximum likelihood (RML) algorithm for adaptively adjusting the parameters of the notch filter, let us consider the sum of squared prediction error

$$V_K = \frac{1}{K+1} \sum_{k=0}^{K} e^2(k) \tag{3.40}$$

where

$$e(k) = \left[1 + \rho\, z^{-\nu}\, \frac{\tilde{C}(z)}{C(z)} \right] x(k). \tag{3.41}$$

We can rewrite (3.41) as

$$e(k) = \left(x(k) + \rho\, x(k-n-\nu) \right) - \boldsymbol{\phi}^T(k)\boldsymbol{\theta} \tag{3.42}$$

where

$$\boldsymbol{\phi}(k) = [\, \phi_1(k), \phi_2(k), \ldots, \phi_n(k) \,]^T \tag{3.43a}$$

$$\phi_i(k) = -x(k-i) - \rho\, x(k-n-\nu+i) + e(k-i) \tag{3.43b}$$

$$\boldsymbol{\theta} = [\, c_1, c_2 \ldots, c_n \,]^T. \tag{3.43c}$$

Using a general Gauss–Newton algorithm for minimizing the squared prediction error in [Ljung, 1981], we have

$$\begin{aligned} \mathbf{R}(k) &= \mathbf{R}(k-1) + \psi(k)\psi^T(k) \\ e(k) &= (\, x(k) + \rho\, x(k-n-\nu)\,) - \phi^T(k)\hat{\theta}(k-1) \\ \hat{\theta}(k) &= \hat{\theta}(k-1) + \mathbf{R}^{-1}(k)\psi(k)e(k) \end{aligned} \tag{3.44}$$

where and $\hat{\theta}(k)$ is the vector of the current estimate of θ and $\psi(k)$ is the gradient vector given by

$$\psi(k) = \left[-\frac{\partial e(k)}{\partial \hat{c}_1}, -\frac{\partial e(k)}{\partial \hat{c}_2}, \ldots, -\frac{\partial e(k)}{\partial \hat{c}_n} \right]^T. \tag{3.45}$$

An expression for $\psi(k)$ can be obtained [Friedlander and Smith, 1984]

$$\psi(k) = \frac{1}{\hat{C}_{k-1}(z)}\, \phi(k) \tag{3.46}$$

where $\hat{C}_k(z)$ is the $C(z)$ polynomial with $\{c_i\}$ replaced by their estimates $\{\hat{c}_i(k)\}$ at time k. Equations (3.44) and (3.46) summarize the *basic RML algorithm* for the adaptive IIR notch filter. However, in order to improve the convergence, tracking capability, computational aspects, and notch bandwidth of the RML algorithm, we need the following several modifications on the basic RML algorithm [Friedlander and Smith, 1984]: (i) It has been observed that the convergence rate of the algorithm improves when the vector $\psi(k)$ is computed by the the following *a posteriori* prediction error

$$e(k) = (\, x(k) + \rho\, x(k-n-\nu)\,) - \phi^T(k)\hat{\theta}(k). \tag{3.47}$$

(ii) To make the algorithm capable of tracking nonstationary signals, the forgetting factor can be introduced by using a weighted least–squares criterion given by

$$V_K = \frac{1}{K+1} \sum_{k=0}^{K} \lambda^{K-k}\, e^2(k) \tag{3.48}$$

where $0 \leq \lambda \leq 1$ is the forgetting factor. The following time–varying forgetting factor is useful in applications involving either stationary or nonstationary signals.

$$\lambda(k) = \beta\,\lambda(k-1) + (1-\beta) \tag{3.49}$$

where $0 \leq \beta \leq 1$. (iii) Defining $\mathbf{P}(k) \triangleq \mathbf{R}^{-1}(k)$ and using the matrix inversion lemma, we can avoid the inversion of $\mathbf{R}(k)$ at every iteration. (iv) To ensure the convergence of the RML algorithm, it is necessary to keep $1/\hat{C}_k(z)$ stable. Thus, the stability should be tested at each iteration. If unstable, the parameters need to be projected back into a stability region. For example, we can replace $\{\hat{c}_i(k)\}$ by $\{\eta^i\,\hat{c}_i(k)\}$, where $0 \leq \eta \leq 1$. (v) It has been found to be useful to replace the prefilter $1/\hat{C}_{k-1}(z)$ in (3.46) by $1/\hat{C}_{k-1}(z/\alpha)$, where

$$\hat{C}_k\left(\frac{z}{\alpha}\right) = 1 + \alpha\,\hat{c}_1(k)z^{-1} + \alpha^2\,\hat{c}_2(k)z^{-2} + \cdots + \alpha^n\,\hat{c}_n(k)z^{-n} \tag{3.50}$$

where $0 \leq \alpha \leq 1$. The effect of the parameter α tends to reduce the number of times the polynomial $\hat{C}_k(z)$ becomes unstable and accelerate the convergence of the RML algorithm. The zero value of α leads to the so called extended least–squares (ELS) algorithm [Friedlander, 1982a]. Moreover, it has been found to be useful to let α be time varying as follows

$$\alpha(k+1) = \gamma\,\alpha(k) + (1-\gamma) \tag{3.51}$$

where $0 \leq \gamma \leq 1$. (vi) To avoid consistently wide notches, the prediction error in (3.44) is replaced by

$$\epsilon(k) = e(k) + \mu\,(\,\bar{c}_n - \hat{c}_n(k-1)\,) \tag{3.52}$$

where $0 \leq \bar{c}_n \leq 1$, and $\bar{c}_n$ is chosen to unity. It causes the following new gradient vector given by

$$\psi(k) = \phi(k) + [\,0, 0, \ldots, 0, \mu\,]^T\,. \tag{3.53}$$

After these modifications, the final RML algorithm for a notch filter becomes

$$\begin{aligned} e(k) &= (\, x(k) + \rho\, x(k-n-\nu)\,) - \boldsymbol{\phi}^T(k)\hat{\boldsymbol{\theta}}(k) \\ \epsilon(k) &= e(k) + \mu\ (\,\bar{c}_n - \hat{c}_n(k-1)\,) \\ \mathbf{P}(k) &= \frac{1}{\lambda}\left[\mathbf{P}(k-1) - \frac{\mathbf{P}(k-1)\boldsymbol{\psi}(k)\boldsymbol{\psi}^T(k)\mathbf{P}(k-1)}{\lambda + \boldsymbol{\psi}^T(k)\mathbf{P}(k-1)\boldsymbol{\psi}(k)}\right] \\ \hat{\boldsymbol{\theta}}(k) &= \hat{\boldsymbol{\theta}}(k-1) + \mathbf{P}(k)\boldsymbol{\psi}(k)\epsilon(k). \\ \boldsymbol{\psi}(k) &= \boldsymbol{\phi}(k) + [\,0, 0, \ldots, 0, \mu\,]^T\,. \end{aligned} \tag{3.54}$$

Table 3.6 summarizes the adaptive IIR notch filter using the final RML algorithm. When the number of filter taps is M, the required number of multiplications per iteration for the RML algorithm is given by

$$4\,M^2 + 7\,M + 6. \tag{3.55}$$

Note that the given computational complexity excluded the required computational complexity for stability test and stabilization procedure. The advantages of the adaptive IIR notch filter based on the RML algorithm are as follows.

- The RML–based adaptive notch filter performing a constrained estimation of the parameters of a general ARMA process is expected to be more accurate than an unconstrained estimator for the ARMA parameters.
- The introduction of the specific penalty function results in much sharper notches.
- The variance of the adaptive IIR notch filter parameter estimates approaches the Cramer–Rao lower bound (CRLB) under certain conditions as the number of data increases.

- The delay parameter $\nu > 1$ is useful for notch filtering of signals in correlated noise.

The disadvantages of the adaptive IIR notch filter based on the RML algorithm are as follows.

- The RML algorithm may converge to a local minimum.

Table 3.6: Summary of the Modified Recursive Maximum Likelihood (RML) Adaptive IIR Notch Filter Algorithm

Parameters	n	twice the required number of notches
	ρ	contant to control the depth of the notch
	ν	delay parameter
	α	constant to move the roots towards the origin
	β and γ	step sizes
	μ and $\bar{c}_n$	constant parameters
Initial Conditions	$\theta(0) = \phi(0) = \psi(0) = \mathbf{0}$	
	$\mathbf{P}_0 = \sigma\mathbf{I}$	with $\sigma \gg 1$
	$x(k) = 0$	for $k = -1, -2, \ldots, -n-\nu$
Computation	for $k = 1, 2, 3, \ldots$	
	$\epsilon(k) = x(k) + \rho\, x(k-n-\nu) - \phi^T(k)\hat{\theta}(k-1) + \mu\ (\bar{c}_n - \hat{c}_n(k-1))$	
	$\mathbf{P}(k) = \frac{1}{\lambda}\left[\mathbf{P}(k-1) - \frac{\mathbf{P}(k-1)\psi(k)\psi^T(k)\mathbf{P}(k-1)}{\lambda+\psi^T(k)\mathbf{P}(k-1)\psi(k)}\right]$	
	$\hat{\theta}(k) = \hat{\theta}(k-1) + \mathbf{P}(k)\psi(k)e(k)$	
	Check stability of $1/\hat{C}(z)$ and perform stability projections	
	$e(k) = x(k) + \rho\, x(k-n-\nu) - \phi^T(k)\hat{\theta}(k)$	
	$\tilde{e}(k) = e(k) - \sum_{i=1}^{n} \alpha^i(k)\, \hat{c}_i(k)\, \tilde{e}(k-i)$	
	$\tilde{x}(k) = x(k) - \sum_{i=1}^{n} \alpha^i(k)\, \hat{c}_i(k)\, \tilde{x}(k-i)$	
	$\phi_i(k) = -x(k-i) - \rho\, x(k-n-\nu+i) + e(k-i)$	
	$\psi_i(k) = -\tilde{x}(k-i) - \rho\, \tilde{x}(k-n-\nu+i) + \tilde{e}(k-i)$	
	$\phi(k) = [\,\phi_1(k), \phi_2(k), \ldots, \phi_n(k)\,]^T$	
	$\psi(k) = [\,\psi_1(k), \psi_2(k), \ldots, \psi_n(k)\,]^T + [\,0, \ldots, 0, \mu\,]^T$	
	$\lambda(k) = \beta\,\lambda(k-1) + (1-\beta)$	
	$\alpha(k) = \gamma\,\alpha(k-1) + (1-\gamma)$	

- The delay parameter $\nu = 0$ may lead the RML algorithm to the trivial solution.
- The delay parameter $\nu > 0$ may introduce superfluous notches.
- To ensure the convergence of the RML algorithm, the stability must be tested at each iteration.
- The numerical problems may occur for high order filters and for near pole–zero cancellations.
- A quite long data is generally required for the convergence of the algorithm.

Example 3.5

Let us consider the same SOI and interference with those in Example 3.4. When the number of data samples is 1,000, the SIR is -3 dB, and the values of the parameters and the initial conditions for the RML algorithm are given by

$$\begin{array}{lll} n = 2 & \rho = 1 & \nu = 1 \\ \mu = 0.5 & \bar{c}_n = 1 & \\ \alpha(0) = 0.8 & \lambda(0) = 0.99 & \mathbf{P}(0) = 100\,\mathbf{I} \\ \beta = 0.998, & \gamma = 0.995, & \end{array}$$

Fig. 3.9 (a) illustrates the averaged magnitude response over 32 transfer functions of the adaptive IIR notch filters using RML algorithm. Now, let us consider that there are two sinusoidal interferences given by

$$I(k) = A_1 \sin(2\pi(0.1)\,k + \phi_1) + A_2 \sin(2\pi(0.3)\,k + \phi_2)$$

where A_i's are the amplitudes and ϕ_i's are independent random initial phases uniformly distributed over $[-\pi, \pi]$. When the number of data

samples is 1,000, the SIR is -3 dB per sinusoid, and the values of the parameters and the initial conditions for the RML algorithm are the same as the previous ones except for $n = 4$, $\mu = 0$, and $\lambda(0) = 0.95$, Fig. 3.9 (b) illustrates the averaged magnitude response over 32 transfer functions of the adaptive IIR notch filters using RML algorithm.

From Fig. 3.9 (a), we observe that the magnitude response of the transfer function exhibits a notch at the corresponding interference frequency, but the DC gains are quite a low, when there is a single sinusoidal interference case. On the other hand, when the interference is the given multiple sinusoidal signal, the transfer function exhibits sharp notches at the corresponding interference frequencies and has much higher DC gains than in the single interference case.

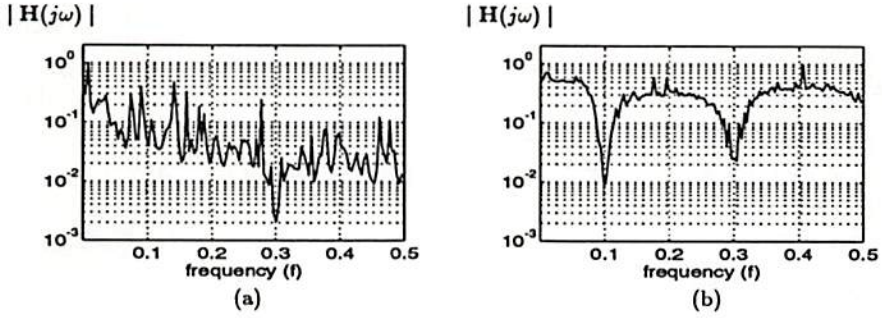

Figure 3.9: Averaged transfer functions of the RML–based adaptive IIR notch filter, (a) when there is a single sinusoidal interference and (b) when there are two sinusoidal interferences.

3.6.3 Minimal Parameter Adaptive IIR Notch Filter with Constrained Poles and Zeros

Nehorai [1985] has suggested an adaptive notch filter to mitigate multiple narrowband or sine waves in an additive broadband process. The algorithm is of recursive prediction error (RPE) form and uses a special

constrained model of IIR with a minimal number of parameters. Other techniques require $2n$ parameters, when the number of input sine waves or narrowband signal components is n. However, this method developed by Nehorai uses the minimal number of parameters equal to n. Let us consider the following IIR notch filter

$$\begin{aligned} H(z^{-1}) &= \frac{C(z^{-1})}{C(\rho\, z^{-1})} \\ &= \frac{1 + c_1 z^{-1} + \cdots + c_n z^{-n} + \cdots + c_1 z^{-2n+1} + z^{-2n}}{1 + \rho c_1 z^{-1} + \cdots + \rho^n c_n z^{-n} + \cdots + \rho^{2n-1} c_1 z^{-2n+1} + \rho^{2n} z^{-2n}} \end{aligned} \tag{3.56}$$

where $0 < \rho < 1$. Note that the coefficients of the numerator polynomial $C(z^{-1})$ have a mirror symmetric form to locate the zeros of the transfer function on the unit circle and that the poles are on the same radial lines as the zeros, but slightly displaced towards the origin. Let $\{x(k)\}$ be the input time series, then the system (error) output of the filter becomes

$$e(k) = \frac{C(z^{-1})}{C(\rho\, z^{-1})} x(k). \tag{3.57}$$

The adaptive estimation algorithm will adjust the coefficients $\{c_i\}$ so as to minimize the cost function

$$V_K = \sum_{k=1}^{K} e^2(k). \tag{3.58}$$

The predicted error in (3.57) can be rewritten as

$$e(k) = x(k) + x(k-2n) - \rho^{2n} e(k-2n) - \boldsymbol{\phi}^T(k)\boldsymbol{\theta} \tag{3.59}$$

where

$$\boldsymbol{\theta} = [\, c_1, c_2, \ldots, c_n \,]^T \tag{3.60a}$$

$$\boldsymbol{\phi}(k) = [\, \phi_1(k), \phi_2(k), \ldots, \phi_n(k) \,]^T \tag{3.60b}$$

and

$$\phi_i(k) = \begin{cases} -x(k-i) - x(k-2n+i) + \\ \qquad \rho^i e(k-i) + \rho^{2n-i} e(k-2n+i), & 1 \le i \le n-1 \\ -x(k-n) + \rho^n e(k-n), & i = n. \end{cases} \tag{3.60c}$$

After rewriting (3.57) as

$$C(\rho\, z^{-1})e(k) = C(z^{-1})x(k) \tag{3.61}$$

and differentiating both sides of (3.61) with respect to $\{c_i\}$, we can obtain, for $1 \le i \le n-1$,

$$C(\rho\, z^{-1})\frac{\partial e(k)}{\partial c_i} + \rho^i e(k-i) + \rho^{2n-i} e(k-2n+i) = x(k-i) + x(k-2n+i) \tag{3.62a}$$

and for $i = n$,

$$C(\rho\, z^{-1})\frac{\partial e(k)}{\partial c_n} + \rho^n e(k-n) = x(k-n). \tag{3.62b}$$

From these expressions we find the gradient of $e(k)$ with respect to the model $\boldsymbol{\theta}$ as

$$\boldsymbol{\psi}(k) = [\psi_1(k), \psi_2(k), \ldots, \psi_n(k)]^T = \frac{\boldsymbol{\phi}(k)}{C(\rho\, z^{-1})} \tag{3.63}$$

where

$$\psi_i(k) = -\frac{\partial e(k)}{\partial c_i}. \tag{3.64}$$

Since the filter $1/C(\rho\, z^{-1})$ is unknown, we have to replace it by its latest available estimate $1/\hat{W}(\rho\, z^{-1}, k)$. In order to improve the filter sensitivity, filter estimates, computational aspects, tracking capability, and convergence rate, the following modifications are needed on the RML algorithm based on the above modeling [Nehorai, 1985]: (i) At the beginning of the algorithm, the algorithm with too narrow notch bandwidth may not sense the presence of input sinusoidal signals; and the estimate of $C(z^{-1})$ is usually not good. For these two reasons, it is useful to apply the algorithm with a small value of ρ at the beginning of the algorithm and increase its value later. To do this procedure we can use the following time–varying $\rho(k)$ given by

$$\rho(k+1) = \gamma\, \rho(k) + (1-\gamma)\rho(\infty) \tag{3.65}$$

where $0 \leq \gamma \leq 1$ and $\rho(\infty)$ is very close to unity. (ii) Defining $\mathbf{P}(k) \triangleq \mathbf{R}^{-1}(k)$ and using the matrix inversion lemma, we can avoid the inversion of $\mathbf{R}(k)$ at every iteration. (iii) The following time–varying forgetting factor is useful in applications involving either stationary or nonstationary signals

$$\lambda(k) = \beta\,\lambda(k-1) + (1-\beta) \tag{3.66}$$

where $0 \leq \beta \leq 1$. (iv) To improve the convergence rate, it has been found to be useful to replace the prediction error by the *a posteriori* prediction error given by

$$e(k) = x(k) - \hat{x}(k \mid \boldsymbol{\phi}(k)). \tag{3.67}$$

After these modifications, the minimal parameter adaptive IIR notch filter algorithm with constrained poles and zeros is summarized in Table 3.7. When the number of filter taps is M, the required number of multiplications per iteration is given by

$$\frac{1}{2}\left(9\,M^2 + 25\,M + 10\right). \tag{3.68}$$

Ng [1987] has presented the approximate maximum likelihood (AML) algorithm to adaptively adjust the filter coefficients $\{c_i\}$. Although the AML algorithm is not asymptotically efficient algorithm, it has proven convergence property for the ARMA and the ARMAX model [Solo, 1979] and does not require on–line monitoring.

The advantages of the minimal parameter adaptive IIR notch filter with constrained poles and zeros are as follows.

- The number of parameters for the adaptive IIR notch filter is the same as the number of narrowband interferences.

- Although the RPE class of algorithms must be incorporated by stability monitoring, the model for the minimal parameter adap-

Table 3.7: Summary of the Minimal Parameter Adaptive IIR Notch Filter Algorithm with Constrained Poles and Zeros

Parameters	n	number of notches	
	ρ	contant to control the depth of the notch	
	β and γ	step sizes	
Initial Conditions	$\theta(0) = \phi(0) = \psi(0) = \mathbf{0}$		
	$\mathbf{P}_0 = \sigma\mathbf{I}$		with $\sigma \gg 1$
	$\lambda(1) = c_1$; $\rho(1) = c_2$; $\rho(\infty) = c_3$		where c_i are constants
	$x(k) = 0$		for $k = -1, -2, \ldots, -2n$

Computation for $k = 0, 1, 2, 3, \ldots$

$$e(k) = x(k) + x(k-2n) - \rho^{2n}(k)\bar{e}(k-2n) - \phi^T(k)\hat{\theta}(k-1)$$

$$\mathbf{P}(k) = \frac{1}{\lambda(k)}\left[\mathbf{P}(k-1) - \frac{\mathbf{P}(k-1)\psi(k)\psi^T(k)\mathbf{P}(k-1)}{\lambda(k)+\psi^T(k)\mathbf{P}(k-1)\psi(k)}\right]$$

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \mathbf{P}(k)\psi(k)e(k)$$

$$\bar{e}(k) = x(k) + x(k-2n) - \rho^{2n}(k)\,\bar{e}(k-2n) - \phi^T(k)\hat{\theta}(k)$$

$$\bar{e}_f(k) = \bar{e}(k) - \rho^{2n}(k)\bar{e}_f(k-2n) - \rho^n(k)\bar{e}_f(k-n)\hat{c}_n(k) - \sum_{i=1}^{n-1}\left[\rho^i(k)\bar{e}_f(k-i) + \rho^{2n-i}(k)\bar{e}_f(k-2n+i)\right]\hat{c}_i(k)$$

$$x_f(k) = x(k) - \rho^{2n}(k)x_f(k-2n) - \rho^n(k)x_f(k-n)\hat{c}_n(k) - \sum_{i=1}^{n-1}\left[\rho^i(k)x_f(k-i) + \rho^{2n-i}(k)x_f(k-2n+i)\right]\hat{c}_i(k)$$

$$\phi_i(k) = \begin{cases} -x(k-i) - x(k-2n+i) + \rho^i(k)\cdot \\ \quad \bar{e}(k-i) + \rho^{2n-1}(k)\,\bar{e}(k-2n+i), & i < n \\ -x(k-n) + \rho^n(k)\,\bar{e}(k-n), & i = n \end{cases}$$

$$\psi_i(k) = \begin{cases} -x_f(k-i) - x_f(k-2n+i) + \rho^i(k)\cdot \\ \quad \bar{e}_f(k-i) + \rho^{2n-1}(k)\,\bar{e}_f(k-2n+i), & i < n \\ -x_f(k-n) + \rho^n(k)\,\bar{e}_f(k-n), & i = n \end{cases}$$

$$\phi(k+1) = [\,\phi_1(k), \phi_2(k), \ldots, \phi_n(k)\,]^T$$

$$\psi(k+1) = [\,\psi_1(k), \psi_2(k), \ldots, \psi_n(k)\,]^T$$

$$\lambda(k+1) = \beta\,\lambda(k) + (1-\beta)$$

$$\rho(k+1) = \gamma\,\rho(k) + (1-\gamma)\,\rho(\infty)$$

tive IIR notch filter provides high stability and saves computations needed to monitor the stability of RPE algorithms.

- The variance of the estimated frequency of the narrowband or sinusoidal signal is of the same order of magnitude as the CRLB for sufficiently large data samples.

- The algorithm does not have any superfluous notches.

- The parameter estimates exhibit smoother and faster convergence than those arising in general ARMA system identification schemes.

Stoica and Nehorai [1988] have presented the performance analysis of the RML algorithm for the adaptive notch filter with constrained poles and zeros: analyses of the narrow notch property, the model stability, the bias, and the mean square error including comparison to the CRLB. The disadvantages of the minimal parameter adaptive IIR notch filter with constrained poles and zeros are as follows.

- We may encounter some realizations that give outlier performance, i.e., the estimates are not close to the typical behavior.

- The number of the outliers increases significantly as the SIR decreases.

- If the filter order is overdetermined, it is required to monitor the filter stability. And, its performance is less good than that using the correct order.

Example 3.6

Let us consider the same SOI and interference with those in Example 3.5. When the SIR is -3 dB and the values of the parameters and the initial conditions for the RML algorithm are given by

$$\begin{array}{lll} n = 1 & \gamma = 0.99 & \beta = 0.99 \\ \rho(0) = 0.8 & \rho(\infty) = 0.995 & \lambda(0) = 0.95 \\ \mathbf{P}(0) = 0.01\,\mathbf{I}, & & \end{array}$$

Fig. 3.10 (a) illustrates the averaged magnitude response over 32 transfer functions of the minimal parameter adaptive IIR notch filter with constrained poles and zeros. Now, let us consider the same two sinusoidal interferences as in the previous exmaple 3.5. When the SIR is -3 dB per sinusoid and the values of the parameters and the initial conditions for the algorithm are the same as the previous ones except for $n = 2$, Fig. 3.10 (b) illustrates the averaged magnitude response over 32 transfer functions of the minimal parameter adaptive IIR notch filter with

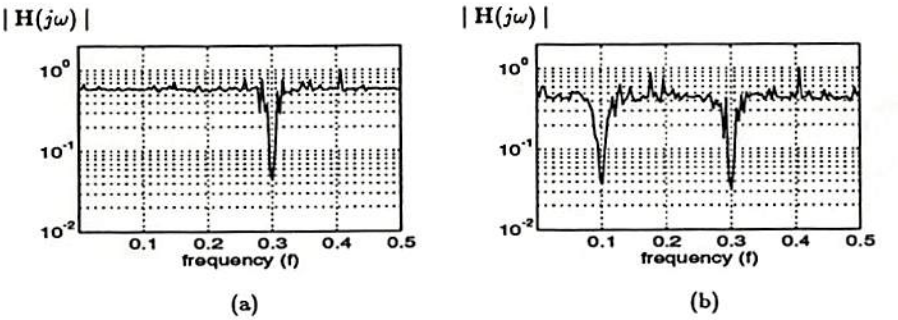

Figure 3.10: Averaged transfer functions of the minimal parameter adaptive IIR notch filter with constrained poles and zeros, (a) when there is a single sinusoidal interference and (b) when there are two sinusoidal interferences.

constrained poles and zeros.

Note that we use 1,000 data samples in both of the cases and that we test the filter stability and perform stability projections if unstable, although Nehorai [1985] argued that it would not be necessary. The averaged magnitude responses shown in Fig. 3.10 illustrate sharp notches with narrow rejection bandwidths at the corresponding interference frequencies in both the cases of single and two sinusoidal interference cases.

3.6.4 General Adaptive IIR Notch Filters

A general structure of an adaptive IIR notch filter is illustrated in Fig. 3.11. Several transfer functions of the bandpass (BP) filter, $H_{BP}(z)$, have been proposed.

Hush *et al.* [1986] have proposed

$$H_{BP}(z) = z^{-1}\,\frac{\frac{1-r^2}{1+r^2}w - (1-r^2)z^{-1}}{1 - wz^{-1} + r^2 z^{-2}} \tag{3.69}$$

where $0 \ll r < 1$ and $-2r < w < 2r$. The poles of $H_{BP}(z)$ form a complex–conjugate pair locating on a circle of radius r. The magnitude response of $H_{BP}(z)$ has unity peak gain and zero phase at an angular frequency $\gamma = \cos^{-1}(w/1+r^2)$. It guarantees that an input sinusoid with frequency γ will be exactly canceled. The optimum value of w in (3.69) that minimizes the mean squared error $\mathbf{E}\{e^2(k)\}$ is $(1+r^2)\,\cos\omega_0$. The following normalized gradient algorithm has been suggested to adjust the parameters.

$$\begin{aligned} \alpha(k) &= \frac{1-r^2}{1+r^2}\,x(k-1) + y(k-1) \\ w(k+1) &= w(k) + \mu\,\frac{\alpha(k)\,e(k)}{\psi(k)} \\ \psi(k) &= \lambda\,\psi + (1-\lambda)\,\alpha^2(k) \end{aligned} \tag{3.70}$$

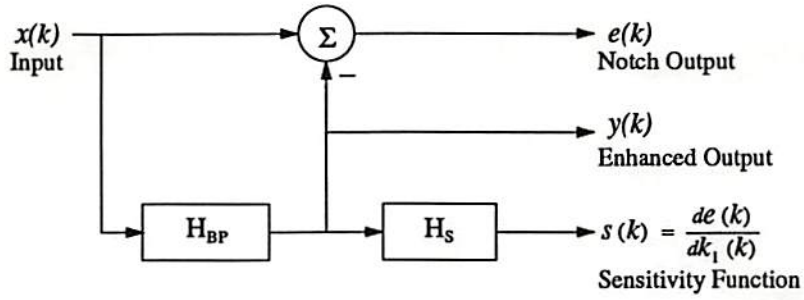

Figure 3.11: General structure of an adaptive infinite impulse response (IIR) notch filter.

where μ is a step size and $0 \ll \lambda < 1$ is a forgetting factor. Note that the reason why the normalized gradient algorithm has been suggested is that the error surface, which is a plot of $\mathbf{E}\{e^2(k)\}$ versus w, becomes less uniform and extremely flat in areas away from the minimum as r approaches 1 or as the input SIR gets smaller [Hush *et al.*, 1986].

Kwan and Martin [1989] have proposed

$$H_{BP}(z) = \frac{-k_2}{2} \frac{(1+z^{-1})(1-z^{-1})}{1-(2-k_2-k_1^2)z^{-1}+(1-k_2)z^{-2}}. \tag{3.71}$$

This filter can be realized by the transfer function of a bilinear transformed second–order filter given by

$$H_{BP}(z) = \frac{r^2-1}{2} \frac{(1+z^{-1})(1-z^{-1})}{1-2r\ \cos\omega_p\ z^{-1}+r^2z^{-2}} \tag{3.72a}$$

by letting

$$k_1 = \sqrt{1+r^2-2r\ \cos\omega_p} \qquad \text{and} \qquad k_2 = 1-r^2 \tag{3.72b}$$

where $0 \ll r < 1$ is the pole radius and ω_p is the normalized pole frequency. Then the transfer function has a peak gain of exactly unity at the frequency given by

$$\omega_{peak} = 2\ \arcsin\left(\frac{k_1}{2\sqrt{1-\frac{k_2}{2}}}\right) \tag{3.73}$$

and the transfer function of the notch filter becomes

$$H_{notch}(z) \triangleq \frac{E(z)}{X(z)} = \frac{2-k_2}{2} \frac{1 - \frac{2(2-k_2-k_1^2)}{2-k_2} z^{-1} + z^{-2}}{1-(2-k_2-k_1^2)z^{-1}+(1-k_2)z^{-2}}. \quad (3.74)$$

The notch frequency can be selected by varying k_1 and the 3 dB rejection bandwidth is kept constant by choosing a fixed value for k_2. The coefficient k_1 is updated by the following gradient algorithm

$$k_1(k+1) = k_1(k) - \mu \frac{e(k)s_s(k)}{\| s(k) \|^2} \quad (3.75)$$

where $\{s_s(k)\}$ is a sensitivity function. The sensitivity function is obtained by letting the filter output go through a sensitivity filter given by

$$H_s(z) = \frac{2k_1 z^{-1}}{1-(2-k_2-k_1^2)z^{-1}+(1-k_2)z^{-2}}. \quad (3.76)$$

The convergence properties of the adaptive IIR notch filter have been analyzed in [Petraglia *et al.*, 1990b].

3.6.5 Adaptive IIR Notch Filter Using a Sign Gradient Algorithm

Nishimura, Kim, and Hirano [1989] have proposed the following transfer function of a second–order notch filter given by

$$H_{notch}(z) = \frac{1+\alpha_0}{2} \frac{1 - 2\alpha_1(k) z^{-1} + z^{-2}}{1 - \alpha_1(k)(1+\alpha_0) z^{-1} + \alpha_0 z^{-2}} \quad (3.77)$$

where $\alpha_1(k)$ is a variable coefficient and the notch angular frequency has the relation of $\cos(\omega_0 T) = \alpha_1(k)$ with a sampling period T. The notch frequency can be changed by varying $\alpha_1(k)$ and the 3 dB rejection bandwidth can be changed by varying α_0. For an adaptive notch filter,

$\alpha_1(k)$ is chosen to minimize the the square value of notch filter output. Its gradient with respect to $\alpha_1(k)$ is given by

$$\frac{\partial e^2(k)}{\partial \alpha_1(k)} = 2\, e(k)\, \psi(k) \tag{3.78}$$

where

$$\begin{aligned} \psi(k) &= (1+\alpha_0)\, \psi(k-1)\, \alpha_1(k) - \alpha_0\, \psi(k-2) + \\ &\qquad (1+\alpha_0)\, (\, e(k-1) - x(k-1)\,) \qquad (3.79a) \\ \psi(k-i) &= \frac{\partial e(k-i)}{\partial \alpha_1(k-i)}. \qquad (3.79b) \end{aligned}$$

The differential $\psi(k)$ is generated by the transfer function given by

$$G(z) = \frac{(1+\alpha_0) z^{-1}}{1 - (1+\alpha_0)\, \alpha_1(k)\, z^{-1} + \alpha_0\, z^{-2}}. \tag{3.80}$$

The input to the transfer function is $\tilde{x}(k) = e(k) - x(k)$. For the gradient algorithm, the update equation becomes

$$\alpha_1(k+1) = \alpha_1(k) - \mu\, e(k)\, \psi(k) \tag{3.81}$$

and for the sign gradient algorithm, it becomes

$$\alpha_1(k+1) = \alpha_1(k) - \mu\, e(k)\, \mathrm{sign}(\psi(k)). \tag{3.82}$$

where μ is a constant controlling the convergence rate and sign$(\cdot)$ designates the sign of a signal. In spite of its slow convergence rate, a sign gradient algorithm is often used for real time applications. When the input is $\{x(k) = C\, \cos(\omega_0 k + \theta) + n(k)\}$ where $\{n(k)\}$ is a white Gaussian random process with zero-mean and variance σ_n^2, the gradient is given as, for the sign gradient algorithm [Nishimura *et al.*, 1989; Martin and Sun, 1986],

$$\psi(k) = -\mathrm{sign}\,(\, \sin(\omega_0 k + \theta)\,). \tag{3.83}$$

Then, a time–varying ordinary difference equation with respect to $\Delta\alpha_1(k)$ becomes

$$\begin{aligned}\Delta\alpha_1(k+1) &= (1-\mu\beta C \mid \sin(\omega_0 k+\theta) \mid)\ \Delta\alpha_1(k) \\ &\quad + \mu C\ \tilde{n}(k)\ \text{sign}(\ \sin(\omega_0 k+\theta)\ ) \qquad (3.84)\end{aligned}$$

where $\{\tilde{n}(k)\}$ is the noise term due to the input noise $\{n(k)\}$. Taking the average of $\mid \sin(\omega_0+\theta) \mid$, we obtain the following time–invariant difference equation

$$\begin{aligned}\Delta\alpha_1(k+1) &= \left(1-\frac{4\mu\,\beta\,C}{\pi}\right)\Delta\alpha_1(k) \\ &\quad + \mu\,C\,\tilde{n}(k)\,\frac{4}{\pi}\sum_{\ell=0}^{\infty}\frac{1}{2\ell+1}\sin(\ (2\ell+1)\omega_0 k+\theta\ ). \qquad (3.85)\end{aligned}$$

From the above equation we can obtain the steady–state variance of $\Delta\alpha_1(k)$ as

$$\sigma^2_{\alpha_1} = \frac{\sigma^2_n}{2\pi}\int_{-\pi}^{\pi}\left|H_1(e^{j\omega})\right|^2\ P(\omega)\ d\omega \qquad (3.86)$$

where

$$H_1(e^{j\omega}) = \frac{4\,\mu/\pi}{1-(1-2\mu C\beta/\pi)\,e^{-j\omega}} \qquad (3.87a)$$

$$P(\omega) = \frac{1}{4}\sum_{\ell=0}^{\infty}\frac{\tilde{N}(\omega-(2\ell+1)\omega_0)+\tilde{N}(\omega+(2\ell+1)\omega_0)}{2\ell+1} \qquad (3.87b)$$

with $\tilde{N}(\omega)$ as the power spectrum of $\tilde{n}(k)$. Note that the steady–state variance of $\Delta\alpha_1(k)$ for the gradient algorithm has been also presented in [Nishimura *et al.*, 1989].

The adaptive notch filter algorithm using a sign gradient is summarized in Table 3.8. The required number of multiplications per iteration for the second–order adaptive IIR filter algorithm shown in Table 3.8 is given by 10. The advantages of the adaptive IIR notch filter using sign gradient algorithm are as follows.

Table 3.8: Summary of the Adaptive IIR Notch Filter Using a Sign Gradient Algorithm

Parameters	α_0	constant to control the 3-dB rejection bandwidth
	α_1	constant to control the notch frequency
	μ	step size
Initial Conditions	α_0	3-dB rejection bandwidth
	$\alpha_1(0) = c_1$	where c_1 is a constant
Computation	for $k = 0, 1, 2, 3, \ldots$	
	$e(k)$	$= (1+\alpha_0)\,\alpha_1(k)\,e(k-1) - \alpha_0\,e(k-2)$ $+ \frac{1+\alpha_0}{2}\,[x(k) - 2\,\alpha_1(k)\,x(k-1) + x(k-2)]$
	$\psi(k)$	$= (1+\alpha_0)\,\alpha_1(k)\,\psi(k-1) - \alpha_0\,\psi(k-2)$ $+ (1+\alpha_0)\,[e(k-1) - x(k-1)]$
	$\alpha_1(k+1)$	$= \alpha_1(k-1) - \mu\,e(k)\,\text{sign}\{\psi(k)\}$

- The algorithm using sign gradients provides a simpler algorithm.
- The variance of $\Delta\alpha_1(k)$ of the sign gradient algorithm is independent of α_0, while that of the gradient algorithm increases as the value of α_0 approaches unity.
- The algorithm using sign gradients provides better stability by alleviating the problem due to the update size dependency on the power of the input signal.

The disadvantages of the adaptive IIR notch filter using sign gradient algorithm are as follows:

- The algorithm using sign gradients usually converges very slowly.
- The algorithm using sign gradients may give poor resolution as the update step size does not go to zero.

- Since the filter notch location is highly dependent on the value of $\alpha_1(k)$ related to the interference frequency and the convergence rate is very slow, we need *a priori* information of the interference frequency for good performance.

Example 3.7

Let us consider the same SOI $\{s(k)\}$ and single interference $\{I(k)\}$ with those in Examples 3.4. For all computer simulations, the following initial conditions are used:

$$\alpha_1(0) = -0.3 \quad \text{and} \quad \mu = 10^{-5}.$$

When the number of data is 1,000 and the SIR is -3 dB, Fig. 3.12 shows the average magnitude response over 32 transfer functions of the adaptive IIR notch filter using the sign gradient algorithm when $\alpha_0 = 0.92$ and $\alpha_0 = 0.7$.

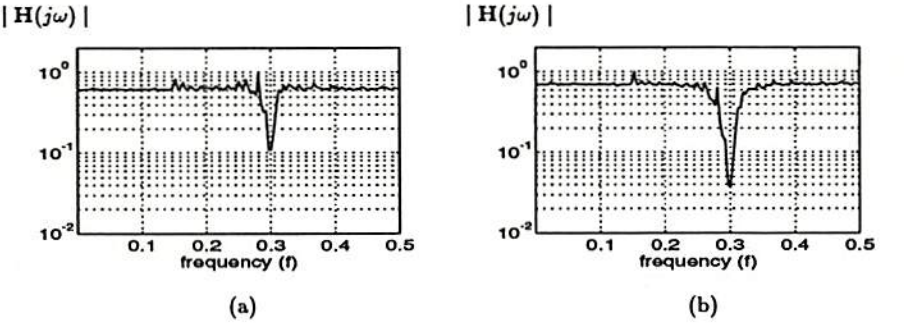

Figure 3.12: Transfer functions of the adaptive IIR notch filter using the sign gradient algorithm when (a) $\alpha_0 = 0.92$ and (b) $\alpha_0 = 0.7$.

When $\alpha_0 = 0.92$, the mean and standard deviation values of the deepest notch positions are 0.2955 and 0.0071, respectively, and when

$\alpha_0 = 0.7$, the mean is 0.2979 and the standard deviation is 0.0071. From the Fig. 3.12 we can observe that the transfer function has deeper notch and wider rejection bandwidth as the value of α_0 decreases.

3.7 Adaptive Sample Rate/Decimation Notch Filters

Strandberg, Soderstrand, and Loomis [1992] have proposed adaptive sample rate filter and adaptive decimation filter. To describe these two adaptive filters, the following assumptions are required: (i) the A/D conversion is ideal (no quantization errors), (ii) the sampling frequency f_s must be exactly four times the frequency of the narrowband interference. When the input $X(f)$ extends from DC to the highest frequency of interest f_{high}, the minimum sampling frequency f_s would be $2f_{high}$ and $f_{high}/2$ would be the lowest frequency where narrowband interference is filtered out. If we specify $X(f)$ to lie between f_{low} and $f_{high} = 2f_{low}$, sampling at $4f_{high}$ will filter components at f_{high} and sampling at $2f_{high}$ will filter components at f_{low} without aliasing the input.

3.7.1 Adaptive Sample Rate Filter

The proposed adaptive sample rate filter is illustrated in Fig. 3.13. For the adaptive sample rate filter, we need to calculate the following parameters

$$\begin{aligned} N &= \frac{f_{high} - f_{low}}{\Delta f} \\ f_{s0} &= 4\, f_{low} \\ f_{si} &= f_{s0} + 4\, \Delta f \cdot i \end{aligned} \tag{3.88}$$

where Δf is a notch width and f_{s0} and f_{sN} are the lowest and highest sample rates required. There are $N + 1$ possible sample rates and the

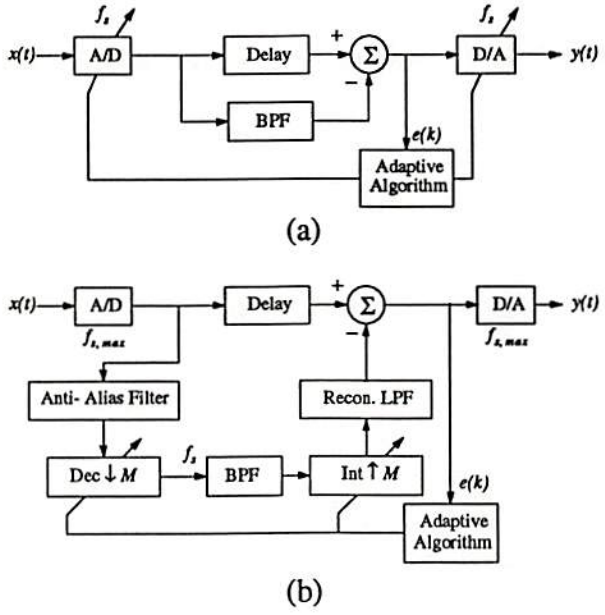

Figure 3.13: (a) Adaptive sample rate filter and (b) adaptive decimation filter.

adaptive algorithm must find the optimum sample rate among the $N+1$ possible sample rates.

The adaptive algorithm is initialized with an initial sample rate f_{sk}, which is one of the $N+1$ possible sample rates, and an initial direction $d = +1$ or $d = -1$. Then, after calculating the error, $e(0)$ at the initial sample rate, we adjust the sample rate by $i = k+d$ in (3.88) and calculate the error with the new sample rate. As long as the new error is less than the old error, we keep d the same and keep changing the sample rate. When the error increases, we use the other direction $-d$. The algorithm will search until it finds the correct sample rate to minimize the output $e(k)$.

3.7.2 Adaptive Decimation Filter

The block diagram of the adaptive decimation filter is also illustrated in Fig. 3.13. Let $f_{s,max}$ and f_s be the fixed sample rate of the A/D converter and the sample rate after decimation, respectively. We will decimate by M, where M goes from m_1 and m_2. The criteria for selecting $f_{s,max}$ and m_1 are as follows

$$\frac{f_{s,max}}{4m_1} - \frac{f_{s,max}}{4(m_1+1)} = 2\,\Delta f \tag{3.89a}$$

$$\frac{f_{s,max}}{4m_1} = f_{high}. \tag{3.89b}$$

Solving for m_1 and $f_{s,max}$ yields

$$\begin{aligned} m_1 &= \frac{f_{high} - 2\,\Delta f}{2\,\Delta f} \\ f_{s,max} &= \frac{4f_{high}\,(f_{high} - 2\,\Delta f)}{2\,\Delta f} \\ m_2 &= \frac{f_{s,max}}{4f_{low}}. \end{aligned} \tag{3.90}$$

If there is not sufficient coverage within the band, one may increase either the filter width Δf or $f_{s,max}$ and calculate m_1 and m_2.

The algorithm is initialized with an initial decimation between m_1 and m_2, and an initial direction $d = +1$ or $d = -1$. As in the adaptive sample rate algorithm, As long as the error decreases, we keep d the same and keep changing the decimation. If the error increases, we use the other direction $-d$.

The advantages of the adaptive sample rate/decimation notch filters are as follows.

- The digital filters do not need to be adapted.
- The filtering process can be accomplished using a wide variety of digital filters.

- Adaptation is easy because there is only one adaptive parameter: sample rate or decimation value.
- The low cost H/W implementations and real-time applications are available because of computationally efficient algorithms.

The disadvantages of the adaptive sample rate/decimation notch filters are as follows.

- The required ideal A/D conversion is not realizable in practice.
- We must know or estimate the interference frequency in advance to set the sampling frequency to be exactly four times that.
- The algorithms cannot be used to mitigate nonseparable multiple interferences.

3.8 Summary

In this chapter, we described several popular adaptive notch filters belonging to two categories: one with the reference input and the other without the reference input. We also presented the structure, critical design parameters, algorithm, computational complexity, advantages and disadvantages of each adaptive notch filter, and demonstrated their performance through examples related to mitigate sinusoidal interferences. Table 3.9 summarizes these all important factors of each algorithm.

The major disadvantage of these adaptive digital notch filters is that all they are based on linear algorithm and second-order statistics. Therefore, we can easily predict that their performance will be very poor when either the relationship between the interference and the reference input is nonlinear or the interference contains any nonlinear components, and also when higher-order statistics, whose order is larger than two, are

Table 3.9: Summary of Adaptive Notch FIlters

algorithms	1	2	3	4	5	6	7	8
category	first	first	first	second	second	second	second	second
realization	fir	lattice	fir	iir	iir	iir	iir	fir/iir
complexity	class A	class B	class B	class B	class B	class B	class A	class A
convergence	slow/fast	fast	fast	fast	slow	fast	slow	fast
notch depth	deep	deep	deep	deep	deep	deep	shallow	
notch width	narrow	narrow	narrow	narrow	wide	narrow	narrow	
critical design parameters	μ, z(k)	L, α_1, α_2, β_1, β_2	q	p, α, β	n, α, β, c_n, γ, μ, ν, ρ	n, ρ, β, γ	α_0, α_1(k)	f_i, f_s, Δf,

Class A: $O(M)$ multiplications per iteration
Class B: $O(M^2)$ multiplications per iteration

1: AIM filter based on the LMS algorithm
2: AIM filter based on the lattice structure
3: AIM filter based on the SER formulation
4: Constrained AIM algorithm
5: AIM algorithm based on the RML algorithm
6: Minimal parameter AIM algorithm
7: AIM algorithm based on sign gradient algorithm
8:AIM algorithm based on sample rate/decimation algorithm

involved. Since the use of the AIM filter algorithms described in this chapter is also limited, we need to consider another group of AIM filter algorithms that can be applied on nonlinear or higher–order statistics environments.

Chapter 4

Methods Based on Nonlinear Cost Functions

4.1 Introduction

In the previous chapter, we have described linear adaptive interference mitigation (AIM) algorithms based on mean–square error (MSE) or least–squares criteria. Since these AIM methods employ second–order statistics, they are limited to tracking linear variations in the second–order statistics domain. In this chapter, we consider AIM methods suited for problems where tracking of nonlinear variations is needed. This class contains algorithms having a nonlinear structure or is based on nonlinear error criteria.

When the relationship between the interference and the reference signal in Fig. 3.1 is not linear due to noise–corrupted transformations, non–invertible transformations, or nonlinear transformation, it is necessary that a nonlinear AIM method be used. Among AIM algorithms having a *nonlinear structure*, we describe the nonlinear AIM method based on Taylor series expansion; the nonlinear AIM method using a nonlinear

function of the system output sequence; and, the nonlinear AIM method based on median filters in the frequency domain. Among algorithms based on *nonlinear criteria*, we describe the AIM method based on least mean $2K$–th algorithm; the AIM method based on the constant modulus algorithm (CMA); and, the AIM method based on the criterion with memory nonlinearity (CRIMNO).

4.2 Nonlinear AIM Method Based on Taylor Series Expansion

Assuming that we employ an FIR filter structure, let its output be

$$y(k) = f(\,\mathbf{z}(k)\,) \tag{4.1}$$

where $\mathbf{z}(k) = [\,z_1(k), z_2(k), \ldots, z_M(k)\,]^T$ is the input vector of the adaptive filter at time k and $f(\cdot)$ is generally a nonlinear scalar function. The Taylor series expansion for $f(\,\mathbf{z}\,)$ about the vector $\mathbf{0}$ is given by

$$f(\,\mathbf{z}\,) = \sum_{n=0}^{\infty} \frac{1}{n!} \left(\sum_{m=1}^{M} z_m \frac{\partial}{\partial z_m} \right)^n f(\,\mathbf{z}\,) \Bigg|_{\mathbf{z}=\mathbf{0}} \tag{4.2}$$

using differential operator notation. This expression can be represented as [Coker and Simkins, 1980]

$$\begin{aligned} f(\,\mathbf{z}\,) \;=\; w^{(0)} &+ \sum_{m=1}^{M} w_m^{(1)}\, z_m + \sum_{m=1}^{M} \sum_{\ell=m}^{M} w_{m,\ell}^{(2)}\, z_m z_\ell \\ &+ \sum_{m=1}^{M} \sum_{\ell=m}^{M} \sum_{n=\ell}^{M} w_{m,\ell,n}^{(3)}\, z_m z_\ell z_n + \cdots \end{aligned} \tag{4.3a}$$

where

$$
\begin{aligned}
w^{(0)} &= f(\mathbf{0}) \\
w_m^{(1)} &= \left.\frac{\partial f(\mathbf{z})}{\partial z_m}\right|_{\mathbf{z}=\mathbf{0}} \\
w_{m,\ell}^{(2)} &= \begin{cases} \left.\frac{1}{2}\frac{\partial^2 f(\mathbf{z})}{\partial z_m^2}\right|_{\mathbf{z}=\mathbf{0}} & \text{for } m=\ell \\ \left.\frac{\partial^2 f(\mathbf{z})}{\partial z_m z_\ell}\right|_{\mathbf{z}=\mathbf{0}} & \text{for } m\neq\ell \end{cases} \qquad (4.3b) \\
&\vdots
\end{aligned}
$$

Assuming that the desired function $f(\mathbf{z})$ may be approximated sufficiently closely by truncation equation (4.3a) to L terms, we can rearrange the elements of the weighting structure to form a "super" weight vector

$$\mathbf{H} = \left[w^{(0)}, w_1^{(1)}, w_2^{(1)}, \ldots, w_M^{(1)}, w_{1,1}^{(2)}, w_{1,2}^{(2)}, \ldots, w_{M,M}^{(2)}, w_{1,1,1}^{(3)}, \ldots \right]^T \quad (4.4)$$

and a "super" argument vector

$$
\begin{aligned}
\mathbf{Z}(k) = \Big[& 1, z_1(k), z_2(k), \ldots, z_M(k), z_1^2(k), z_1(k)z_2(k), z_1(k)z_3(k), \ldots, \\
& z_1^3(k), z_1^2(k)z_2(k), z_1(k)z_2^2(k), z_1(k)z_2(k)z_3(k), \ldots \Big]^T \quad (4.5)
\end{aligned}
$$

containing all possible cross products of $\{z_m(k),\ m = 1, 2, \ldots, M\}$ up to order $L-1$. Then we can write

$$f(\mathbf{z}) \approx \hat{f}(\mathbf{z}) = \mathbf{H}^T\mathbf{Z} = y \quad (4.6a)$$

and

$$\mathbf{E}\{e^2\} = \mathbf{E}\{\,(d-y)^2\,\} = \mathbf{E}\{d^2\} - 2\mathbf{E}\{d\,\mathbf{Z}\} + \mathbf{H}^T\mathbf{E}\{\mathbf{Z}\mathbf{Z}^T\}\mathbf{H} \quad (4.6b)$$

where $\{d\}$ denotes the desired signal. Since this equation shows that the expected value of the squared error is a quadratic function of $\mathbf{H}$, the

LMS algorithm can be used for adaptation.

$$\mathbf{H}(k+1) = \mathbf{H}(k) + 2\,\mu\,e(k)\,\mathbf{Z}(k) \tag{4.7}$$

where μ is the adaptive step size and $\{e(k)\}$ denotes the system output. Then, the nonlinear AIM algorithm will converge to a solution providing the maximum output SNR possible with the given filter structure. In practice, the nonlinear AIM algorithm based on linear and quadratic terms has been widely applied [Coker and Simkins, 1980].

The only difference between the linear AIM method based on the LMS algorithm and the nonlinear AIM algorithm based on Taylor series expansion is the filter tap input vector $\mathbf{Z}(k)$. As such, by replacing the filter tap input vector in Table 3.1 with (4.5), we can easily get the nonlinear AIM method. The required number of real multiplications per iteration for the nonlinear AIM algorithm is the sum of the required complexity of the LMS algorithm and the required complexity for generating the input vector that depends on the chosen order and terms. Some of advantages of the nonlinear AIM method based on the Taylor series expansion are:

- The update equation of the nonlinear AIM algorithm uses the LMS algorithm.
- The nonlinear AIM algorithm can represent a class of nonlinear systems and be superior in some cases to conventional linear AIM algorithms.

Some of disadvantages of the nonlinear AIM method are:

- The convergence characteristics of the nonlinear AIM algorithm are not very well documented, especially, with signals whose statistics change appreciably with time.
- The nonlinear AIM algorithm using Taylor series requires more computations than the equivalent linear algorithm.

Example 4.1

The primary input consists of the sum of two signals

$$x(k) = s(k) + I(k)$$

where $\{s(k)\}$ and $\{I(k)\}$ denote the SOI and the interference, respectively. The SOI $\{s(k)\}$ is a BPSK signal satisfying $\Pr\{s_0 = -1\} = \Pr\{s_1 = 1\} = 0.5$ whose power spectrum is illustrated in Fig. 4.1 (a). The reference signal is given by

$$w(k) = 1.3\,\sin(2\pi(0.3)\,k + \pi/4).$$

We assume that the relationship between the interference and the reference signal is nonlinear, as follows

$$\begin{aligned} I(k) \;=\; & 1.25\,w^2(k) - 1.5\,w^2(k-3) - 0.95\,w^2(k-4) \\ & + 0.5\,w(k) + 0.25\,w(k-1) - 0.4\,w(k-2) - 0.75\,w(k-3). \end{aligned}$$

Then, the power spectrum of the primary input is illustrated in Fig. 4.1 (b). Note that the power spectrum has large values not only at the normalized reference frequency 0.3, but also around 0.4 due to the nonlinear relationship between the reference signal and the interference. Also, note that we use the last 256 data to estimate power spectra. We assume that there are no uncorrelated noises at the primary and reference inputs, i.e, $\{z(k) = w(k)\}$, and that the input vector of the nonlinear AIM method is

$$\mathbf{Z}(k) = \Big[\,1, z(k), z(k-1), \ldots, z(k-5), z^2(k), z^2(k-1), \ldots, z^2(k-5)\,\Big].$$

When the number of filter taps is 13 and the step size parameter μ is 0.005, the linear AIM method based on the LMS algorithm and the

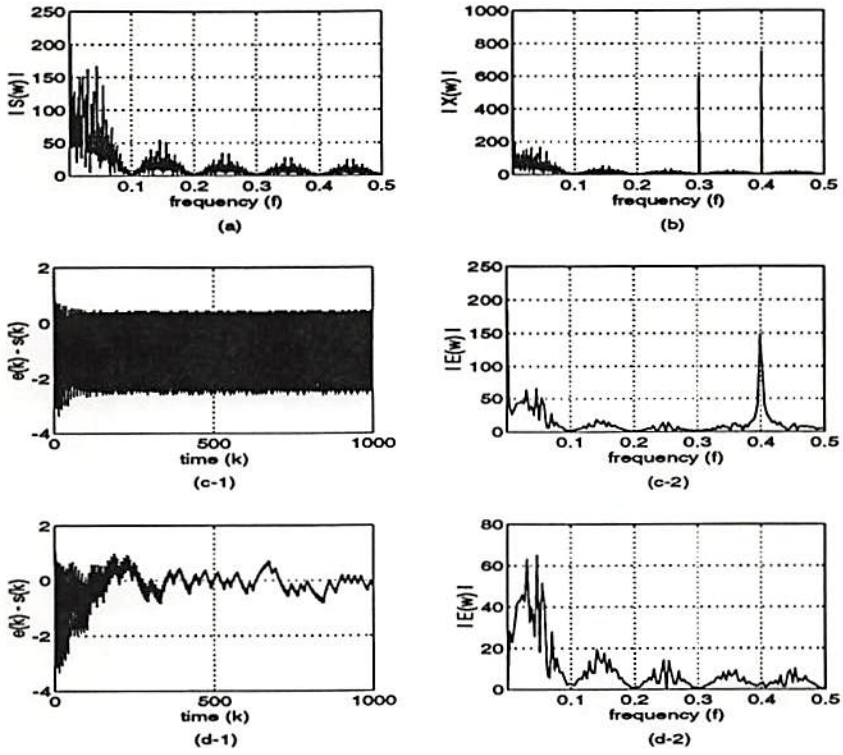

Figure 4.1: (a) Power spectrum of the SOI $\{s(k)\}$, (b) power spectrum of the received signal $\{x(k)\}$, (c–1) error curve $\{s(k) - e_\ell(k)\}$, (c–2) power spectrum of the system output $\{e_\ell(k)\}$ obtained by the linear AIM method, (d–1) error curve $\{s(k) - e_t(k)\}$, and (d–2) power spectrum of the system output $\{e_t(k)\}$ obtained by the nonlinear AIM method.

nonlinear AIM methods based on Taylor series expansion are used to mitigate the interference and recover the SOI.

Fig. 4.1 (c–1) and (c–2) show the error curve between the SOI and the system output (recovered SOI) $\{e_\ell(k)\}$ obtained by the linear AIM method and the power spectrum of the system output, respectively. We can observe large values around 0.4 in the power spectrum of the system output (Fig. 4.1 (c–2)). Large errors in time domain shown in Fig 4.1 (c–1) results from the existing interference components. Thus, it is apparent

that the linear AIM algorithm cannot effectively mitigate the interference components due to nonlinear relationship. On the other hand, Fig. 4.1 (d–1) and (d–2) illustrating the corresponding error curve and power spectrum of $\{e_t(k)\}$, respectively, obtained by the nonlinear AIM method indicate that the nonlinear AIM method based on Taylor series expansion can effectively mitigate the interference because the nonlinear method takes into account the nonlinear relationship between the interference and the reference signal.

4.3 Nonlinear Error AIM Method

Previous studies [Glover, 1977; Conolly and Su, 1986] have shown that when the AIM method based on the LMS algorithm centered at ω_1 is excited by a sinusoid of different frequency ω_2, the resulting coefficients oscillate at the sum and the difference frequencies of the two sinusoids. To reduce the time–varying steady–state behavior of the AIM filter coefficients based on LMS algorithm, Douglas and Meng [1991b] have inserted a memoryless nonlinearity into the error update equation of the stochastic gradient method, as shown in Fig. 4.2. This nonlinearity is chosen to optimize a non–mean–square error criterion matched to a given problem. The use of non–mean–square error criteria in stochastic gradient adaptation has been used to improve the performance of the algorithm [Walach and Widrow, 1984; Douglas and Meng, 1990; 1992] or to simplify the computation of the stochastic gradient estimate [Duttweiler, 1982; Xue and Liu, 1986; Douglas and Meng, 1991a]. Under slow adaptation conditions near convergence, it has been shown that the optimum error nonlinearity in stochastic gradient method is given by

$$f(e_k) = \gamma \frac{p_I'(e_k)}{p_I(e_k)}, \tag{4.8}$$

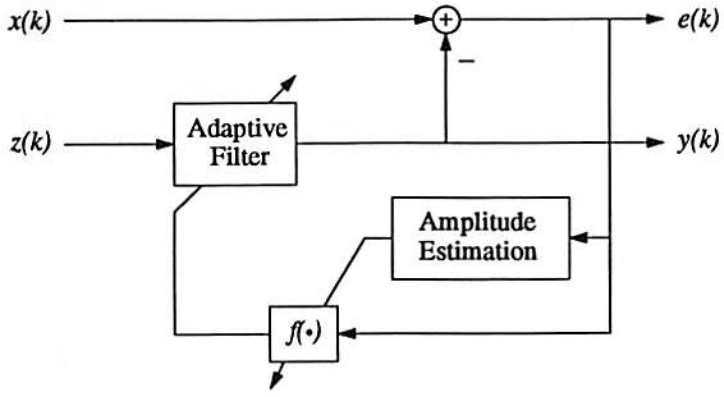

Figure 4.2: Nonlinear error AIM filter structure using a nonlinear function of the error.

where γ is a constraint parameter, the prime denotes functional differentiation, and $p_I(\cdot)$ is the probability density function of the interference $\{I(k)\}$. The stochastic gradient algorithm with this nonlinearity achieves the smallest MSE for a given adaptation rate, as the step size tends to zero [Douglas and Meng, 1990]. The reduction in excess MSE that the nonlinear algorithm provides over the LMS algorithm can be expressed as an improvement factor α defined by

$$\alpha = \frac{\text{final excess MSE of LMS algorithm}}{\text{final excess MSE of modified algorithm}} \tag{4.9a}$$

$$= \frac{\mathrm{E}^2\{f'(I_k)\}\,\mathrm{E}\{I_k^2\}}{\mathrm{E}\{f^2(I_k)\}} \tag{4.9b}$$

where the constraint parameter γ is chosen such that the adaptation speeds of the two algorithms are the same. Using the calculus of variations, it can be shown that the nonlinearity in (4.8) maximizes the performance factor in (4.9b) for all continuous nonlinearities $f(\cdot)$ defined over the region of support of the interference probability density $p_I(\cdot)$ [Douglas and Meng, 1991b].

Let the interference $\{I(k)\}$ be a pure sinusoidal signal given by

$$I(k) = C\ \sin(\omega_0 k + \theta_0). \tag{4.10}$$

Since the phase θ_0 is initially unknown, we can consider any one interference sample to be generated from a uniform random variable u_k over the range $[0, 1]$ as follows

$$I(k) \sim C\ \sin(2\pi u_k). \tag{4.11}$$

Thus, the density function of the interference becomes

$$p_I(x) = \frac{1}{\pi\sqrt{C^2 - x^2}}, \qquad |\ x\ | < C \tag{4.12}$$

and the optimum nonlinearity in (4.8) can be obtained

$$f(x) = \gamma\ \frac{x}{C^2 - x^2}, \qquad |\ x\ | < C. \tag{4.13}$$

Since the quantities $\mathbf{E}\{f'(x)\}$ and $\mathbf{E}\{f^2(x)\}$ are found to be infinite, the following nonlinearity should be considered.

$$f_\delta(x) = \begin{cases} \gamma\ \frac{x}{C^2 - x^2}, & |\ x\ | \leq (1-\delta)C \\ \gamma\ \frac{x}{C^2\ \delta(2-\delta)}, & (1-\delta)C\ <|\ x\ |<\ C. \end{cases} \tag{4.14}$$

Note that as $\delta \to 0$, the nonlinearity $f_\delta(x)$ in (4.14) approaches the nonlinearity $f(x)$ in (4.13). Using $f_\delta(x)$, we have

$$E\{f'_\delta(x)\} = \frac{2\gamma}{C^2\pi\chi^{3/2}}\left\{\frac{2}{3}\beta^3 + \beta\chi + \sqrt{\chi}\left(\frac{\pi}{2} - \sin^{-1}\beta\right)\right\} \tag{4.15a}$$

$$E\{f^2_\delta(x)\} = \frac{\gamma^2}{C^2\pi\chi^{3/2}}\left\{\frac{2}{3}\beta^3 + \beta + \frac{1}{\sqrt{\chi}}\left(\frac{\pi}{2} - \sin^{-1}\beta\right)\right\} \tag{4.15b}$$

where $\beta = 1 - \delta$ and $\chi = \delta(2 - \delta)$. Plotting α as a function of δ, it can be shown that $\lim_{\delta\to 0}\alpha = \infty$ [Douglas and Meng, 1991b]. Douglas and Meng [1991b] have chosen the nonlinearity to be

$$f(x) = \begin{cases} 0, & |\ x\ | < C \\ x, & |\ x\ | \geq C \end{cases} \tag{4.16}$$

so that adaptation for the modified algorithm is the same as LMS adaptation when the error is large. When the error is small, the deadzone about the origin effectively turns off the adaptation when the weights are close to the steady–state Wiener solution. However, the large tails of the probability density function $p_I(\cdot)$ near the edges of the deadzone keep the solution near the optimum for small but nonzero step size. Since the interference amplitude C is, in practice, generally unknown *a priori*, it is difficult to decide the width of the deadzone. To estimate the amplitude, the power of $\{e(k)\}$ in Fig. 4.2 can be estimated as follows

$$P_e(k+1) = (1-\rho)P_e(k) + \rho\, e^2(k) \tag{4.17}$$

where $0 < \rho < 1$. Using the instantaneous estimate of the interference amplitude $\sqrt{2P_e(k)}$, the width of the deadzone will be chosen. To simplify computation, the square root is replaced by a divide using a Taylor series expansion upon the right hand side of (4.17).

The nonlinear error AIM method is summarized in Table 4.1. When the number of filter taps is M, the required number of multiplications per iteration for the nonlinear error AIM algorithm is given by

$$2\,M + 4. \tag{4.18}$$

In addition to the number of multiplications, note that we need the square root calculation per each iteration. The advantages of the nonlinear error AIM method are as follows.

- It can remove the sinusoidal variations in filter coefficients that result from reducing the rejection effects at the notch frequency.
- It improves the rejection at the notch frequency and enhances tracking performance.

Table 4.1: Summary of the AIM Method Based on the Nonlinear Error Algorithm

Parameters	$\mathbf{H} = [h_1, h_2, \ldots, h_M]^T$	filter coefficient vector
	M	number of filter taps
	μ and ρ	step sizes
Initial Condition	$\mathbf{H}(0) = \mathbf{0}$	
Data at k	$x(k)$	primary input
	$\mathbf{z}(k) = [z_1(k), z_2(k), \ldots, z_M(k)]^T$	reference vector
Computation	for $k = 0, 1, 2, 3, \ldots$	
	$e(k) = x(k) - \mathbf{H}^T(k)\,\mathbf{z}(k)$	
	$P_e(k+1) = (1-\rho)P_e(k) + \rho\, e^2(k)$	
	$\mathbf{H}(k+1) = \begin{cases} \mathbf{H}(k), & \lvert e(k)\rvert < \sqrt{2P_e(k)} \\ \mathbf{H}(k) + \mu\,\mathbf{z}(k)\,e(k), & \text{otherwise} \end{cases}$	

The disadvantages of the nonlinear error AIM method are as follows.

- It is sensitive to the variations in the deadzone width.
- Its performance is sensitive to choices of values of both the parameters μ and ρ.
- It introduces a number of aperiodic components in the spectrum domain due to the nonlinear nature of the filter update equations.

Example 4.2

Let us assume that the primary input $\{x(k)\}$ consist of the sum of the sinusoidal SOI and an additive sinusoidal interference as follows.

$$x(k) = s(k) + I(k)$$

where $\{s(k)\}$ and $\{I(k)\}$ denote the SOI and the interference, respectively, given by

$$\begin{aligned} s(k) &= \sqrt{2}\,\sin(2\pi(0.4)\,k + \pi/3) \\ I(k) &= \sqrt{2}\,\sin(2\pi(0.3)\,k + \pi/4). \end{aligned}$$

The reference input is given by

$$z(k) = \sin(2\pi(0.3)\,k + \phi)$$

where ϕ is a random phase uniformly distributed over $[-\pi, \pi]$. When the number of filter taps M is 8 and the step size μ is 0.0125, Fig. 4.3 shows the averaged magnitude response over 32 transfer functions of the nonlinear error AIM filter and the averaged power spectrum $\mid E(j\omega) \mid$ over 32 estimated power spectra of the system output $\{e(k)\}$ after convergence. Note that we use the last 256 system output data samples to estimate the power spectra. An typical error curve between the system output and the SOI is also illustrated in Fig. 4.3. Note that the other step size ρ in (4.17) for estimating the amplitude of the system output is 0.0195.

It can be observed that the nonlinear error AIM method converges faster and eliminates the interference more effectively than the LMS–based AIM method shown in Fig 3.3 and that the power spectrum of the system output shown in Fig 4.3 (b) have components around 0.2 which is not related to either the SOI and the interference. This seems because of nonlinear nature mentioned before. Through computer simulations, we observe that the performance of the nonlinear error AIM algorithm is very sensitive to a choice of value of the parameter ρ. If the value of ρ is relatively small, the nonlinear error AIM algorithm performs similarly to the AIM method based on the LMS algorithm as far as excess error is concerned. If the value of ρ is relatively large, the nonlinear error AIM method might not eliminate the interference at all.

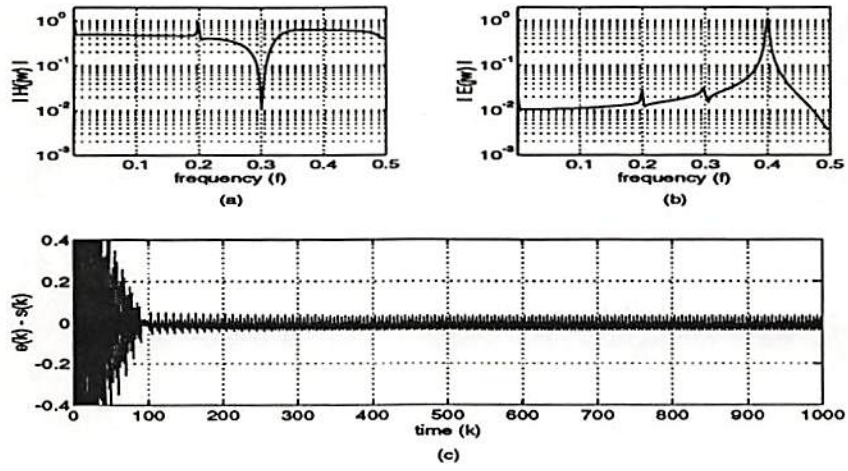

Figure 4.3: (a) Transfer function of the nonlinear error AIM filter after convergence, (b) the average of estimated and normalized power spectra of the system outputs, $\{e(k)\}$ obtained by the nonlinear error AIM algorithm, and (c) an error curve $\{e(k) - s(k)\}$.

4.4 AIM Method Based on Least Mean Fourth Algorithm

When $\{z(k)\}$ and $\{e(k)\}$ denote the input and output of a system, respectively, let us consider the general K–th order problem of minimizing

$$\mathbf{E}\{e^{2K}(k)\} \tag{4.19}$$

for $K = 1, 2, \ldots$. Extending the Widrow–Hoff algorithm to be able to minimize $\mathbf{E}\{e^{2K}(k)\}$ for an arbitrary choice of K and using the instantaneous gradient, the new filter coefficient vector update equation becomes [Walach and Widrow, 1984]

$$\mathbf{H}(k+1) = \mathbf{H}(k) - \mu\, \hat{\nabla}_{\mathbf{H}} \left(e^{2K}(k)\right) \tag{4.20}$$

where $\mathbf{H}(k)$ denotes the filter coefficient vector and $\widehat{\nabla}_{\mathbf{H}}$ denotes the instantaneous gradient operation with respect to $\mathbf{H}$ given by

$$\widehat{\nabla}_{\mathbf{H}}\left(e^{2K}(k)\right) = -2\,K\,e^{2K-1}(k)\,\mathbf{z}(k). \tag{4.21}$$

In these expressions, $\mathbf{z}(k)$ denotes the filter input vector at time k. Note that for $K = 1$, the update equation (4.20) becomes that of the LMS algorithm. The sufficient conditions for convergence of the mean and the variance, the time constants, and the misadjustment $M(K)$ for a arbitrary value of K are given in [Walach and Widrow, 1984; Nikias and Petropulu, 1993]. The least mean fourth (LMF) adaptive algorithm can be viewed as a special case of the general K–th order algorithm for $K = 2$. The LMF adaptive algorithm has the form of

$$\mathbf{H}(k+1) = \mathbf{H}(k) + \mu\,e^3(k)\,\mathbf{z}(k) \tag{4.22}$$

where μ is a step size constant.

The only difference between the general least mean K–th algorithm and the LMS algorithm is the filter update equation. Therefore, replacing the filter update equation for the LMS algorithm in Table 3.1 with (4.20), we can easily get the general K–th order algorithm. When the number of filter taps is M, the required number of real multiplications per iteration for the general least mean K–th adaptive algorithm is given by

$$2\,M + 2\,K - 1. \tag{4.23}$$

The advantages of the AIM method based on the least mean K–th order algorithm are as follows.

- When $\alpha(K) \triangleq \dfrac{M(1)}{M(K)} > 1$, the algorithm with $K > 1$ has lower misadjustment for the same speed of convergence compared to the LMS algorithm.

- The required computational complexity is insignificantly more than that of the LMS algorithm.

Its corresponding disadvantages are as follows:

- The optimal choice of K can be determined when the moments of the noise are known.
- Even when $\alpha(K) > 1$, the choice of large K is generally associated with decrease in the degree of stability of the algorithm for the given initial conditions.

Example 4.3

Consider the same SOI $\{s(k)\}$, interference $\{I(k)\}$, and reference input $\{z(k)\}$ with those in Example 4.2. When the number of filter taps M is 8 and the step size parameter μ is 0.004167, Fig. 4.4 illustrates the averaged magnitude response over 32 transfer functions of the LMF–based AIM algorithm, the averaged power spectrum over 32 estimated power spectra of the system output $\{e(k)\}$ after convergence, and a typical error curve between the system output (recovered SOI) and the SOI. Note that we use the last 256 system output data to estimate the power spectra.

As shown in this figure, the AIM method based on the LMF algorithm mitigates the sinusoidal components related to the reference input from the primary input. Compared to Fig. 3.3, the LMF adaptive algorithm can have smaller excess error when the LMF adaptive algorithm converges as fast as the LMS adaptive algorithm. The expected improvement of the LMF adaptive algorithm is almost 6 dB [Walach and

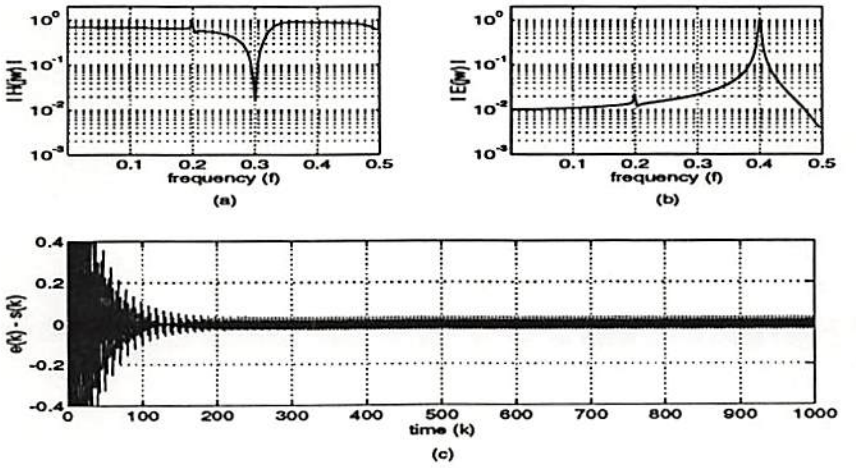

Figure 4.4: (a) Transfer function of the AIM filter based on the LMF algorithm, (b) the average of estimated and normalized power spectra of the system outputs, $\{e(k)\}$, obtained by the AIM method based on the LMF algorithm, and (c) an error curve $\{e(k) - s(k)\}$.

Widrow, 1984]. Note that its nonlinear structure of update equation introduces components around 0.2 in the power spectrum of the system output which is irrelevant to the frequencies of the SOI and interference.

4.5 AIM Methods Based on Median Filters

Since the spectrum of the SOI corrupted by narrowband interferences usually contains large amplitude impulses around the center frequency of each interference, frequency domain processing is suitable for interference suppression.

Filters based on order statistics usually have good edge preservation properties and good behavior in the presence of heavy tailed additive noises. The best known and most widely used filter based on order statistics is the *median* filter [Pitas and Venetsanopoulos, 1990]. When

the window size of a median filter is $M = 2L+1$, the conventional median filter is defined by

$$y(k) = \text{Median } \{ x(k-L), x(k-L+1), \ldots, x(k-1), x(k), x(k+1), \ldots, x(k+L-1), x(k+L) \} \quad (4.24)$$

where $\{x(k)\}$ and $\{y(k)\}$ denote the input and output of the median filter, respectively. The fundamental properties of the median filter are that it preserves signals with smooth transitions and that it eliminates impulses narrower than L. Therefore, if the spectrum of the SOI is sufficiently smooth, the conventional median filter is an ideal filter to mitigate impulsive components in frequency domain. The median filter, its modifications, and other nonlinear filters based on order statistics are presented and analyzed in [Pitas and Venetsanopoulos, 1990].

In the cases of several SOIs whose spectra are not smooth, the conditional median filter, however, should be used. The conditional median filter is a median filter that selectively eliminates impulses depending on both their relative width compared to the window size and their relative amplitude compared to adjacent amplitudes. The conditional median filter is defined as [Kasparis, 1991; Kasparis *et al.*, 1991]

$$y_c(k) = \begin{cases} x(k), & \text{when } |y(k) - x(k)| < C \\ y(k), & \text{when } |y(k) - x(k)| \geq C \end{cases} \quad (4.25)$$

where $\{y_c(k)\}$ denotes the output of the conditional median filter, $\{y(k)\}$ denotes the output of the median filter according to (4.24), and C is a threshold constant. Therefore, the conditional median filter will not affect any component that dose not meet the threshold condition and will eliminate any impulse that meets both the maximum bandwidth imposed by the window size and minimum amplitude set by the constant C.

To mitigate narrowband interferences by using a median filter scheme, one has to get the power spectrum of the received signal and pass its

values in the frequency domain through a median filter. The advantages of the AIM technique based on median filters are as follows.

- Since the median filter approach replaces a range of frequency components where interferences are expected to be with adjacent components rather than eliminating, we can reduce distortion and artifacts on the recovered SOI in time domain.
- It is not necessary to know the center frequency, bandwidth, phase, and total power of each interference in advance.
- We can selectively eliminate only those frequency components that meet the minimum power and maximum bandwidth conditions.
- There are only one or two filter parameters to determine the maximum bandwidth and minimum power requirements.
- The adaptation time is not needed.

Its corresponding disadvantages are as follows:

- It is important to choose a proper window size. A small window size can eliminate impulsive frequency components close together, but increase complexity. On the other hand, a large window size cannot eliminate impulsive frequency components close together.
- The conventional median filter is not suitable for SOIs whose power spectra are not sufficiently smooth enough.
- There is no theoretical way to choose the threshold constant C for the conditional median filter.
- A small value of the threshold constant C for the conditional median filter results in eliminating larger components of the SOI, while

a large threshold value fails to eliminate smaller components of the interference. Consequently, *a priori* information of the SOI and the interference is required for satisfying performance of the median filter.

Example 4.4

We assume that the SOI is a BPSK signal satisfying $\Pr\{s_0 = -1\} = \Pr\{s_1 = 1\} = 0.5$ and the narrowband interferences are given by

$$I_i(k) = a_i \sin(2\pi f_i k + \phi_i)$$

where

$$\begin{array}{lll} a_1 = 1.0 & a_2 = 0.7 & a_3 = 0.85 \\ f_1 = 0.1 & f_2 = 0.15 & f_3 = 0.3 \end{array}$$

and ϕ_i's are independent random variables uniformly distributed over $[-\pi, \pi]$. The estimated power spectra of the SOI and the received signal $\{x(k) = s(k) + I(k)\}$ are illustrated in Fig. 4.5 (a) and (b), respectively. When the window size of a median filter M is 7, the power spectrum of the output obtained by the conventional median filter defined by (4.24) is illustrated in Fig. 4.5 (c). When the threshold constant C is set to 200 and $M = 7$, Fig. 4.5 (d) shows the power spectrum of the output obtained by the conditional median filter defined by (4.25).

Although the conventional median filter can eliminate the impulsive components in the frequency domain, the spectrum of its output becomes over–smoothed, because the spectrum of the SOI is not sufficiently smooth. The over–smoothed spectrum of the recovered SOI results in large errors in the time domain. Thus, it is not suitable to use the conventional median filter on the given SOI. On the other hand, the spectrum of output obtained by the conditional median filter not only

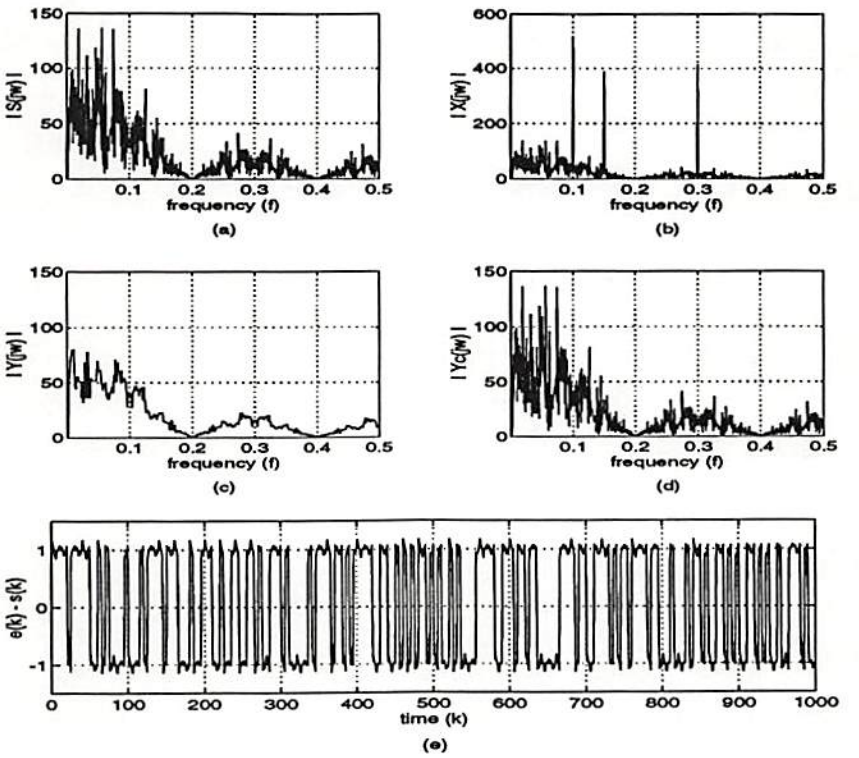

Figure 4.5: (a) Power spectrum of the SOI $\{s(k)\}$, (b) power spectrum of the received signal $\{x(k)\}$, (c) power spectrum of $\{y(k)\}$ obtained by the conventional median filter with $M = 7$, (d) power spectrum of $\{y_c(k)\}$ obtained by the conditional median filter with $M = 7$ and $C = 200$, and (e) the recovered SOI by the conditional median filter in time domain.

eliminates impulsive peaks that meet the threshold condition, but also preserves valuable SOI components that do not meet the condition. The recovered SOI obtained by the conditional median filter in time domain is illustrated in Fig. 4.5 (e) that is very similar to the given SOI. Thus, this example demonstrates that a proper choice of the threshold can improve the performance of the conditional median filter over that of the conventional median filter.

4.6 AIM Method Based on Constant Modulus Algorithm

When the SOI, $\{s(k)\}$, has a constant envelope, such as FM or QPSK signal, the constant modulus algorithm (CMA) can be used to compensate the effects of multipath and interferences [Treichler and Agee, 1983]. Since both multipath propagation and additive interference disrupt the constant envelope property of the SOI, the CMA can adaptively adjust the filter coefficients to remove the variations by sensing the received envelope variations. In the process, the CMA can suppress the various interferences.

Assuming that the input signal is available as a complex, discrete-time process and that the filter coefficients are updated with each input sample, the filter output $\{y(k)\}$ may be written as

$$y(k) = \mathbf{x}^T(k)\ \mathbf{H}(k) \tag{4.26}$$

where $\mathbf{x}(k)$ and $\mathbf{H}(k)$ are $M \times 1$ vectors of the input data and the filter coefficients, respectively, given by

$$\mathbf{x}(k) = [\, x(k), x(k-1), \cdots, x(k-M+1)\,]^T \tag{4.27a}$$

$$\mathbf{H}(k) = [\, h_0(k), h_1(k), \cdots, h_{M-1}(k)\,]^T . \tag{4.27b}$$

Note that M denotes the number of the filter taps. The objective of the CMA is to restore $\{y(k)\}$ having a constant instantaneous modulus (envelope) on the average. A general criterion is given by

$$J = d[\, F\{y(k)\}, F\{s(k)\}\,] \tag{4.28}$$

where $d(\cdot)$ and $F(\cdot)$ are length metrics to be defined for a specific algorithm. Assuming that the SOI is scaled so that $\mid s(k) \mid = 1$, the length metrics are chosen to yield the following criterion

$$J = \frac{1}{4}\, \mathrm{E}\left\{ \left[\mid y(k) \mid^2 - 1\right]^2 \right\} \tag{4.29}$$

where $\mathbf{E}\{\cdot\}$ denotes the statistical expectation. Although there are many possible approaches for adjusting the filter coefficients, a simple gradient search algorithm can be employed to minimize J for simple hardware implementations. Assume that the filter coefficient vector is updated by

$$\mathbf{H}(k+1) = \mathbf{H}(k) - \mu \, \nabla_H J_k \tag{4.30}$$

where μ is a step size and $\nabla_H J_k$ is a gradient. The adaptive algorithm is obtained by replacing the true gradient with an instantaneous gradient estimate given by

$$\hat{\nabla}_H J_k = \left(\mid y(k) \mid^2 -1 \right) \cdot y(k) \, \mathbf{x}^*(k) \tag{4.31}$$

where "*" denotes the complex conjugation. With the scalar term $\epsilon(k)$ defined by

$$\epsilon(k) = \left(\mid y(k) \mid^2 -1 \right) \cdot y(k), \tag{4.32}$$

the CMA can be compactly written as

$$\mathbf{H}(k+1) = \mathbf{H}(k) - \mu \, \epsilon(k) \mathbf{x}^*(k). \tag{4.33}$$

Note that the CMA in (4.33) is very similar to the complex version of the LMS algorithm. The CMA has been analyzed by a number of research groups: Jamali and Wood [1990] have shown the error surface analysis for the complex CMA to be quadratic in the parameter kronecker product space and derived closed–form analytical expressions similar to the Wiener type for the optimum filter coefficients. Treichler, Wolff, and Johnson [1991] have described several situations in which misconvergence of the CMA occurs and suggested that a firmer understanding is needed for its behavior in the presence of cyclostationary and/or quasi–periodic, non–white inputs. Swaminathan and Tugnait [1993] have used the estimated FIR channel impulse response to initialize the CMA–based

adaptive filter in order to avoid undesirable equilibria and saddle points. Chan, Petraglia, and Shynk [1989] have presented several efficient block implementations of the CMA in the frequency domain using convolution, filter banks, and multirate signal processing.

When we use the CMA to mitigate narrowband interferences, two problems can arise [Treichler and Agee, 1983]: (i) notch compromise and (ii) interference capture. The notch compromise means that we cannot use a deep and wide notch for interference mitigation because the complete notch process not only removes the interference but also distorts the SOI. For instance, when the SOI is a FM signal, the deeper the notch is, the more AM is induced into the FM signal. The second problem is more serious. When both the SOI and the interference have constant envelopes and are spectrally non–overlapping, it is possible to find two different filter solutions, one which suppresses the interference and the other which captures the interference and suppresses the SOI. Various methods have been suggested to avoid the interference capture such as the linearly constrained constant modulus (LCCM) approach [Rude and Griffiths, 1989; 1990], approaches using both by properly initialization and by adding a term which penalizes "un–needed" gains in the impulsive response [Gooch and Daellenbach, 1989], and approaches through whitening [Treichler and Larimore, 1985]. There are a lot of versions of the CMA for interference mitigation through modifications and hybrid structures suggested by a lot of researchers: Goldberg and Iltis [1987; 1988], Satorius *et al.* [1988], and Mendoza, Reed, Hsia, Agee [1989a; 1989b]. While the CMA has been developed for constant modulus signals, it has been shown that the algorithm is useful for a wider range of transmitted signals [Godard, 1980; Wolff *et al.*, 1988].

The AIM method based on the CMA is summarized in Table 4.2. Note that the initial condition of the filter coefficient vector $\mathbf{H}(0)$ is not

a zero vector as usual, because the adaptation process will stop when $\mathbf{H}(k) = \mathbf{0}$. So, it is important not to use the traditional all–zero initial vector for the CMA. When the number of filter taps is M, the required number of real multiplications per iteration for the CMA is given by

$$2\,M + 4. \tag{4.34}$$

The advantages of the AIM method based on the CMA are as follows.

- The required computational complexity is very low, because the CMA is similar to the complex version of the LMS–based algorithm.

- In the vicinity of convergence, the CMA filter will converge to the Wiener filter.

Table 4.2: Summary of the AIM Method Based on the Constant Modulus Algorithm (CMA)

Parameters	$\mathbf{H} = [\,h_1, h_2, \ldots, h_M\,]^T$	filter coefficient vector
	M	number of filter taps
	μ	step size
Initial Condition	$\mathbf{H}(0) = [\,1, 0, 0, \ldots, 0\,]^T$	
Data at k	$\mathbf{x}(k) = [\,x(k), x(k-1), \ldots, x(k-M+1)\,]^T$	input vector
	$y(k)$	filter output
Computation	for $k = 0, 1, 2, 3, \ldots$	
	$y(k) = \mathbf{x}^T(k)\,\mathbf{H}(k)$	
	$\epsilon(k) = \left(\,\lvert y(k)\rvert^2 - 1\,\right)\, y(k)$	
	$\mathbf{H}(k+1) = \mathbf{H}(k) - \mu\,\epsilon(k)\,\mathbf{x}^*(k)$	

The disadvantages of the AIM method method based on the CMA are as follows.

- If the interference has a constant modulus, the CMA–based AIM filter may capture the interference rather than the desired SOI.
- There is no explicit way to prevent interference capture.

Example 4.5

We assume that the input signal $\{x(k)\}$ is given by

$$x(k) = s(k) + I(k)$$

where $\{s(k)\}$ is the SOI that is the BPSK signal satisfying $\Pr\{s_0 = -1\} = \Pr\{s_1 = 1\} = 0.5$ and $\{I(k)\}$ is the narrowband interference given by

$$I(k) = 3.5 \sin(2\pi\,(0.3)\,k + \pi/4).$$

When the number of filter taps M is 31 and the step size μ is 2.5×10^{-5}, Fig. 4.6 shows the estimated power spectrum of the SOI corrupted by the interference and the transfer function of the CMA–based AIM filter. Note that we use 512 input data for estimating the power spectrum and that we use the filter coefficients at the last iteration for the transfer function. In addition, the error signal between the SOI and the system output $\{y(k)\}$ in time domain is shown in Fig. 4.6 (c).

This figure shows that the CMA can provide a sharp notch at the frequency of a narrowband interference and be used to mitigate a narrowband interference. Compared to other linear or nonlinear algorithms, the CMA, however, converges slowly and has quite large excess errors. Also, the performance of the CMA is sensitive to the amplitude of the interference and the choice of the step size.

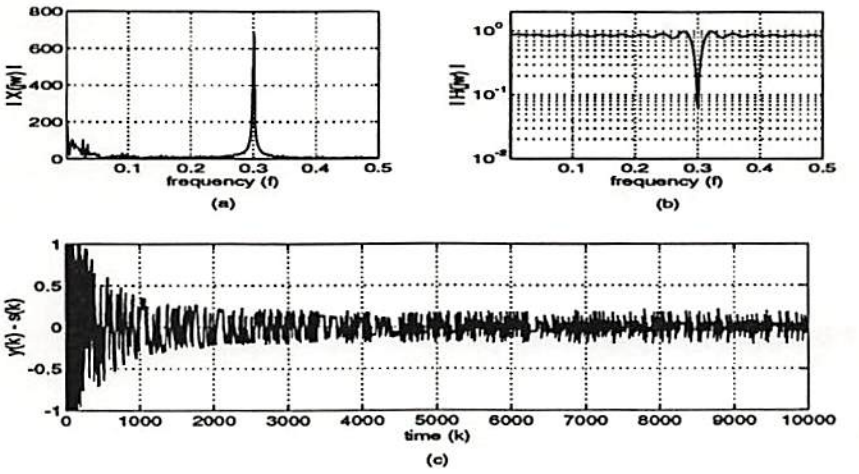

Figure 4.6: (a) Power spectrum of the input data $\{x(k)\}$, (b) the transfer function of the adaptive filter based on the CMA after convergence, and (c) error curve $\{y(k) - s(k)\}$.

4.7 AIM Method Based on the CRIMNO Algorithm

Using the fact that the transmitted data are statistically independent, Chen, Nikias, and Proakis [1991; 1992] have developed the new criterion with memory nonlinearity (CRIMNO) for blind equalization. When the transmitted SOI is considered as a statistically independent process, we can adaptively adjust the coefficients of an interference mitigation filter using the following criterion for the CRIMNO algorithm with memory size L given by

$$V^{(p)} = w_0\, J_0 + w_1\, J_1 + \cdots + w_L\, J_L \tag{4.35}$$

where

$$J_0 = \mathrm{E}\left\{\left(|\, y(k)\,|^p - R_p\right)^2\right\} \tag{4.36a}$$

$$J_\ell = |\, \mathrm{E}\{y(k)y^*(k-\ell)\}\,|^2 \tag{4.36b}$$

for $\ell = 1, 2, \ldots, L$. In the expression of (4.36a), R_p is defined as

$$R_p = \frac{\mathrm{E}\{\,|s(k)|^{2p}\,\}}{\mathrm{E}\{\,|s(k)|^{p}\,\}}. \tag{4.37}$$

To find the minimum point of the criterion, we differentiate the cost function and use the steepest descent method given by

$$\mathbf{H}(k+1) = \mathbf{H}(k) - \mu\,\frac{\partial V^{(p)}}{\partial \mathbf{H}(k)} \tag{4.38}$$

where μ is the step size and $\mathbf{H}(k)$ is the filter coefficient vector given by

$$\mathbf{H}(k) = [\,h_1(k), h_2(k), \ldots, h_{M-1}(k)\,] \tag{4.39}$$

where M is the number of filter coefficients. When $p = 2$, the filter coefficient update equation becomes [Chen, *et al.*, 1992]

$$\begin{aligned}
\mathbf{H}(k+1) \;=\; \mathbf{H}(k) - \mu\,\Big[\; & 4w_0\mathrm{E}\{\mathbf{x}^*(k)y(k)\,(|y(k)|^2 - R_2)\} \\
& + 2w_1\Big(\mathrm{E}\{\mathbf{x}^*(k-1)y(k)\}\,\mathrm{E}\{y^*(k)y(k-1)\} \\
& \qquad + \mathrm{E}\{\mathbf{x}^*(k)y(k-1)\}\,\mathrm{E}\{y(k)y^*(k-1)\}\Big) \\
& + \cdots \\
& + 2w_L\Big(\mathrm{E}\{\mathbf{x}^*(k-L)y(k)\}\,\mathrm{E}\{y^*(k)y(k-L)\} \\
& \qquad + \mathrm{E}\{\mathbf{x}^*(k)y(k-L)\}\,\mathrm{E}\{y(k)y^*(k-L)\}\Big)\;\Big]
\end{aligned} \tag{4.40}$$

where $\mathbf{x}(k)$ is the $M \times 1$ vector of the received signals constructed by

$$\mathbf{x}(k) = [\,x(k), x(k-1), \ldots, x(k-M+1)\,]. \tag{4.41}$$

Instead of the ensemble averages in (4.40), we can use the empirical averages, at each iteration, over a sliding window with length P defined as

$$E^{(e)}\{f(k)\} = \frac{1}{P}\sum_{p=1}^{P} f(k-p+1) \tag{4.42}$$

or can use single point estimates. When we replace the ensemble averages by the single point estimates, we obtain the stochastic gradient CRIMNO algorithm as follows.

$$\begin{aligned}\mathbf{H}(k+1) &= \mathbf{H}(k) - \mu \Big[4w_0 \mathbf{x}^*(k) y(k) \left(|y(k)|^2 - R_2 \right) \\ &\quad + 2w_1 \left(\mathbf{x}^*(k) y(k) \, |y(k-1)|^2 + \mathbf{x}^*(k-1) y(k-1) \, |y(k)|^2 \right) \\ &\quad + \cdots \\ &\quad + 2w_L \left(\mathbf{x}^*(k) y(k) \, |y(k-L)|^2 + \mathbf{x}^*(k-L) y(k-L) \, |y(k)|^2 \right) \Big].\end{aligned} \tag{4.43}$$

Note that the filter outputs $\{y(k-\ell), \quad \ell = 1, 2, \ldots, L\}$ can be calculated by either $y(k-\ell) = \mathbf{x}^T(k-\ell)\mathbf{H}(k-\ell)$ for the CRIMNO version I or $y(k-\ell) = \mathbf{x}^T(k-\ell)\mathbf{H}(k)$ for the CRIMNO version II which requires extensive computation time [Chen, *et al.*, 1992]. When we use a small step size, it is noticed that the difference between the two versions of the CRIMNO algorithm is negligible. Therefore, we will use the CRIMNO version I that requires less computational complexity. When we use the stochastic gradient CRIMNO algorithm, a constellation eye shrinkage problem may occur. To solve the problem, we can use an automatic gain control (AGC) unit defined as [Chen, *et al.*, 1992]

$$G(k) = \left(\frac{\mathbf{E}\{ \, |s(k)|^2 \, \}}{\mathbf{E}\{ \, |y(k)|^2 \, \}} \right)^{1/2}. \tag{4.44}$$

In (4.44), $\mathbf{E}\{ \, |y(k)|^2 \, \}$ is obtained by the following update equation.

$$\mathbf{E}\{ \, |y(k)|^2 \, \} = (1-\beta) \, \mathbf{E}\{ \, |y(k-1)|^2 \, \} + \beta \cdot |y(k)|^2 \tag{4.45}$$

where $0 < \beta \ll 1$ is the forgetting factor. There are various extensions of the CRIMNO algorithm, such as adaptive weight CRIMNO algorithm, colored CRIMNO (CCRIMNO) algorithm, and higher–order correlation CRIMNO algorithm. When the weights $\{w_\ell, \quad \ell = 0, 1, 2, \ldots, L\}$ in the adaptive weight CRIMNO algorithm are adjusted by an ad hoc

way, it has been shown that the convergence of the adaptive weight CRIMNO algorithm is faster than that of the fixed weight CRIMNO algorithm [Chen, *et al.*, 1992]. Chen and Nikias [1992] have applied the fixed–weight and adaptive–weight CRIMNO algorithms to fractionally–space blind equalization problem and shown that the fractionally–spaced CRIMNO algorithm converges faster than the synchronous CRIMNO algorithm. Tsakalides and Nikias [1993] have developed the CCRIMNO algorithm that can be used for blind deconvolution problems when the input signals are colored.

The CRIMNO–based adaptive interference mitigation method is summarized in Table 4.3. When the number of filter taps and the memory size are M and L, respectively, the required number of real multiplications per iteration for the version I CRIMNO algorithm is given by [Chen, *et al.*, 1992]

$$4\,M + 3\,L + 5 \tag{4.46a}$$

and the required real multiplications per iteration for the version II CRIMNO algorithm is given by

$$ML + 4\,M + 8\,L + 5. \tag{4.46b}$$

The advantages of the AIM method based on the CRIMNO algorithm are as follows.

- The AIM method based on the CRIMNO algorithm does not exhibit the interference capture phenomena.
- The adaptive weight CRIMNO algorithm improves the convergence speed rate.

The disadvantages of the AIM method based on the CRIMNO algorithm are as follows.

Table 4.3: Summary of the AIM Method Based on the CRIMNO Algorithm with $p = 2$

Parameters	$\mathbf{H} = [\, h_1, h_2, \ldots, h_M \,]^T$	filter coefficient vector
	M	number of filter taps
	L	memory size
Initial Condition	$\mathbf{H}(0) = [\, 1, 0, 0, \ldots, 0, 0 \,]^T$	
Data at k	$\mathbf{x}(k) = [\, x(k), x(k-1), \ldots, x(k-M+1) \,]^T$	input vector
	$y(k)$	filter output
Computation	for $k = 0, 1, 2, 3, \ldots$	
	$y(k) = \mathbf{x}^T(k)\, \mathbf{H}(k)$	
	$e_1(k) = 2\, w_0\, \lvert y(k) \rvert^2 + w_1\, \lvert y(k-1) \rvert^2 + \cdots + w_L\, \lvert y(k-L) \rvert^2 - 2\, w_0\, R_2$	
	$e_2(k) = w_1\, \mathbf{x}^*(k-1)\, y(k-1) + \cdots + w_L\, \mathbf{x}^*(k-L)\, y(k-L)$	
	$e(k) = 2\, \left(\mathbf{x}^*(k)\, y(k)\, e_1(k) + e_2(k)\, \lvert y(k) \rvert^2 \right)$	
	$\mathbf{H}(k+1) = \mathbf{H}(k) - \mu\, e(k)$	

- When the SOI is not independent but colored, we need more information of the SOI for the CCRIMNO–based algorithm.
- We need extra calculation for the AGC unit for the CRIMNO–based algorithm.

Example 4.6

Let us consider the same SOI and interference as in the previous example 4.5. When the number of filter taps M is 31 and the step size μ is 0.5×10^{-5}, Fig. 4.7 illustrates the estimated power spectrum of the SOI corrupted by the interference and the transfer function of the

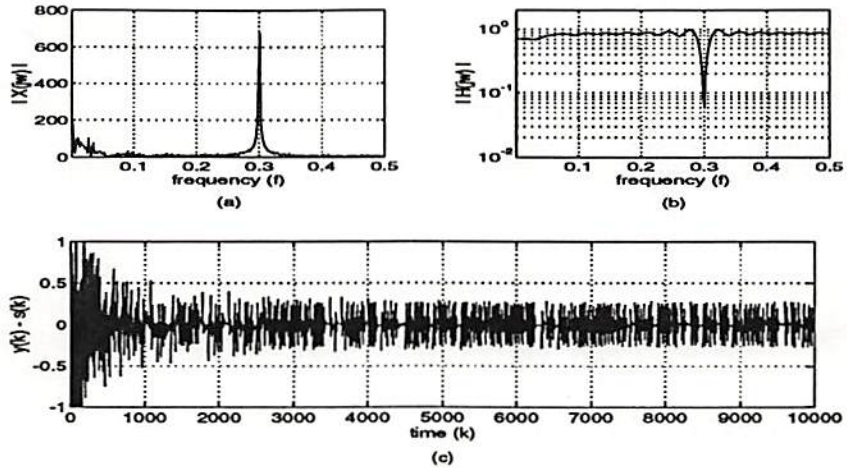

Figure 4.7: (a) Power spectrum of the input data $\{x(k)\}$, (b) the transfer function of the CRIMNO–based AIM filter with $L = 4$, and (c) error curve $\{y(k) - s(k)\}$.

CRIMNO–based adaptive filter with memory size $L = 4$. The weights for the CRIMNO algorithm are set as follows.

$$w_\ell = \begin{cases} 1, & \ell = 0 \\ 0.5 \cdot (0.7)^\ell, & \ell \neq 0. \end{cases}$$

Note that we use 512 input data for estimating the power spectrum and that we use the filter coefficients at the last iteration for the transfer function. In addition, the error signal between the SOI and the system output $\{y(k)\}$ in time domain is shown in Fig. 4.7 (c), when the gain of the AGC unit is set by

$$G(k) = \left(\frac{1}{\widehat{y^2}(k)} \right)^{1/2}.$$

where

$$\widehat{y^2}(k) = (1 - \beta)\, \widehat{y^2}(k-1) + \beta\ |y(k)|^2 .$$

with $\widehat{y^2}(0) = 1.5$ and $\beta = 0.05$.

This figure shows that the CRIMNO–based adaptive algorithm can provide a sharp notch at the frequency of a narrowband interference and be used to mitigate a narrowband interference. The performance of the AIM based on the CRIMNO is very similar to that of the AIM based on the CMA algorithm. The convergence speed is relatively slow and the excess error is relatively large.

4.8 Summary

We have described a number of nonlinear AIM methods and their algorithmic properties. For a good performance, with any nonlinear AIM method, more computational complexity and a priori knowledge of the SOI and the interference are required. For example, we need the proper order L for which the nonlinear function may be approximated sufficiently closely by truncation equation in the nonlinear AIM method based on Taylor series expansion and we need the proper size of window and the proper threshold constant in the nonlinear AIM method using a median filter. Although the AIM methods based on the CMA algorithm and the CRIMNO algorithm for blind equalization can be used to mitigate narrowband interferences, these algorithms generally converge slowly and have relative large excess error.

Chapter 5

Higher-Order Statistics-Based Methods

5.1 Introduction

In the previous chapters, we have assumed that there is no additive noises at the primary and reference input measurements because the additive interferences are usually much more stronger. However, this assumption is not valid in practice. When there exist additive noises at the given measurements which are uncorrelated from each other, the performance of an adaptive interference mitigation (AIM) method developed under the "no additive noise" assumption deteriorates substantially.

When the interference and the additive noises are assumed to be non–Gaussian and Gaussian, respectively, by addressing the AIM problem in a higher–order statistics (HOS) domain we have a certain advantage based on the fact that HOS–based techniques are blind to any kind of a Gaussian process (white or colored). Therefore, for problems where tracking of HOS variations is required or for AIM algorithms whose update equation is not affected by Gaussian noises, we are better off using

AIM algorithms based on HOS. After briefly reviewing the useful definitions and properties of cumulants and polyspectra, we describe AIM algorithms based on HOS (AIM–HOS) which belong to the first category of interference mitigation problems where a reference signal for the interference is available.

5.2 Definitions and Preliminaries on Higher–Order Statistics

Given a set of n real random variables $\{x_1, x_2, \ldots, x_n\}$, their joint cumulants of order $r = k_1 + k_2 + \cdots + k_n$ are defined as [Papoulis, 1984]

$$\text{Cum}\{ x_1^{k_1}, \ldots, x_n^{k_n} \} \triangleq (-j)^r \left. \frac{\partial^r \ln \Phi(\omega_1, \ldots, \omega_n)}{\partial \omega_1^{k_1} \cdots \partial \omega_n^{k_n}} \right|_{\omega_1 = \cdots = \omega_n = 0} \tag{5.1a}$$

where $\Phi(\omega_1, \omega_2, \ldots, \omega_n)$ is their joint characteristic function defined as

$$\Phi(\omega_1, \omega_2, \ldots, \omega_n) = \mathbf{E}\{ \exp[\, j\,(\omega_1 x_1 + \cdots + \omega_n x_n)\,]\,\} \tag{5.1b}$$

where $\mathbf{E}\{\cdot\}$ denotes the expectation operation. Let us note that the joint moments of order r of the same set of random variables are given by

$$\begin{aligned} \text{Mom}\{ x_1^{k_1}, \ldots, x_n^{k_n} \} &\triangleq \mathbf{E}\{ x_1^{k_1}\, x_2^{k_2} \cdots x_n^{k_n} \} \\ &= (-j)^r \left. \frac{\partial^r \Phi(\omega_1, \omega_2, \ldots, \omega_n)}{\partial \omega_1^{k_1} \partial \omega_2^{k_2} \cdots \partial \omega_n^{k_n}} \right|_{\omega_1 = \cdots = \omega_n = 0.} \end{aligned} \tag{5.2}$$

The general relationship between moments of $\{x_1, x_2, \ldots, x_n\}$ and joint cumulants of order n is given by

$$\begin{aligned} &\text{Cum}\{ x_1^{k_1}, x_2^{k_2}, \ldots, x_n^{k_n} \} \\ &\quad = \sum (-1)^{p-1}\, (p-1)!\, \mathbf{E}\{ \prod_{i \in s_1} x_i \}\, \mathbf{E}\{ \prod_{i \in s_2} x_i \} \cdots \mathbf{E}\{ \prod_{i \in s_p} x_i \} \end{aligned} \tag{5.3}$$

where the summation extends over all partitions $(s_1, s_2, \ldots, s_p)$, $p = 1, 2, \ldots, n$, of the set of integers $(1, 2, \ldots, n)$. When the random variables

have zero mean, then the relation becomes

$$\text{Cum}\{ x_1, x_2 \} = \text{Mom}(x_1, x_2) = \mathbf{E}\{x_1 x_2\} \tag{5.4a}$$

$$\text{Cum}\{ x_1, x_2, x_3 \} = \text{Mom}\{ x_1, x_2, x_3 \} = \mathbf{E}\{x_1 x_2 x_3\} \tag{5.4b}$$

$$\begin{aligned}\text{Cum}\{ x_1, x_2, x_3, x_4 \} = \text{Mom}\{ x_1, x_2, x_3, x_4 \} &- \mathbf{E}\{x_1 x_2\}\mathbf{E}\{x_3 x_4\} \\ &- \mathbf{E}\{x_1 x_3\}\mathbf{E}\{x_2 x_4\} - \mathbf{E}\{x_1 x_4\}\mathbf{E}\{x_2 x_3\}.\end{aligned} \tag{5.4c}$$

The cumulants have several useful properties as follows [Nikias and Petropulu, 1993]:

Property 1 *If* $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ *are constants and* $\{x_1, x_2, \ldots, x_n\}$ *are random variables, then*

$$\text{Cum}\{ \lambda_1 x_1, \lambda_2 x_2, \ldots, \lambda_n x_n \} = \left(\prod_{i=1}^{n} \lambda_i \right) \text{Cum}\{ x_1, x_2, \ldots, x_n \}.$$

Property 2 *Cumulants are symmetric in their arguments, i.e.,*

$$\text{Cum}\{ x_1, x_2, \ldots, x_n \} = \text{Cum}\{ x_{\ell_1}, x_{\ell_2}, \ldots, x_{\ell_n} \}$$

where $(\ell_1, \ell_2, \ldots, \ell_n)$ *is a permutation of* $(1, 2, \ldots, n)$.

Property 3 *If a subset of the random variables* $\{x_1, x_2, \ldots, x_n\}$ *is independent of the rest, then*

$$\text{Cum}\{ x_1, x_2, \ldots, x_n \} = 0.$$

Property 4 *If the set of random variables* $\{x_1, x_2, \ldots, x_n\}$ *are independent of the random variables* $\{y_1, y_2, \ldots, y_n\}$, *then*

$$\text{Cum}\{ x_1 + y_1, \ldots, x_n + y_n \} = \text{Cum}\{ x_1, \ldots, x_n \} + \text{Cum}\{ y_1, \ldots, y_n \}.$$

Note that in general

$$\text{Mom}\{ x_1 + y_1, \ldots, x_n + y_n \} \neq \text{Mom}\{ x_1, \ldots, x_n \} + \text{Mom}\{ y_1, \ldots, y_n \}$$

and that

$$\text{Mom}\{ x_1 + y_1, x_2, \ldots, x_n \} = \text{Mom}\{ x_1, \ldots, x_n \} + \text{Mom}\{ y_1, x_2, \ldots, x_n \}.$$

Property 5 *If the set of random variables* $\{x_1, x_2, \ldots, x_n\}$ *is jointly Gaussian, then all the information about their distribution is contained in the moments of order* $n \leq 2$. *This leads to the fact that all joint cumulants of order* $n > 2$ *are identical to zero for Gaussian random vectors.*

Suppose that the process $\{x(k), \quad k = 0, \pm 1, \pm 2, \ldots\}$ is real, strictly stationary, with n–th order cumulants sequence $c_x^n(\ell_1, \ell_2, \ldots, \ell_{n-1})$ defined as

$$c_x^n(\ell_1, \ell_2, \ldots, \ell_{n-1}) = \text{Cum}\{\, x(k), x(k+\ell_1), \ldots, x(k+\ell_{n-1})\,\}. \quad (5.5)$$

Assuming that the cumulant sequence satisfies the condition

$$\sum_{\ell_1=-\infty}^{\infty} \cdots \sum_{\ell_{n-1}=-\infty}^{\infty} | \, c_x^n(\ell_1, \ell_2, \ldots, \ell_{n-1}) \, | < \infty \quad (5.6)$$

the n–th order cumulant spectrum $C_x^n(\omega_1, \omega_2, \ldots, \omega_{n-1})$ is defined as

$$C_x^n(\omega_1, \ldots, \omega_{n-1}) = \sum_{\ell_1=-\infty}^{\infty} \cdots \sum_{\ell_{n-1}=-\infty}^{\infty} c_x^n(\ell_1, \ell_2, \ldots, \ell_{n-1}) \cdot \exp\{-j\,(\omega_1\ell_1 + \omega_2\ell_2 + \cdots + \omega_{n-1}\ell_{n-1})\} \quad (5.7)$$

where $| \, \omega_i \, | \leq \pi$, for $i = 1, 2, \ldots, n-1$ and $| \, \omega_1 + \omega_2 + \cdots + \omega_{n-1} \, | \leq \pi$. Note that the cumulant spectra is also called as higher–order spectra or polyspectra. When n is two, the cumulant spectrum becomes the power spectrum. When n is three and four, the cumulant spectra are called bispectrum and trispectrum, respectively.

For those who are not familiar to or are interested more deeply in HOS, we refer to the tutorial papers and books by [Nikias and Ragnuveer, 1987], [Mendel, 1991], [Nikias and Mendel, 1993], and [Nikias and Petropulu,1993].

5.3 AIM Algorithms Based on Higher–Order Statistics

Let us consider the first category of interference mitigation problems illustrated in Fig. 3.1 and defined as

$$x(k) = s(k) + I(k) + n_p(k) \tag{5.8a}$$

$$z(k) = w(k) + n_r(k) \tag{5.8b}$$

where $\{x(k)\}$ and $\{z(k)\}$ denote measurements of the primary and reference sensors, respectively. The signal $\{s(k)\}$ is the signal of interest (SOI) and any kind of a signal, i.e., deterministic or random, or a combination of both. The SOI is uncorrelated with the interference and the reference signal. The additive interference $\{I(k)\}$ is a *non–Gaussian* signal (narrowband or wideband). The reference signal $\{w(k)\}$ is assumed to be also *non–Gaussian* and highly correlated with the interference only. The measurement noise sources $\{n_p(k)\}$ and $\{n_r(k)\}$ are assumed to be zero–mean, white or colored Gaussian random processes. They are uncorrelated with each other and uncorrelated with the SOI, interference, and reference signal. Moreover, we assume that the relationship between the interference and reference signal can be represented by

$$I(k) = \sum_m g(k;m)w(k-m) \tag{5.9}$$

where $\{g(k;m)\}$ is either unknown or time–varying. Let $\{y(k)\}$ denote an AIM filter output given by

$$\begin{aligned} y(k) &= \sum_{m=0}^{M-1} h_m(k)\, z(k-m) \\ &= \sum_{m=0}^{M-1} h_m(k) \left[w(k-m) + n_r(k-m)\right] \end{aligned} \tag{5.10}$$

where M denotes the number of taps and $\{h_m(k),\ m = 0, 1, \ldots, M-1\}$ is the AIM filter coefficients. These filter coefficients can be represented by a vector form:

$$\mathbf{H}(k) = \Big[\, h_0(k), h_1(k), \ldots, h_{M-1}(k) \,\Big]^T. \tag{5.11}$$

Then, the AIM system output $\{e(k)\}$, which is the recovered SOI, is obtained by

$$e(k) = x(k) - y(k) = s(k) + n_e(k) \tag{5.12}$$

where $\{n_e(k)\}$ is the AIM system output noise. We can represent the system output noise as

$$n_e(k) = \left(I(k) - \sum_{m=0}^{M-1} h_m(k) w(k-m) \right) + n_p(k) - \sum_{m=0}^{M-1} h_m(k) n_r(k-m). \tag{5.13}$$

Although an AIM method can eliminate the interference completely, it is important to note that an AIM method cannot eliminate the uncorrelated noises and its system output contains terms related to both the noises and the recovered SOI.

To develop an AIM–HOS algorithm, we need to additionally assume that there exists at least one order n that *is greater than two* ($n > 2$) such that the n–th order cumulants of the reference signal are not identically zero. Under this assumption, let $c_{xz\cdots z}(\ell_1, \ell_2, \ldots, \ell_{n-1})$ denote the n–th order cross–cumulants between the primary and reference inputs and $c_{yz\cdots z}(\ell_1, \ell_2, \ldots, \ell_{n-1})$ denote the n–th order cross–cumulants between the AIM–HOS filter output and the reference input, i.e.,

$$c_{xz\cdots z}(\ell_1, \ell_2, \ldots, \ell_{n-1}) = \text{Cum}\{\, x(k), z(k+\ell_1), z(k+\ell_2), \ldots, z(k+\ell_{n-1}) \,\} \tag{5.14a}$$

$$c_{yz\cdots z}(\ell_1, \ell_2, \ldots, \ell_{n-1}) = \text{Cum}\{\, y(k), z(k+\ell_1), z(k+\ell_2), \ldots, z(k+\ell_{n-1}) \,\} \tag{5.14b}$$

where Cum$\{\cdot\}$ is the abbreviation of cumulant operator. Since the n–th order cumulants of a stationary, zero–mean, Gaussian process are identically zero, we have

$$\begin{aligned} c_{xz\cdots z}(\ell_1, \ell_2 \ldots, \ell_{n-1}) &= c_{Iw\cdots w}(\ell_1, \ell_2, \ldots, \ell_{n-1}) \\ &= \sum_m g(k;m)\, c_{w\cdots w}(m+\ell_1, m+\ell_2, \ldots, m+\ell_{n-1}) \qquad (5.15a) \\ c_{yz\cdots z}(\ell_1, \ell_2, \ldots, \ell_{n-1}) &= \\ &\sum_{m=0}^{M-1} h_m(k)\, c_{w\cdots w}(m+\ell_1, m+\ell_2, \ldots, m+\ell_{n-1}) \qquad (5.15b) \end{aligned}$$

where $c_{w\cdots w}(\cdot)$ denotes the n–th cumulants of the reference signal. Although what we can access is the reference input that is the reference signal $\{w(k)\}$ corrupted by uncorrelated noise $\{n_r(k)\}$, the higher–order cumulants $c_{w\cdots w}(\ell_1, \ell_2, \ldots, \ell_{n-1})$ becomes identical to $c_{z\cdots z}(\ell_1, \ell_2, \ldots, \ell_{n-1})$ because of the blindness of a Gaussian process in a HOS domain. Thus, we can rewrite (5.15b) as

$$\begin{aligned} c_{yz\cdots z}(\ell_1, \ell_2, \ldots, \ell_{n-1}) &= \\ &\sum_{m=0}^{M-1} h_m(k)\, c_{z\cdots z}(m+\ell_1, m+\ell_2, \ldots, m+\ell_{n-1}). \qquad (5.16) \end{aligned}$$

Thus, the cross–cumulant function $c_{yz\cdots z}(\ell_1, \ell_2, \ldots, \ell_{n-1})$ can be obtained (estimated) from the given reference input measurement and the AIM filter coefficients. If the characteristics of $\{h_m(k),\ m = 0, 1, \ldots, M-1\}$ are close enough to or the same as those of unknown $\{g(k;m)\}$, these two higher–order cumulants, $c_{xz\cdots z}(\ell_1, \ldots, \ell_{n-1})$ and $c_{yz\cdots z}(\ell_1, \ldots, \ell_{n-1})$, become identical to each other. Thus, we define the "criterion of goodness" as the sum of the squared errors between these two n–th order cumulants:

$$\begin{aligned}\xi_g &= \sum_{\ell_1}\cdots\sum_{\ell_{n-1}}\Big[c_{xz\cdots z}(\ell_1,\ell_2,\ldots,\ell_{n-1}) - c_{yz\cdots z}(\ell_1,\ell_2,\ldots,\ell_{n-1})\Big]^2 \\ &= \sum_{\ell_1}\cdots\sum_{\ell_{n-1}}\Big[c_{xz\cdots z}(\ell_1,\ell_2,\ldots,\ell_{n-1}) \\ &\qquad - \sum_{m=0}^{M-1} h_m(k)c_{z\cdots z}(m+\ell_1, m+\ell_2,\ldots,m+\ell_{n-1})\Big]^2 \end{aligned} \tag{5.17}$$

where $(\ell_1,\ell_2,\ldots,\ell_{n-1})$ may be defined to include the whole $(n-1)$–D plane $\mathcal{R}^{n-1}$. When we use a proper and reduced domain $\Gamma \subset \mathcal{R}^{n-1}$, we can simplify the criterion and get the following new criterion ξ given by

$$\begin{aligned}\xi &= \sum_{(\ell_1,\ldots,\ell_{n-1})\ \in\ \Gamma}\cdots\sum\Big[c_{xz\cdots z}(\ell_1,\ldots,\ell_{n-1}) \\ &\qquad - \sum_{m=0}^{M-1} h_m(k)c_{z\cdots z}(m+\ell_1,\ldots,m+\ell_{n-1})\Big]^2 \end{aligned} \tag{5.18a}$$

or

$$\xi = \Big(\mathbf{C}_{xz\cdots z} - \mathbf{C}_{z\cdots z}\mathbf{H}_h(k)\Big)^T\Big(\mathbf{C}_{xz\cdots z} - \mathbf{C}_{z\cdots z}\mathbf{H}_h(k)\Big) \tag{5.18b}$$

where ξ is a special case of ξ_g. In these expressions of ξ, $\mathbf{C}_{xz\cdots z}$ is an $L\times 1$ column vector and $\mathbf{C}_{z\cdots z}$ is an $L\times M$ matrix. The parameter L denotes the number of points in the chosen set Γ. *To increase the reliability of the filter coefficients we have to choose L greater than the number of taps M.* When the ℓ–th row component of vector $\mathbf{C}_{xz\cdots z}$ is $c_{xz\cdots z}(\ell_1,\ldots,\ell_{n-1})$, the ℓ–th row of $\mathbf{C}_{z\cdots z}$ becomes

$$\begin{aligned}\Big[c_{z\cdots z}(\ell_1,\ldots,\ell_{n-1}), c_{z\cdots z}(\ell_1+1,\ldots,\ell_{n-1}+1),\cdots \\ \cdots, c_{z\cdots z}(\ell_1+M-1,\ldots,\ell_{n-1}+M-1)\Big]\end{aligned} \tag{5.19}$$

where $\ell = 1,2,\ldots,L$. The vector $\mathbf{H}_h(k)$ denotes an $M\times 1$ AIM–HOS filter coefficient vector. The gradient–type algorithm is employed to develop an adaptive scheme with the gradient of the criterion given by

$$\nabla_h(k) \equiv \frac{\partial\xi}{\partial\mathbf{H}_h(k)} = 2\Big(\mathbf{C}_{z\cdots z}^T\mathbf{C}_{z\cdots z}\mathbf{H}_h(k) - \mathbf{C}_{z\cdots z}^T\mathbf{C}_{xz\cdots z}\Big). \tag{5.20}$$

Thus, the AIM filter update equation takes the form

$$\mathbf{H}_h(k+1) = \mathbf{H}_h(k) - \mu_h \, \nabla_h(k) \tag{5.21}$$

where μ_h is the step size chosen as

$$0 < \mu_h < \frac{1}{\mathrm{tr}\{\mathbf{C}^T_{z\cdots z}\mathbf{C}_{z\cdots z}\}} \tag{5.22}$$

in order to satisfy the stability requirement without knowledge of the spread of the eigenvalues of $\mathbf{C}^T_{z\cdots z}\mathbf{C}_{z\cdots z}$ [Chiang and Nikias, 1990b]. In (5.22), $\mathrm{tr}\{\cdot\}$ is the trace operator of a matrix. Since the gradient $\nabla_h(k)$ and the step size μ_h consist of the n–th order cumulants of the primary and reference inputs which are not affected by Gaussian uncorrelated noises, the update equation is independent from the uncorrelated noises. Figure 5.1 shows the block diagram structure of the AIM–HOS algorithm for mitigating the interference.

In practice, the n–th order cumulant vector $\mathbf{C}_{xz\cdots z}$ and matrix $\mathbf{C}_{z\cdots z}$ need to be substituted by their estimates $\widehat{\mathbf{C}}_{xz\cdots z}(k)$ and $\widehat{\mathbf{C}}_{z\cdots z}(k)$ at the k–th iteration. To reduce the required computational complexity, it is desired to recursively calculate the higher–order cumulant functions. Therefore, in practice, the gradient estimate becomes

$$\begin{aligned} \widehat{\nabla}_h(k) &\equiv \frac{\partial \hat{\xi}(k)}{\partial \mathbf{H}_h(k)} \\ &= 2\left(\widehat{\mathbf{C}}^T_{z\cdots z}(k)\widehat{\mathbf{C}}_{z\cdots z}(k)\mathbf{H}_h(k) - \widehat{\mathbf{C}}^T_{z\cdots z}(k)\widehat{\mathbf{C}}_{xz\cdots z}(k)\right) \end{aligned} \tag{5.23}$$

and the AIM filter update equation becomes

$$\mathbf{H}_h(k+1) = \mathbf{H}_h(k) - \mu_h(k) \, \widehat{\nabla}_h(k) \tag{5.24}$$

where

$$\mu_h(k) = \frac{\mu}{\mathrm{tr}\left\{\widehat{\mathbf{C}}^T_{z\cdots z}(k)\widehat{\mathbf{C}}_{z\cdots z}(k)\right\}}. \tag{5.25}$$

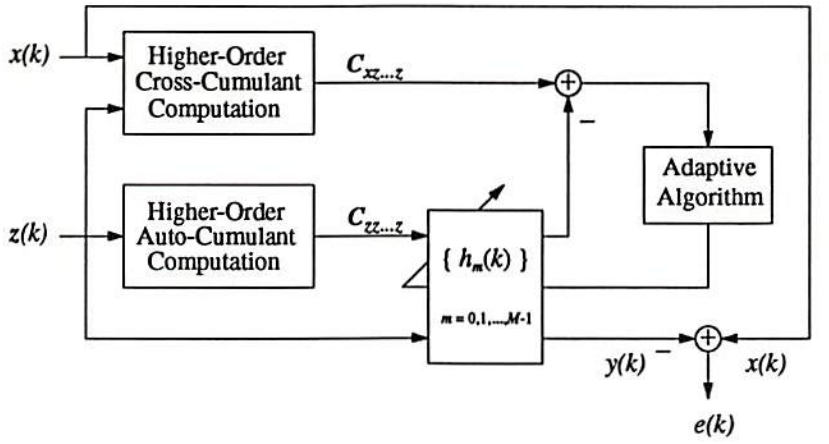

Figure 5.1: The configuration of the adaptive interference mitigation filter based on higher–order statistics (AIM–HOS).

Note that μ is a constant less than unity to ensure the algorithm stability.

The advantages of the AIM–HOS algorithms can be summarized as follows.

- Since the gradient and step size parameter of the AIM–HOS algorithm consist of higher–order cumulants in which the effects from Gaussian uncorrelated noises vanish, its filter update equation is independent from the Gaussian uncorrelated noises (in theory).
- Since the AIM–HOS algorithm uses a smoothing gradient, it takes advantage of the uncorrelation between the SOI and the reference signal.
- The AIM–HOS algorithm's performance is insensitive to the reference signal statistics and the choice of the step size parameter.

Its major disadvantages are as follows:

- Its computational complexity is very heavy. For example, when the number of taps is 8 and the number of lags for AIM–HOS algorithm

is 14, the number of required multiplication per iteration of the AIM method based on the normalized LMS algorithm is 26; and, those of AIM methods based on second–order and fourth–order statistics become 1,278 and 3,420, respectively.

- Estimate of higher–order cumulants, especially when $n > 3$, can be statistically "unstable" because of large variances.

5.4 AIM Algorithms Based on Second–Order Statistics

We briefly discuss the performance properties of the following AIM methods based on second–order statistics: the AIM algorithm based on the LMS algorithm, the AIM algorithm based on the normalized LMS algorithm, and the AIM–HOS algorithm with order $n = 2$.

5.4.1 AIM Algorithm Based on the LMS Algorithm

For a good narrow–bandwidth notch, as described in Section 3.2, the relationship among the step size (μ), the number of taps (M), and the amplitude of reference signal (A_w) should satisfy [Glover, 1977]

$$\frac{\mu M A_w^2}{4} \ll 1. \tag{5.26}$$

Since we can control only M and μ, the performance of the AIM–LMS algorithm is highly dependent on A_w which is unknown or time–varying. For a relatively small amplitude A_w, the AIM–LMS algorithm usually converges very slowly. On the other hand, for a relatively large amplitude, the AIM–LMS algorithm converges faster, but produces large excess errors after convergence. Therefore, even though there are no uncorrelated noises at the primary and reference inputs, the performance of the AIM-LMS algorithm is *problem–dependent*, i.e., its performance is very sensi-

tive to a given problem. In addition, the performance of the AIM–LMS algorithm can more substantially degrade when the interference is a sum of multiple stationary or non–stationary signals [Shin and Nikias, 1993]. This is due to the fact that there is no theoretical analysis that provides a choice for the optimal number of filter taps and the step size.

Example 5.1

Let us assume that the SOI is the BPSK signal having two states, s_0 and s_1, satisfying $\Pr\{s_0\} = \Pr\{s_1\} = 0.5$ and

$$s(k) = \begin{cases} \cos(2\pi(0.43)\,k), & \text{for } s_0 \\ -\cos(2\pi(0.43)\,k), & \text{for } s_1 \end{cases}$$

where the duration of one state is 8 samples. The reference signal is assumed to be a real–valued sine wave

$$w_i(k) = A_i \sin(2\pi f_i T\, k + \phi_i), \quad i = 1, 2, 3$$

where ϕ_i are random phases uniformly distributed over $[-\pi, \pi]$ and are independent from each other. Each interference frequency is fixed at

$$f_1 T = 0.1, \quad f_2 T = 0.25, \quad f_3 T = 0.3.$$

Then the interferences $\{I_i(k)\}$ are generated through moving average (MA) systems excited by the corresponding reference signal $\{w_i(k)\}$. The three MA coefficients equal $\mathrm{MA}_1 = [1,\ 0.1,\ -0.3,\ 0.5,\ 0.25]$, $\mathrm{MA}_2 = [1,\ 0.5,\ -0.1,\ -0.75,\ 0.41,\ 0.21,\ 0.12,\ -0.05,\ 0.01,\ -0.2,\ -0.15]$, and $\mathrm{MA}_3 = [1,\ -0.2,\ 0.2,\ 0.7,\ -0.65,\ 0.3,\ -0.25]$, respectively.

We assume that there are no additive uncorrelated noise sources and that the interference is $\{I_2(k)\}$ with $A_2 = 1$. The numbers of taps of the AIM–LMS algorithm is 16. The error curves between the SOI and

the system output obtained by the AIM–LMS algorithm is illustrated in Fig. 5.2, when $\mu = 0.0031$, 0.0047, 0.0063, and 0.0094. From this figure, it can be observed that as the value of the step size μ increases, both the convergence speed and the excess error of the AIM–LMS algorithm increase and that small difference of the step size causes big differences on the performance of the AIM–LMS algorithm.

Now, when A_2 takes different values, Fig. 5.3 illustrates the error curves between the SOI and the system outputs obtained by the AIM–LMS algorithm ($M = 16$ and $\mu = 0.0063$). From this figure, we can observe that when the number of taps and the step size parameters are fixed, the AIM–LMS algorithm is quite sensitive to the reference signal statistics. A small value of A_2 causes slow convergence, whereas a large value results in fast convergence at the expense of large excess

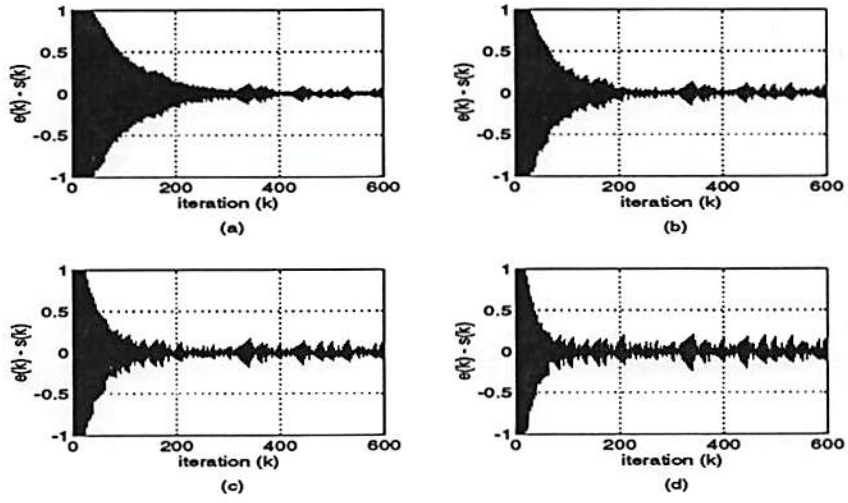

Figure 5.2: The error curves $\{e(k) - s(k)\}$ obtained by the AIM–LMS algorithm when $\{I_2\}$ with $A_2 = 1$ and the step size μ is (a) 0.0031, (b) 0.0047, (c) 0.0063, and (d) 0.0094.

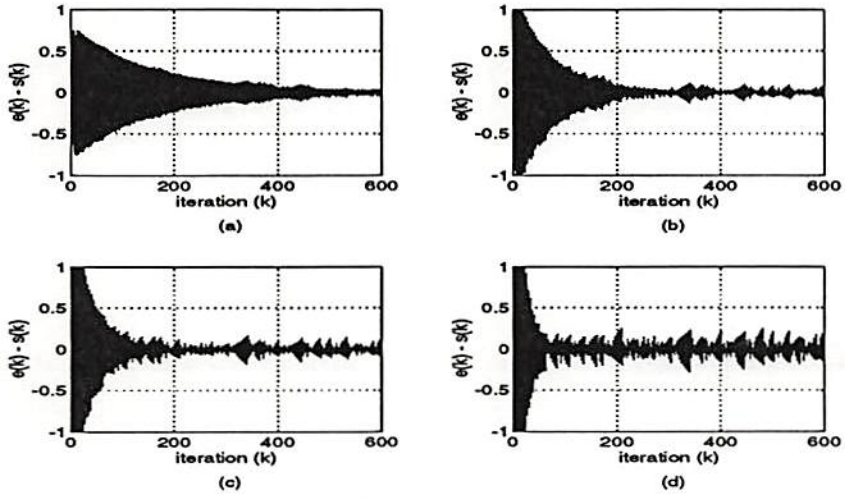

Figure 5.3: The error curves $\{e(k) - s(k)\}$ obtained by the AIM–LMS algorithm with $\mu = 0.0063$, when the reference signal amplitude is (a) 0.5, (b) 0.75, (c) 1, and (d) 1.414.

errors. When the interference is multiple or its characteristics are time–varying, the performance of the AIM–LMS algorithm is demonstrated in [Shin and Nikias, 1993],

5.4.2 AIM Algorithm Based on the Normalized LMS Algorithm

To avoid the *gradient noise amplification* due to the tap–input vector in the AIM–LMS algorithm, we can employ the AIM algorithm using the normalized LMS algorithm (AIM–NLMS). The filter update equation is given by

$$\mathbf{H}(k+1) = \mathbf{H}(k) + \frac{\mu}{\alpha + \| \mathbf{z}(k) \|^2} e(k)\, \mathbf{z}(k) \tag{5.27}$$

where $\alpha > 0$, $0 < \mu < 2$, and $\| \mathbf{z}(k) \|$ is the Euclidean norm of the tap–input vector. In (5.27), $\mathbf{H}(k)$ denotes the AIM–NLMS filter coefficient vector and $\mathbf{z}(k)$ is the tap–input vector. However, if there exist non–

negligible uncorrelated noises at the primary and reference inputs, then the AIM–NLMS algorithm coefficients $\mathbf{H}(k)$ are affected directly by both the noise sources at the primary and reference inputs, because $\mathbf{z}(k)$ contains terms related to the uncorrelated noise at the reference input and $\{e(k)\}$ contains terms related to the uncorrelated noises at the primary ad reference inputs. Thus, the AIM–NLMS algorithm, whose filter update equation is corrupted by the uncorrelated noises, can be expected not to effectively mitigate the interference. Through simulations, it is observed that when the amplitude of the reference is fixed at a constant, the performance of the AIM–NLMS algorithm is sensitive to the step size μ [Shin and Nikias, 1994]. The difference between the AIM methods based on the LMS algorithm and the NLMS algorithm is their update equation. We can get the AIM–NLMS algorithm substituting (5.27) for the update equation in Table 3.1. When the number of filter taps is M, the required number of multiplications per iteration for the AIM–NLMS algorithm is given by

$$3\,M + 2. \tag{5.28}$$

Example 5.2

Let us assume the same problem as in Example 5.1. We assume that there are no additive uncorrelated noise sources and that the interference is $\{I_2(k)\}$ with $A_w = 1$. The numbers of taps of the AIM–NLMS algorithm is 16. The error curves between the SOI and the system output obtained by the AIM–NLMS algorithm is shown in Fig. 5.4, when $\mu = 0.995$, 0.75, 0.4, and 0.1. From this figure we notice that as the step size parameter μ decreases, both the convergence speed and the excess error of the AIM–NLMS algorithm decrease; and their differences

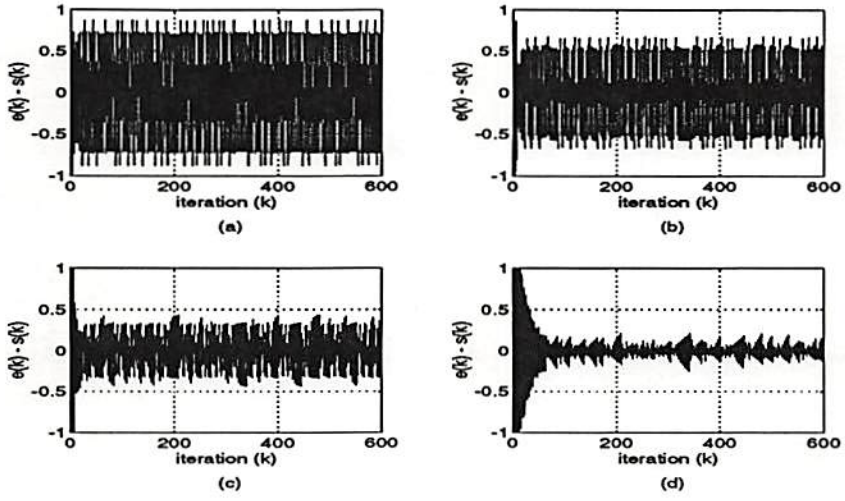

Figure 5.4: The error curves $\{e(k) - s(k)\}$ obtained by the AIM–NLMS algorithm with $M = 16$, when $\{I_2(k)\}$ with $A_2 = 1$ and the step size μ is (a) 0.995, (b) 0.75, (c) 0.4, and (d) 0.1.

are quite large. Also, we can observe that the performance of the AIM–NLMS algorithm depends on the choice of the constant α in its filter update equation. If the value of $\| \mathbf{z}(k) \|^2$ does never become close to zero, then α can be a small value. Otherwise, the value of α should be large enough to avoid division by very small number. So, it is important to note that the value of α must be carefully chosen to get good results through the AIM–NLMS algorithm because both the step size parameter and the value of α can affect the performace of the AIM–NLMS algorithm.

5.4.3 AIM–SOS Algorithm

The AIM–SOS algorithm is exactly the AIM–HOS algorithm with $n = 2$. Since the AIM–SOS algorithm is a member of the class of AIM–HOS algorithms, we can expect that the performance of the AIM–SOS

algorithm is very similar to that of the AIM–HOS algorithms with $n > 2$, when the uncorrelated noise power is negligible or very small. However, the AIM–SOS algorithm is generally not independent from the uncorrelated noises. To demonstrate this fact, let us consider the gradient for the AIM–SOS algorithm, $\nabla_s(k)$, defined as

$$\nabla_s(k) = 2 \left(\mathbf{C}_{zz}^T \mathbf{C}_{zz} \mathbf{H}_s(k) - \mathbf{C}_{zz}^T \mathbf{C}_{xz} \right) \tag{5.29}$$

where $\mathbf{H}_s(k)$ denotes an $M \times 1$ AIM–SOS filter coefficient vector. In (5.29), $\mathbf{C}_{xz}$ and $\mathbf{C}_{zz}$ denote the matrices consisting of cross–correlation functions between the primary and reference inputs and autocorrelation functions of the reference input, respectively. Since the measurement noises in the primary and reference inputs are uncorrelated, the cross–correlation function becomes

$$c_{xz}(\ell) = \mathbf{E}\{x(k)\, z(k+\ell)\} = \mathbf{E}\{I(k)\, w(k+\ell)\} \tag{5.30}$$

and is not influenced by the uncorrelated noises. On the other hand, the autocorrelation function of the reference input becomes the sum of the two autocorrelation functions of the reference signal $\{w(k)\}$ and the uncorrelated noise $\{n_r(k)\}$:

$$\begin{aligned} c_{zz}(\ell) &= \mathbf{E}\{z(k)\, z(k+\ell)\} \\ &= \mathbf{E}\{w(k)\, w(k+\ell)\} + \mathbf{E}\{n_r(k)\, n_r(k+\ell)\}. \end{aligned} \tag{5.31}$$

Therefore, it is apparent that the gradient of the AIM–SOS algorithm and consequently its filter update equation are affected by the uncorrelated noises and the performance of the AIM–SOS algorithm depends on autocorrelation functions of the uncorrelated noise at the reference input.

The AIM–SOS algorithm is summarized in Table 5.1. When the number of filter taps is M and the number of lags used in the AIM–SOS

Table 5.1: Summary of the Adaptive Interference Mitigation Algorithm Based on Second–Order Statistics (AIM–SOS) with M Taps

Parameters	M	number of taps number of equations
	L	number of equations
	μ	adaptatoin parameter
	λ	forgetting factor
Initial Conditions	$\mathbf{H}_s(0) = \mathbf{0}$	
	for $\ell = 0, 1, \ldots, N-1$	
	$\hat{c}_{zz}(0;\ell) = 0; \quad \hat{c}_{zz}(0;\ell) = 0$	
Data at k	$x(k)$	primary input
	$\mathbf{z}(k) = [\, z(k), z(k-1), \ldots, z(k-M+1)\,]^T$	reference vector
Computation	for $k = 0, 1, 2, 3, \ldots$	
	for $\ell = 0, 1, \ldots, L-1$	
	$\hat{c}_{zz}(k;\ell) = \lambda\, \hat{c}_{zz}(k-1;\ell) + x(k) z(k+\ell)$	
	for $m = 0, 1, \ldots, M-1$	
	$\hat{c}_{zz}(k;\ell+m) = \lambda\, \hat{c}_{zz}(k-1;\ell) + z(k) z(k+\ell+m)$	
	If $\lambda = 1$, $\hat{c}_{zz}(k;\ell) = \frac{1}{k}\, \hat{c}_{zz}(k;\ell)$; $\hat{c}_{zz}(k;\ell) = \frac{1}{k}\, \hat{c}_{zz}(k;\ell)$	
	Construct vector $\hat{\mathbf{C}}_{zz}(k)$ and matrix $\hat{\mathbf{C}}_{zz}(k)$	
	$e(k) = x(k) - \mathbf{H}_s^T(k)\, \mathbf{z}(k)$	
	$\hat{\nabla}_s(k) = 2\, \left[\, \hat{\mathbf{C}}_{zz}^T(k)\, \hat{\mathbf{C}}_{zz}(k)\, \mathbf{H}_s(k) - \hat{\mathbf{C}}_{zz}^T(k)\, \hat{\mathbf{C}}_{zz}(k) \,\right]$	
	$\mu_s(k) = \mu \,/\, \left(\alpha + \mathrm{tr}\left\{ \hat{\mathbf{C}}_{zz}^T(k) \hat{\mathbf{C}}_{zz}(k) \right\}\right)$	
	$\mathbf{H}_s(k+1) = \mathbf{H}_s(k) - \mu_s(k)\, \hat{\nabla}_s(k)$	

algorithm is L, the required number of multiplications per iteration for the AIM–SOS algorithm is given by

$$L\,\left(M^2 + 3M + 2\right) + M^2 + 2M + 2. \tag{5.32}$$

The advantages of the AIM–SOS algorithm are as follows:

- The AIM–SOS algorithm has the lowest computational complexity among the class of AIM–HOS algorithms.

- Its performance is similar to that of the AIM–HOS algorithms with $n > 2$, when the power of the uncorrelated noise at the reference sensor is negligible or very small.

Along with the disadvantages of the AIM–HOS algorithms outlined earlier in this chapter, an additional drawback of the AIM–SOS algorithm is :

- Its filter update equations are directly affected by the autocorrelation function of the noise at the reference input.

Example 5.3

Consider the same SOI as in the previous examples. And, let $\sigma^2_{n_p}$ and $\sigma^2_{n_r}$ denote variances of the Gaussian uncorrelated noises at the primary and reference inputs, respectively. When the interference is $\{I_2(k)\}$ with $A_2 = 0.75$ and $\sigma^2_{n_p} = \sigma^2_{n_r} = 0.001$, the error curves between the SOI and the system output obtained by the AIM–SOS algorithm ($M = 16$, $L = 30$, and $\mu = 0.99$) is illustrated in Fig. 5.5 (a). Now, consider that the interference is $\{I_1(k)\}$ with $A_1 = 0.4$ and that $\sigma^2_{n_p} = 0.001$ and $\sigma^2_{n_r} = 0.01$. Then, the error curves between the SOI and the AIM–SOS system output ($M = 8$, $L = 14$, and $\mu = 0.99$) are shown in Fig. 5.5 (b). Note that we have used $\{\hat{c}_{xz}(\ell),\ \ell = 0, 1, \ldots, L-1\}$ for the AIM–SOS algorithm.

When the power of additive noises is very small or negligible, the AIM–SOS algorithm converges fast. In fact, it has small excess error after convergence as illustrated in Fig 5.5 (a). On the other hand, when the noise power of the reference input is relatively large, the AIM–SOS algorithm performs very poorly as shown in Fig. 5.5 (b).

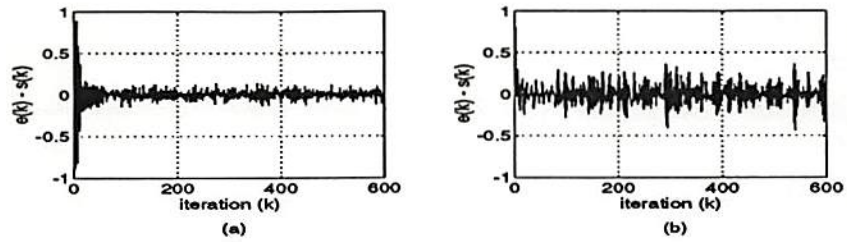

Figure 5.5: The error curves $\{e(k) - s(k)\}$ obtained by the AIM–SOS algorithm, (a) when $M = 16$, $L = 30$, $\mu = 0.99$, $\{I_2(k)\}$ with $A_2 = 0.75$, and $\sigma^2_{n_p} = \sigma^2_{n_r} = 0.001$; and (b) when $M = 8$, $L = 14$, $\mu = 0.99$, $\{I_1(k)\}$ with $A_1 = 0.4$, and $\sigma^2_{n_p} = 0.001$, and $\sigma^2_{n_r} = 0.01$.

5.5 AIM Algorithm Based on Third–Order Statistics

When the reference signal has non–zero third–order cumulants, we can use the AIM algorithms based on third–order statistics (AIM-TOS) to mitigate the interference.

If the domain $\Gamma = \{(\ell_1, \ell_2)\} \subset \mathcal{R}^2$ is chosen for a simplified criterion of goodness, then we can represent the criterion as

$$\begin{aligned}\hat{\xi}(k) &= \sum_{\ell_1}\sum_{\ell_2}\Big[\,\hat{c}_{xzz}(k;\ell_1,\ell_2) - \sum_{m=0}^{M-1} h_m(k)\,\hat{c}_{zzz}(k;m+\ell_1,m+\ell_2)\,\Big]^2 \\ &= \Big(\hat{\mathbf{C}}_{xzz}(k) - \hat{\mathbf{C}}_{zzz}(k)\mathbf{H}_t(k)\Big)^T\Big(\hat{\mathbf{C}}_{xzz}(k) - \hat{\mathbf{C}}_{zzz}(k)\mathbf{H}_t(k)\Big) \qquad (5.33)\end{aligned}$$

where $\hat{\mathbf{C}}_{xzz}(k)$ is an $L \times 1$ column vector and $\hat{\mathbf{C}}_{zzz}(k)$ is an $L \times M$ matrix at the k–th iteration. The parameter L denotes the number of equations: i.e., the number of points in the chosen set Γ. The parameter L must be larger than M. And, $\mathbf{H}_t(k)$ denotes an $M \times 1$ AIM–TOS filter coefficient vector. The structure of the two matrices is described in the previous section.

At time k, we can obtain the third–order cumulant estimate vector $\hat{\mathbf{C}}_{xzz}(k)$ and matrix $\hat{\mathbf{C}}_{zzz}(k)$ by calculating their elements as follows

$$\begin{aligned} \hat{c}_{xzz}(k;\ell_1,\ell_2) &= \frac{1}{k}\sum_{i=1}^{k} x(i)z(i+\ell_1)z(i+\ell_2) \\ \hat{c}_{zzz}(k;\ell_1,\ell_2) &= \frac{1}{k}\sum_{i=1}^{k} z(i)z(i+\ell_1)z(i+\ell_2). \end{aligned} \tag{5.34}$$

In the non–stationary interference cases, we introduce a forgetting factor, $0 < \lambda < 1$ and obtain $\hat{\mathbf{C}}_{xzz}(k)$ and $\hat{\mathbf{C}}_{zzz}(k)$ by calculating the corresponding moments as follows

$$\begin{aligned} \hat{c}_{xzz}(k;\ell_1,\ell_2) &= \sum_{i=1}^{k} \lambda^{k-i}\, x(i)z(i+\ell_1)z(i+\ell_2) \\ \hat{c}_{zzz}(k;\ell_1,\ell_2) &= \sum_{i=1}^{k} \lambda^{k-i}\, z(i)z(i+\ell_1)z(i+\ell_2). \end{aligned} \tag{5.35}$$

Dandawate and Giannakis [1989] have developed a similar scheme as an AIM–HOS algorithm using the third–order cumulants and have shown that their filter is a better estimate of the relationship between the interference and reference signals in the case where a noise source is present only at the reference input.

5.6 AIM Algorithm Based on Fourth–Order Statistics

Assuming that the interference is a sinusoidal signal whose phase is a random variable uniformly distributed over $[-\pi, \pi]$, we describe an adaptive algorithm to eliminate the interference. Since the third–order cumulants of the interference are identically zero [Swami and Mendel, 1991; Shin and Mendel, 1992], the AIM algorithm has to be based on fourth–order statistics (AIM–FOS).

For a simplified criterion of goodness for the AIM–FOS algorithm, let us choose the domain $\Gamma_1 = \{(\ell_1, \ell_2, \ell_3)\} \subset \mathcal{R}^3$ that satisfies the following conditions

$$0 \le \ell_1, \ell_2, \ell_3 \le \ell - 1, \qquad \ell_1 \ge \ell_2, \quad \text{and} \quad \ell_1 \ge \ell_3 \tag{5.36}$$

where ℓ is a positive integer greater than unity. Letting $\ell_1 - \ell_2 = i$ and $\ell_1 - \ell_3 = j$, we can represent the simplified criterion of goodness, $\hat{\xi}(k)$, as

$$\begin{aligned}\hat{\xi}(k) &= \sum_{\ell_1=0}^{\ell-1} \sum_{i=0}^{\ell_1} \sum_{j=0}^{\ell_1} \Big[\hat{C}_{xzzz}(k; \ell_1, \ell_1 - i, \ell_1 - j) \\ &\qquad - \sum_{m=0}^{M-1} h_m(k)\, \hat{c}_{zzzz}(k; m + \ell_1, m + \ell_1 - i, m + \ell_1 - j) \Big]^2 \qquad (5.37) \\ &= \Big(\hat{\mathbf{C}}_{xzzz}(k) - \hat{\mathbf{C}}_{zzzz}(k) \mathbf{H}_f(k) \Big)^T \Big(\hat{\mathbf{C}}_{xzzz}(k) - \hat{\mathbf{C}}_{zzzz}(k) \mathbf{H}_f(k) \Big)\end{aligned}$$

where $\hat{\mathbf{C}}_{xzzz}(k)$ is an $L \times 1$ column vector and $\hat{\mathbf{C}}_{zzzz}(k)$ is an $L \times M$ matrix at the k–th iteration. Note that $L = \frac{\ell(\ell+1)(2\ell+1)}{6}$. We choose L so that $L > M$. The structure of the two matrices is described in section 5.3. And, $\mathbf{H}_f(k)$ denotes an $M \times 1$ AIM–FOS filter coefficient vector. Note that when we choose a different domain Γ_2 from Γ_1, the performance of the AIM–FOS algorithm with Γ_2 may be different from that of the AIM–FOS algorithm with Γ_1. The choice of a proper domain for the AIM–HOS algorithm is an open problem.

Let us explain how to calculate the fourth–order cumulant functions at time k. For the fourth–order cumulant estimate vector $\hat{\mathbf{C}}_{xzzz}(k)$ and matrix $\hat{\mathbf{C}}_{zzzz}(k)$ by calculating their elements as follows

$$\begin{aligned}\hat{c}_{xzzz}(k; \ell_1, \ell_2, \ell_3) &= \hat{r}_{xzzz}(k; \ell_1, \ell_2, \ell_3) - \hat{r}_{xz}(k; 0, \ell_1) \hat{r}_{zz}(k; \ell_2, \ell_3) \\ &\quad - \hat{r}_{xz}(k; 0, \ell_2) \hat{r}_{zz}(k; \ell_1, \ell_3) - \hat{r}_{xz}(k; 0, \ell_3) \hat{r}_{zz}(k; \ell_1, \ell_2) \qquad (5.38a)\end{aligned}$$

$$\hat{c}_{zzzz}(k;\ell_1,\ell_2,\ell_3) = \hat{r}_{zzzz}(k;\ell_1,\ell_2,\ell_3) - \hat{r}_{zz}(k;0,\ell_1)\hat{r}_{zz}(k;\ell_2,\ell_3) - \hat{r}_{zz}(k;0,\ell_2)\hat{r}_{zz}(k;\ell_1,\ell_3) - \hat{r}_{zz}(k;0,\ell_3)\,\hat{r}_{zz}(k;\ell_1,\ell_2) \quad (5.38b)$$

where

$$\begin{aligned}
\hat{r}_{xzzz}(k;\ell_1,\ell_2,\ell_3) &= \frac{1}{k}\sum_{i=1}^{k} x(i)z(i+\ell_1)z(i+\ell_2)z(i+\ell_3) \\
\hat{r}_{zzzz}(k;\ell_1,\ell_2,\ell_3) &= \frac{1}{k}\sum_{i=1}^{k} z(i)z(i+\ell_1)z(i+\ell_2)z(i+\ell_3) \\
\hat{r}_{xz}(k;\ell_1,\ell_2) &= \frac{1}{k}\sum_{i=1}^{k} x(i+\ell_1)z(i+\ell_2) \\
\hat{r}_{zz}(k;\ell_1,\ell_2) &= \frac{1}{k}\sum_{i=1}^{k} z(i+\ell_1)z(i+\ell_2)
\end{aligned} \quad (5.39)$$

In the non–stationary interference cases, we introduce a forgetting factor, $0 < \lambda < 1$, and calculate the corresponding moments as follows

$$\begin{aligned}
\hat{r}_{xzzz}(k;\ell_1,\ell_2,\ell_3) &= \sum_{i=1}^{k} \lambda^{k-i}\, x(i)z(i+\ell_1)z(i+\ell_2)z(i+\ell_3) \\
\hat{r}_{zzzz}(k;\ell_1,\ell_2,\ell_3) &= \sum_{i=1}^{k} \lambda^{k-i}\, z(i)z(i+\ell_1)z(i+\ell_2)z(i+\ell_3) \\
\hat{r}_{xz}(k;\ell_1,\ell_2) &= \sum_{i=1}^{k} \lambda^{k-i}\, x(i+\ell_1)z(i+\ell_2) \\
\hat{r}_{zz}(k;\ell_1,\ell_2) &= \sum_{i=1}^{k} \lambda^{k-i}\, z(i+\ell_1)z(i+\ell_2).
\end{aligned} \quad (5.40)$$

The AIM–FOS algorithm is summarized in Table 5.2. Note that we introduce the constant parameter α in the step size equation in Table 5.2 to prevent the step size from being infinity (or very large) when the reference signal is very small: i.e., $\mathrm{tr}\left\{\hat{\mathbf{C}}^T_{zzzz}(k)\hat{\mathbf{C}}_{zzzz}(k)\right\}$ becomes close to zero. The computational complexity of an algorithm is an important aspect in the adaptation process. When all the signals are assumed to be real, the number of multiplications per iteration is given by

$$L\,(\,M^2 + 20M + 19\,) + M^2 + 2M + 2. \quad (5.41)$$

Table 5.2: Summary of the Adaptive Interference Mitigation Algorithm Based on Fourth–Order Statistics (AIM–FOS) with M Taps

---

Parameters

M	number of taps number of equations	
L	parameter to decide the number of equations	
λ	forgetting factor	
μ	adaptatoin parameter	

Initial Conditions

$\mathbf{H}_f(0) = \mathbf{0}$

for $\ell = 0, 1, \ldots, L-1$,

  for $i, j = 0, 1, \ldots, \ell$,

$\hat{c}_{zzzz}(0; \ell, \ell - i, \ell - j) = 0$

$\hat{c}_{zzzz}(0; \ell, \ell - i, \ell - j) = 0$

Data at k

$x(k)$	primary input
$\mathbf{z}(k) = [\, z(k), z(k-1), \ldots, z(k-M+1) \,]^T$	reference vector

Computation

for $k = 0, 1, 2, 3, \ldots$

  for $\ell = 0, 1, \ldots, L-1$ and $i, j = 0, 1, \ldots, \ell$

$\hat{r}_{zzzz}(k; \ell, \ell - i, \ell - j) = \lambda\, \hat{r}_{zzzz}(k-1; \cdot) + x(k) z(k+\ell) z(k+\ell-i) z(k+\ell-j)$

$\hat{r}_{zz}(k; \ell_1, \ell_2) = \lambda\, \hat{r}_{zz}(k-1; \cdot) + x(k+\ell_1) z(k+\ell_2)$

$\hat{r}_{zz}(k; \ell_1, \ell_2) = \lambda\, \hat{r}_{zz}(k-1; \cdot) + z(k+\ell_1) z(k+\ell_2)$

  for $m = 0, 1, \ldots, M-1$

$\hat{r}_{zzzz}(k; \ell + m, \ell - i + m, \ell - j + m) = \hat{r}_{zzzz}(k-1; \cdot) + z(k) z(k+\ell+m) \cdot z(k+\ell-i+m) z(k+\ell-j+m)$

$\hat{r}_{zz}(k; \ell_1, \ell_2) = \lambda\, \hat{r}_{zz}(k-1; \cdot) + z(k+\ell_1) z(k+\ell_2)$

If $\lambda = 1$, then $\hat{r}(k; \cdot) = \frac{1}{k}\, \hat{r}(k; \cdot)$

Calculate $\hat{c}_{zzzz}(\cdot)$ and $\hat{c}_{zzzz}(\cdot)$ with (5.38a) and (5.38b)

Construct vector $\hat{\mathbf{C}}_{zzzz}(k)$ and matrix $\hat{\mathbf{C}}_{zzzz}(k)$

$e(k) = x(k) - \mathbf{H}_f^T(k)\, \mathbf{z}(k)$

$\hat{\nabla}(k) = 2 \left[\hat{\mathbf{C}}_{zzzz}^T(k)\, \hat{\mathbf{C}}_{zzzz}(k)\, \mathbf{H}(k) - \hat{\mathbf{C}}_{zzzz}^T(k)\, \hat{\mathbf{C}}_{zzzz}(k) \right]$

$\mu(k) = \mu \,/\, \left(\alpha + \mathrm{tr} \left\{ \hat{\mathbf{C}}_{zzzz}^T(k) \hat{\mathbf{C}}_{zzzz}(k) \right\} \right)$

$\mathbf{H}_f(k+1) = \mathbf{H}_f(k) - \mu(k)\, \hat{\nabla}(k)$

Note that the given computational complexity is obtained by recursively calculating higher–order cumulants as shown in Table 5.2. It is apparent that the computational burden per iteration required by the AIM–FOS algorithm is very heavy.

Example 5.4

To compare the performance of the AIM–FOS algorithm with that of the AIM–SOS algorithm, let us consider the same problems as in Example 5.3. When the stationary narrowband interference is $\{I_2(k)\}$ with $A_2 = 0.75$ and $\sigma_{n_p}^2 = \sigma_{n_r}^2 = 0.001$, the error curve between the SOI and the system output obtained by the AIM–FOS algorithms ($N = 16$, $L = 30$, $\mu = 0.99$) is illustrated in Fig. 5.6 (a). As expected, the performance of the AIM–FOS algorithm is very similar to that of the AIM–SOS algorithm shown in Fig. 5.5 (a) when the noise powers are quite small.

Next, consider that the interference is $\{I_1(k)\}$ with $A_1 = 0.4$. When $\sigma_{n_p}^2 = 0.001$ and $\sigma_{n_r}^2 = 0.01$, the error curve between the SOI and the system output obtained by the AIM–FOS algorithms ($N = 8$, $L = 14$ and $\mu = 0.99$) is illustrated in Fig. 5.6 (b). When the noise power of the reference input is large, the AIM–SOS algorithm shown in Fig. 5.5 (b) performs very poorly while the AIM–FOS algorithm, although converges slowly, it performs better. The reason why the AIM–FOS algorithm converges slowly is due to the fact that a large number of data is required for fourth–order cumulants to suppress additive Gaussian noises. This example demonstrates that the AIM–FOS algorithm outperforms the AIM–SOS algorithm in the environments where the additive noise at the receivers are Gaussian and have large variance.

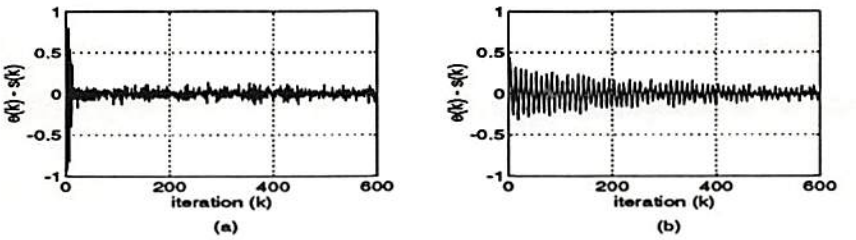

Figure 5.6: The error curves $\{e(k) - s(k)\}$ obtained by the AIM–FOS algorithm, (a) when $M = 16$, $\mu_s = 0.99$, $\{I_2(k)\}$ with $A_2 = 0.75$, and $\sigma^2_{n_p} = \sigma^2_{n_r} = 0.001$; and (b) when $M = 8$, $\mu_s = 0.99$, $\{I_1(k)\}$ with $A_1 = 0.4$, and $\sigma^2_{n_p} = 0.001$, and $\sigma^2_{n_r} = 0.01$.

Example 5.5

This example presents the performance of the AIM–FOS algorithm with different values of the step size parameter μ. Consider the same SOI and interference as in the Example 5.2: i.e., the stationary interference is $\{I_2(k)\}$ with $A_2 = 1$. When $M = 16$ and $\ell = 4$, the error curves between the SOI and the system outputs obtained by the AIM–FOS algorithm are presented in Fig. 5.7 with four different step size parameters: $\mu = 0.995$, 0.75, 0.4, and 0.1. From this figure, it can be observed that there are no noticeable performance differences of the AIM–FOS algorithm, in terms of convergence speed and excess error, among the results with different step size parameter μ.

Through the Examples 5.1, 5.2, and 5.3, we can conclude that the performance of the AIM–FOS algorithm is *less sensitive to the choice of step size parameter*; *less sensitive to the reference signal statistics*; and *less sensitive to the Gaussian uncorrelated noises* than AIM algorithms based on second–order statistics, in terms of convergence speed and excess error. Furthermore, since we don't know the relationship between the reference and the interference, it is difficult to decide the number of

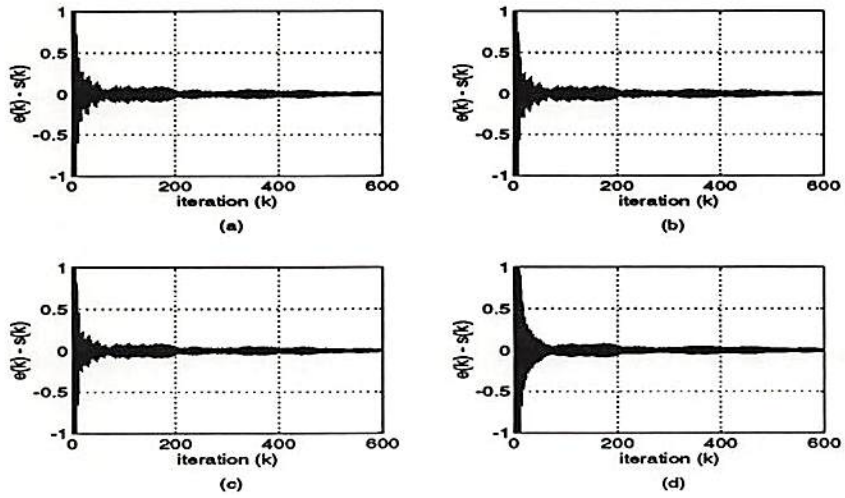

Figure 5.7: The error curves $\{e(k) - s(k)\}$ obtained by the AIM–FOS algorithm ($M = 16$ and $\ell = 4$), when $\{I_2(k)\}$ with $A_2 = 1$ and the step size μ is (a) 0.995, (b) 0.75, (c) 0.4, and (d) 0.1.

an AIM filter taps to mitigate multiple interferences. Thus, the number of an AIM filter taps can be sometimes overestimated and sometimes underestimated. It has been observed that the AIM–FOS algorithm outperforms the AIM–NLMS algorithm when we underestimate the number of the AIM filter taps [Shin and Nikias, 1994].

Example 5.6

We assume that there exist stationary, additive, zero–mean Gaussian uncorrelated noise sources at the primary and reference inputs. When the stationary single interference is $\{I_2(k)\}$ with $A_2 = 2$, the error curves between the SOI and the system outputs obtained by the AIM–NLMS algorithm ($N = 16$ and $\mu_\ell = 0.075$) and the AIM–FOS algorithm ($N = 16$, $L = 4$, and $\mu_f = 0.99$) are shown in Fig. 5.8 with

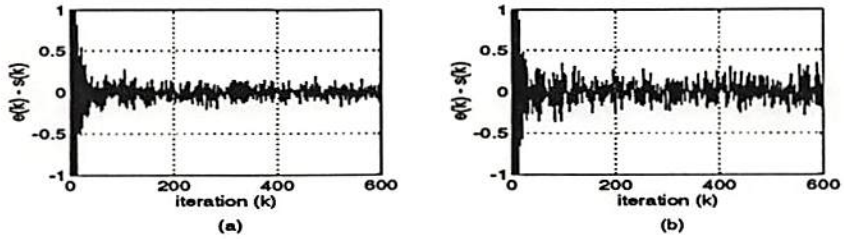

Figure 5.8: For interference $\{I_2(k)\}$ with $A_2 = 2$, the error curves between the SOI and the system output obtained by the AIM–FOS algorithm ($M = 16$, $\ell = 4$, and $\mu = 0.99$) when the variance of the Gaussian uncorrelated noises is (a) 0.0025 and (b) 0.0075.

$\sigma^2_{n_p} = \sigma^2_{n_r} = 0.0025$ and 0.0075. Although the additive uncorrelated noises deteriorate both performance of the AIM–NLMS and AIM–FOS algorithms, as expected, it has been observed that AIM–FOS algorithm performs better than the AIM–NLMS algorithm when the noise variance is small [Shin and Nikias, 1994].

Now, let us assume that the variances of the uncorrelated noises are both 0.005. The multiple interference is $\{I_1(k) + I_2(k)\}$ with $A_1 = 0.75$ and $A_2 = 2$ or $\{I_1(k) + I_2(k) + I_3(k)\}$ with $A_1 = 1$, $A_2 = 0.5$, and $A_3 = 0.75$. When the interference is the sum of the two interferences, the error curves between the SOI and the system outputs obtained by the AIM–NLMS algorithm ($N = 16$ and $\mu_\ell = 0.75$) and AIM–FOS algorithm ($N = 16$, $L = 5$, and $\mu_f = 0.99$) are shown in Fig. 5.9 (a). When the interference is the sum of the three interferences, the error curves between the SOI and the system output obtained by AIM–FOS algorithm ($N = 24$, $\ell = 5$, and $\mu = 0.99$) are shown in Fig. 5.9 (b). Again, although the additive uncorrelated noises deteriorate both performance of the AIM–NLMS and AIM–FOS algorithms, the AIM–FOS algorithm performs better than the AIM–NLMS algorithm for multiple interference cases.

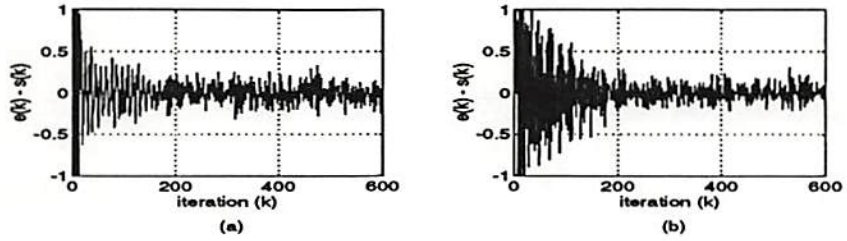

Figure 5.9: When the variance of the Gaussian uncorrelated noises is 0.005, (a) the error curve between the SOI and the system output obtained by the AIM–FOS algorithm ($M = 16$, $\ell = 5$, and $\mu = 0.99$) for the interference $\{I_1(k) + I_2(k)\}$; and (b) the error curve between the SOI and the system output obtained by the AIM–FOS algorithm ($M = 24$, $\ell = 5$, and $\mu = 0.99$) for the interference $\{I_1(k) + I_2(k) + I_3(k)\}$.

Example 5.7

Now, let us assume that there are no additive uncorrelated Gaussian noises at the primary and reference inputs and that the interference is the sum of three interferences, $\{I_1(k) + I_2(k) + I_3(k)\}$ that are generated by a stationary reference signal $\{w_1(k)\}$, a reference signal with time–varying amplitude $\{w_2(k)\}$, and a reference signal with time–varying frequency $\{w_3(k)\}$, respectively. The frequency of $\{w_1(k)\}$ are 0.25 and its amplitude is 0.75; the frequency of $\{w_2(k)\}$ is 0.35 and its amplitude is time–varying; the amplitude of $\{w_3(k)\}$ is unity and its frequency is time–varying. The time–varying amplitude of $\{w_2(k)\}$ and the time–varying frequency of $\{w_3(k)\}$ are shown in Fig. 5.10. The error curves between the SOI and the system outputs obtained by the AIM–NLMS algorithm ($N = 24$ and $\mu_\ell = 0.075$) and the AIM–FOS algorithm ($N = 24$, $L = 5$, and $\mu_f = 0.99$) are also shown in this figure. This simulation result demonstrates that we can use the AIM–FOS algorithm to eliminate multiple non–stationary interference using the reference signal that is the sum of the corresponding reference signals.

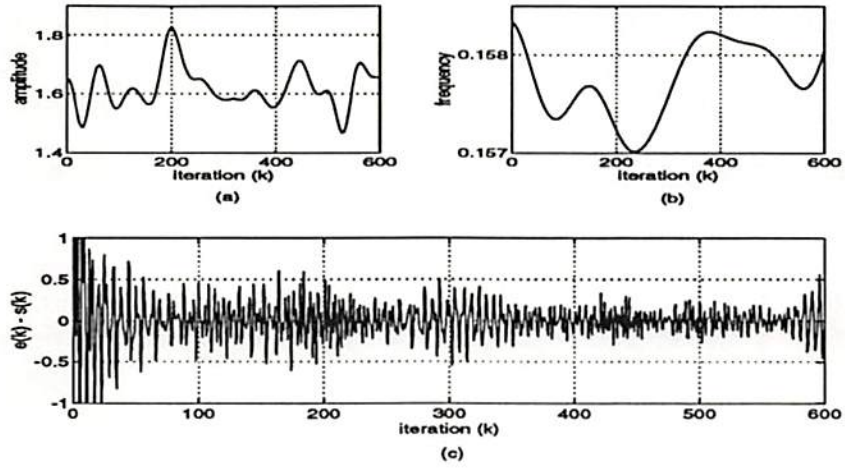

Figure 5.10: When there are no uncorrelated noise sources and the interference is the sum of one stationary ($A_1 = 0.75$ and $f_1T = 0.25$) and two non–stationary interferences with (a) time–varying amplitude A_2 and $f_2T = 0.35$ and (b) time–varying frequency f_3T and $A_3 = 1.0$; the error curve between the SOI and the system output obtained by (c) the AIM–FOS algorithm ($M = 24$, $\ell = 5$, and $\mu = 0.99$).

Note that the performance of the AIM–NLMS algorithm is presented in [Shin and Nikias, 1994] when the interference is stationary or non–stationary with uncorrelated noises at the primary and reference inputs.

5.7 Summary

We have described the AIM–HOS formulations using a gradient–type algorithm and higher–order statistics, when the reference signal is available. When the reference signals are non–Gaussian with non–zero third–order and fourth–order cumulants, we can use the AIM–TOS and the AIM–FOS algorithms, respectively, to mitigate the presence of interference. Since their filter coefficients are updated in higher–order statistics domains, the update equations are independent from Gaussian noises

at the primary and reference inputs. Consequently, the AIM–HOS algorithms can more effectively mitigate the interference than other AIM algorithms whose filter update equation is affected by the uncorrelated noises. The AIM–HOS algorithms can be utilized to mitigate single and multiple interferences which are stationary or non–stationary.

Chapter 6

Nonlinear Methods For Noisy Channels

6.1 Introduction

In this chapter, we consider adaptive interference mitigation (AIM) schemes that produce the prediction of a narrowband interference corrupting a broadband signal of interest (SOI).

Due to their superior performance against jamming and interception, wideband communication formats have been of interest in military and several commercial applications [Poor, 1992]. With their many other advantageous properties, wideband communication formats offer an effective way of combating narrowband interference. Since the transmission bandwidths of spread–spectrum signals are much greater than their message bandwidths, the system has an inherent capability to reject interference signal whose bandwidths are small compared to that of the wideband SOI. It has been shown that the narrowband interference rejection capability of spread–spectrum systems can be improved substantially by suitably processing the received signal [Hsu and Giordano, 1978; Li and

Milstein, 1982; Iltis and Milstein, 1984; Milstein, 1988]. Narrowband interference suppression can be based on the following idea. Since the spread spectrum SOI has a nearly flat spectrum, it cannot be predicted accurately from its past or future values. On the other hand, the narrowband interference signal can be predicted accurately because of being narrowband. Hence, a prediction of the received signal based on either previous or future values of the received signal will be an estimate of the interference. By subtracting a prediction of the received signal and using the resulting prediction error as the input to the matched filter, the effect of the interfering signal can be mitigated.

Although it is well known that linear detection and estimation techniques are optimal for a Gaussian process, it has been observed that these linear techniques are not optimal for a non–Gaussian process and the difference in performance between linear and nonlinear processing for a non–Gaussian process is dramatic due to the effects of the heavy tails of its distribution. Thus, the utilization of nonlinear techniques is desired and required for performance improvement. Assuming that the broadband SOI corrupted by a narrowband interference is transmitting over an additive noise channel, we describe *nonlinear AIM techniques* whose performance are substantially better than linear AIM methods.

6.2 Preliminaries

This section presents models for the spread–spectrum signal of interest (SOI) and narrowband interferences; and, the generalized Gaussian distributions and the non–Gaussian filtering algorithm with known parameters developed by Masreliez [1975].

6.2.1 Direct–Sequence Spread–Spectrum (DSSS) System Model

The total direct–sequence spread–spectrum (DSSS) transmitted SOI, $\{s(t)\}$, can be expressed as

$$s(t) = \sum_k b_k \, m(t - kT_b) \tag{6.1}$$

where $\{b_k\}$ is the binary information sequence, $\{m(t)\}$ is the modulation waveform, and T_b is the bit duration. For a DSSS system the low–pass equivalent modulation waveform is written as

$$m(t) = \sum_{k=0}^{K-1} c_k \, q(t - k\tau_c) \tag{6.2}$$

where $\{c_k\}$ is the binary pseudonoise (PN) chip sequence used to spread the transmitted signal, $\{q(t)\}$ is a rectangular pulse of duration τ_c, and K is the number of chips per message bit. When each bit has the same modulation sequence: i.e., $T_b = K\tau_c$, then the received signal, $\{x(t)\}$, is of the form

$$x(t) = \alpha \, s(t - \tau) + n(t) + I(t) \tag{6.3}$$

where α is an attenuation factor and τ is a delay offset. The processes $\{n(t)\}$ and $\{I(t)\}$ denote a channel noise process and a narrowband interference, respectively. It can be assumed that $\alpha = 1$ and $\tau = 0$, because these parameters do not change the analysis with respect to a channel. After the received waveform has been processed by a matched filter and sampled at the chip rate $1/\tau_c$, the received signal becomes

$$x(k) = s(k) + n(k) + I(k). \tag{6.4}$$

If the PN sequence is assumed to be truly random, due to the rectangular shape of $\{q(t)\}$, the sequence $\{s(k)\}$ can be considered as an independent, identically distributed (i.i.d.) binary random sequence with

$\Pr\{s(k) = -1\} = \Pr\{s(k) = 1\} = 0.5$. Also, the sequences $\{s(k)\}$, $\{n(k)\}$, and $\{I(k)\}$ are assumed to be mutually independent. This transmission system model is illustrated in Fig. 6.1.

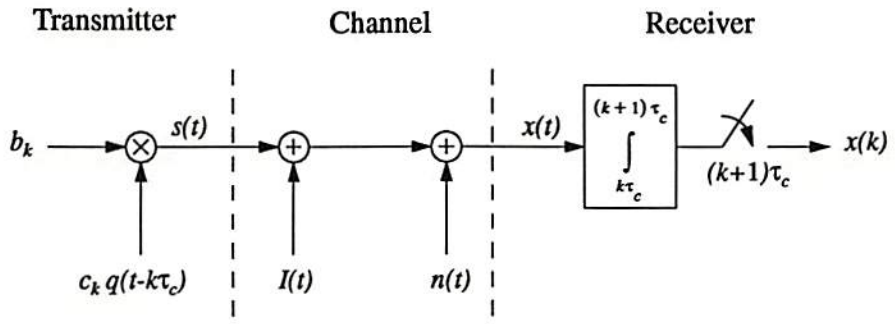

Figure 6.1: Baseband model for the received signal.

6.2.2 Autoregressive Interference Model

One way to (approximately) describe a narrowband interference is by using a finite sum of sinusoids. However, a more physically accurate broad class of narrowband interferences consists of those which can be modeled as an N–th order autoregressive (AR) process of the form

$$I(k) = \sum_{n=1}^{N} \phi_n\, I(k-n) + g(k) \tag{6.5}$$

where $\{g(k)\}$ is a zero–mean, white Gaussian process and $\{\phi_n,\ n = 1, 2, \ldots, N\}$ are the AR parameters. Then, the *received signal* $\{x(k)\}$ can be modeled using the following state–space representation:

$$\begin{aligned} \mathbf{u}(k) &= \mathbf{\Phi u}(k-1) + \mathbf{w}(k) \\ x(k) &= \mathbf{H u}(k) + v(k) \\ v(k) &= n(k) + s(k) \end{aligned} \tag{6.6a}$$

where

$$\mathbf{u}(k) = [\, I(k),\ I(k-1),\ \ldots,\ I(k-N+1)\,]^T$$

$$\mathbf{w}(k) = [\, g(k),\ 0,\ 0,\ \ldots,\ 0\,]^T$$

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_{N-1} & \phi_N \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{6.6b}$$

$$\mathbf{H} = [\, 1,\ 0,\ 0,\ \ldots,\ 0\,].$$

The sequence $\{n(k)\}$ is the noise process derived by using a channel model and $\{s(k)\}$ is the DSSS SOI. The processes $\{g(k)\}$ and $\{v(k)\}$ are zero–mean, independent sequences which are mutually independent.

6.2.3 Generalized Gaussian Distributions

The *generalized Gaussian distributions* [Gray, 1979] are a family of symmetrical probability density functions defined by

$$gG(x, \alpha, \beta) = \frac{\alpha}{2\beta\Gamma(\frac{1}{\alpha})}\, e^{-\left(\frac{|x|}{\beta}\right)^{\alpha}} \tag{6.7a}$$

where

$$\begin{array}{ll} -\infty < x < \infty & \\ \Gamma(\cdot) & \text{is the gamma function;} \\ \alpha > 0 & \text{is the shape parameter;} \\ \beta > 0 & \text{is the scale parameter.} \end{array} \tag{6.7b}$$

It is also called *Subbotin's distribution* in the statistical literature. Miller and Thomas [1972] used generalized Gaussian distributions in detection theory as a model for non–Gaussian noise. In the sequel, $gG(x, \alpha, \beta)$ denotes the generalized Gaussian distribution with parameters α and β.

The generalized Gaussian family covers a wide range of distributions. When $\alpha = 1$, it becomes the double exponential distribution (Laplace distribution)

$$gG(x, \beta, 1) = \frac{1}{2\beta} e^{-\frac{|x|}{\beta}} \qquad -\infty < x < \infty. \tag{6.8}$$

When $\alpha = 2$, it becomes the Gaussian distribution

$$gG(x, \beta, 2) = \frac{1}{\beta\sqrt{\pi}} e^{-\left(\frac{|x|}{\beta}\right)^2} \qquad -\infty < x < \infty \tag{6.9}$$

where its variance σ^2 becomes $\beta^2/2$. As α tends toward infinity, it approaches the uniform distribution

$$gG(x, \beta, \infty) = \frac{1}{2\beta} \qquad -\beta < x < \beta. \tag{6.10}$$

As α tends toward zero, the chances of finding an event in a length of finite samples goes to zero. That can loosely describe the certain event. In this chapter, we assume that the distribution of the channel noise process $\{n(k)\}$ is a member of the generalized Gaussian distributions.

6.2.4 Masreliez Filter with Known Parameters

An approximate conditional mean (ACM) filter, which can be considered as an extension of the Kalman–Bucy filter, has been developed by Masreliez [1975] for estimating the state of a linear system described by (6.6a) with non–Gaussian either *state noise* $\{\mathbf{w}(k)\}$ or *measurement noise* $\{v(k)\}$. It is assumed that these noises are zero–mean independent sequences and mutually independent and that the initial state $\mathbf{u}(0)$ is independent of the future plant and measurement disturbances. For more general cases, the dimensions of the plant state vector and the measurement vector are assumed to be $N \times 1$ and $R \times 1$, respectively. When the minimum variance estimator is considered as the optimal estimator, the

optimal estimator is given by

$$\hat{\mathbf{u}}(k) = E\{\mathbf{u}(k) \mid \mathbf{X}(k)\} \tag{6.11}$$

where $\mathbf{X}(k) = [\,\mathbf{x}(1),\ \mathbf{x}(2),\ \ldots,\ \mathbf{x}(k)\,]^T$. Then, Masreliez [1975] has introduced the following two theorems constructing nonlinear filters for non–Gaussian processes.

Suppose that the observation prediction density $f(\mathbf{x}(k) \mid \mathbf{X}(k-1))$ is twice differentiable. In the case where the observation prediction density is not Gaussian, we have the following theorem.

Theorem 6.1 *Assume that* $f(\,\mathbf{u}(k) \mid \mathbf{X}(k-1)\,)$ *is a Gaussian density with mean* $\bar{\mathbf{u}}(k)$ *and covariance matrix* $\mathbf{M}(k)$ *and that* $E\{\mathbf{w}(k)\mathbf{w}^T(j)\} = \mathbf{Q}(k)\delta(k-j)$. *Then the conditional expectation* $\hat{\mathbf{u}}(k)$ *and its conditional covariance* $\mathbf{P}(k) = E\{(\hat{\mathbf{u}}(k) - \mathbf{u}(k))(\hat{\mathbf{u}}(k) - \mathbf{u}(k))^T \mid \mathbf{X}(k)\}$ *satisfy*

$$\begin{aligned} \hat{\mathbf{u}}(k) &= \bar{\mathbf{u}}(k) + \mathbf{M}(k)\mathbf{H}^T\,\mathbf{g}_k(\,\mathbf{x}(k)\,) \\ \mathbf{P}(k) &= \mathbf{M}(k) - \mathbf{M}(k)\mathbf{H}^T\,\mathbf{G}_k(\,\mathbf{x}(k)\,)\,\mathbf{H}\mathbf{M}(k) \\ \mathbf{M}(k+1) &= \mathbf{\Phi}\mathbf{P}(k)\mathbf{\Phi}^T + \mathbf{Q}(k) \end{aligned} \tag{6.12a}$$

where $\mathbf{g}_k(\,\mathbf{x}(k)\,)$ *is a column vector with components*

$$\{\mathbf{g}_k(\,\mathbf{x}(k)\,)\}_i = -\frac{\frac{\partial f(\,\mathbf{x}(k)|\mathbf{X}(k-1)\,)}{\partial(\mathbf{x}(k))_i}}{f(\,\mathbf{x}(k) \mid \mathbf{X}(k-1)\,)} \tag{6.12b}$$

and $\mathbf{G}_k(\mathbf{x}(k))$ *is a matrix with elements*

$$\{\mathbf{G}_k(\,\mathbf{x}(k)\,)\}_{ij} = \frac{\partial\{\mathbf{g}_k(\,\mathbf{x}(k)\,)\}_i}{\partial(\mathbf{x}(k))_j}. \tag{6.12c}$$

In the situation of Gaussian observation disturbances and non–Gaussian plant noises, we have the following theorem.

Theorem 6.2 *Assume that the density for the observation noise* $\{v(k)\}$ *is Gaussian with zero mean and covariance matrix* $\mathbf{R}(k)$ *and that the matrix* $\mathbf{H}^T\mathbf{R}^{-1}(k)\mathbf{H}$ *is nonsingular for all* k. *Then the minimum variance*

estimator for $\mathbf{u}(k)$ *may be computed from the following relations:*

$$\begin{aligned} \mathbf{T}^{-1}(k) &\equiv \mathbf{H}^T\mathbf{R}^{-1}(k)\mathbf{H} \\ \hat{\mathbf{u}}(k) &= \mathbf{T}(k)\mathbf{H}^T\Big(\mathbf{R}^{-1}(k)\mathbf{x}(k) - \mathbf{g}_k(\,\mathbf{x}(k)\,)\Big) \\ \mathbf{P}(k) &= \mathbf{T}(k) - \mathbf{T}(k)\mathbf{H}^T\,\mathbf{G}_k(\,\mathbf{x}(k)\,)\,\mathbf{H}\mathbf{T}(k) \\ \mathbf{M}(k+1) &= \boldsymbol{\Phi}\mathbf{P}(k)\boldsymbol{\Phi}^T + \mathbf{Q}(k) \end{aligned} \tag{6.13}$$

where $\mathbf{g}_k(\cdot)$ *and* $\mathbf{G}_k(\cdot)$ *are as in Theorem 1.*

In these theorems, it is assumed that the parameters of the interference are *a priori* exactly known. In practice, however, these parameters are rarely known in advance or may vary with time. Thus, in order to effectively mitigate the interference with either unknown or time–varying characteristics, one should use an AIM algorithm which can adjust itself to the variation of the interference characteristics.

6.3 Linear AIM Methods

We review first linear AIM methods based on the LMS algorithm and discuss their interference mitigation performance.

6.3.1 One–sided Linear AIM Method Based on LMS Algorithm

The tap–weights in the one–sided linear predictor are adjusted using the LMS algorithm

$$\mathbf{H}_{\ell 1}(k) = \mathbf{H}_{\ell 1}(k-1) + \frac{\mu}{\hat{P}_{\ell 1}(k)}\, e_{\ell 1}(k)\,\mathbf{X}(k) \tag{6.14a}$$

where $\mathbf{X}(k)$ is the input tap vector given by

$$\mathbf{X}(k) = \Big[\, x(k-1),\, x(k-2),\, \ldots,\, x(k-M)\,\Big]^T, \tag{6.14b}$$

$\mathbf{H}_{\ell 1}(k)$ is the weight vector

$$\mathbf{H}_{\ell 1}(k) = \Big[h_1(k),\, h_2(k),\, \ldots,\, h_M(k) \Big]^T, \tag{6.14c}$$

μ is a tuning constant controlling the stability and convergence rate of the algorithm, and $\hat{P}_{\ell 1}(k)$ is an estimate of the input power obtained by

$$\hat{P}_{\ell 1}(k) = \hat{P}_{\ell 1}(k-1) + \gamma \Big(\| \mathbf{X}(k) \|^2 - \hat{P}_{\ell 1}(k-1) \Big). \tag{6.14d}$$

In (6.14d), γ is another step size constant. This procedure is needed to make the choice of tuning constant invariant to changes in input signal levels. The output of the one–sided linear prediction AIM method $\{e_{\ell 1}(k)\}$ is obtained by

$$e_{\ell 1}(k) = x(k) - \hat{x}_{\ell 1}(k) \tag{6.15a}$$

where

$$\hat{x}_{\ell 1}(k) = \mathbf{X}^T(k)\, \mathbf{H}_{\ell 1}(k-1) \tag{6.15b}$$

is the predicted value of the received signal $\{x(k)\}$ based on the M immediate past received values. The prediction error, which is the output, is the estimate of the wideband SOI sequence. The structure of the one–sided linear prediction AIM method is shown in Fig. 6.2. The advantages and disadvantages of the one–sided linear prediction AIM method are similar to those of the LMS algorithm.

6.3.2 Two–sided Linear AIM Method Based on LMS Algorithm

The tap–weights in the two–sided linear predictor are adjusted using the LMS algorithm

$$\mathbf{H}_{\ell 2}(k) = \mathbf{H}_{\ell 2}(k-1) + \frac{\mu}{\hat{P}_{\ell 2}(k)}\, e_{\ell 2}(k)\, \mathbf{X}_{\ell 2}(k) \tag{6.16a}$$

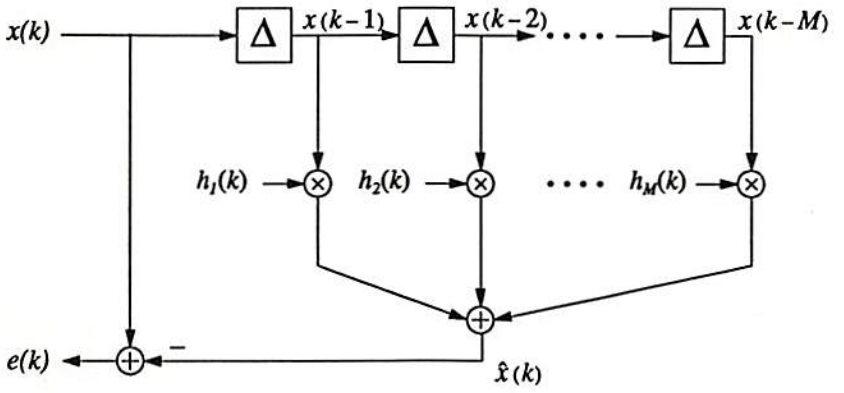

Figure 6.2: The structure of the one–sided linear prediction AIM method.

where $\mathbf{X}_{\ell 2}(k)$ is the input vector [Li and Milstein, 1982]

$$\mathbf{X}_{\ell 2}(k) = \Big[x(k+\frac{M}{2}),\, x(k+\frac{M}{2}-1), \ldots, x(k+1), \\ x(k-1),\, x(k-2),\, \ldots,\, x(k-\frac{M}{2}) \Big]^T, \quad (6.16b)$$

$\mathbf{H}_{\ell 2}(k)$ is the weight vector of the two–sided linear predictor, μ is a tuning constant, and $\hat{P}_{\ell 2}(k)$ is an estimate of the input power obtained by

$$\hat{P}_{\ell 2}(k) = \hat{P}_{\ell 2}(k-1) + \gamma \left(\| \mathbf{X}_{\ell 2}(k) \|^2 - \hat{P}_{\ell 2}(k-1) \right). \quad (6.16c)$$

In (6.16c), γ is another step size constant. The output of the two–sided linear prediction AIM method $\{e_{\ell 2}(k)\}$ is obtained by

$$e_{\ell 2}(k) = x(k) - \hat{x}_{\ell 2}(k) \quad (6.17a)$$

where

$$\hat{x}_{\ell 2}(k) = \mathbf{X}_{\ell 2}^T(k)\, \mathbf{H}_{\ell 2}(k-1) \quad (6.17b)$$

is the predicted value of the received signal $\{x(k)\}$ based on the $M/2$ immediate past and $M/2$ immediate future received values. The advantages of the two–sided linear prediction AIM method are:

- Since it is based on the LMS algorithm and symmetric structure, it is simple and has low computational complexity.
- When the frequency of the interference tone is near the carrier frequency, its performance is better than the performance of the one–sided linear prediction AIM method with the same number of taps (in some cases by approximately 6 dB) [Li and Milstein, 1982].

It shares the same disadvantages as the one–sided linear prediction AIM method described in the previous section.

Example 6.1

The SOI is an i.i.d. binary random sequence with $\Pr\{s_k = -1\} = \Pr\{s_k = 1\} = 0.5$. The background white noise source $\{n(k)\}$ is a Laplace ($\alpha = 1$) process whose power is kept constant at 0.01. The number of filter taps M is 10 for each linear prediction AIM method. The performance comparison is made in terms of signal–to–interference plus noise ratio (SINR) improvement defined as

$$\text{SINR improvement (dB)} = 10 \log_{10} \frac{\mathbf{E}\{\,|x(k) - s(k)\,|^2\}}{\mathbf{E}\{\,|e(k) - s(k)\,|^2\}}$$

where $\{x(k)\}$ is the given input data and $\{e(k)\}$ is the output of each linear AIM algorithm. After 1,000 iterations, the variances over windows of last 500 data points are calculated for the SINR improvement. Five different input SINRs: -15 dB, -10 dB, -5 dB, 0 dB, and 5 dB, are considered. The input SINR is defined as

$$\text{SINR at the input (dB)} = 10 \log_{10} \frac{\mathbf{E}\{s^2(k)\}}{\mathbf{E}\{\,|x(k) - s(k)\,|^2\}}.$$

When the narrowband interference is a single–tone sinusoidal signal given by

$$I(k) = A_I \sin(2\pi(0.15)k + \phi)$$

where A_I denotes the amplitude determined by the given SINR and ϕ is a random phase uniformly distributed over $[-\pi, \pi]$. Then, let us use the one–sided and two–sided linear prediction AIM methods described in the previous sections to mitigate the single–tone interference. The averaged SINR improvements over 10 independent runs obtained by the two linear prediction AIM methods are summarized in Table 6.1.

From Table 6.1, it is observed that the performance of the two–sided linear AIM method is a little better than that of the one–sided linear AIM method. Another way to compare the performance between the two linear AIM algorithms is to illustrate the *error curves* between the SOI and the system output (restored SOI) obtained by the each linear AIM method: i.e., $\{e(k) - s(k)\}$. Fig. 6.3 illustrates typical error curves when the input SINR is -10 dB. From Fig. 6.3, we can observe that although the corresponding SINR improvements are high, both the error curves contain a large number of *outliers* whose magnitude is greater and closer to unity. Note that these outliers cause signal detection errors and that the signal detection capability is more important than the SINR improvement.

Table 6.1: The SINR Improvement of the One–Sided and the Two–Sided Linear Prediction AIM Method in the Cases of a Single Sinusoidal Interference in Laplace Noise Channel with Variance 0.01.

Input SINR (dB)	SINR Improvement (dB)	
	One–Sided Linear AIM	Two–Sided Linear AIM
-15	18.8511	20.6088
-10	13.8603	15.6373
-5	9.2113	10.6769
0	4.7012	6.0416
5	0.1823	2.2290

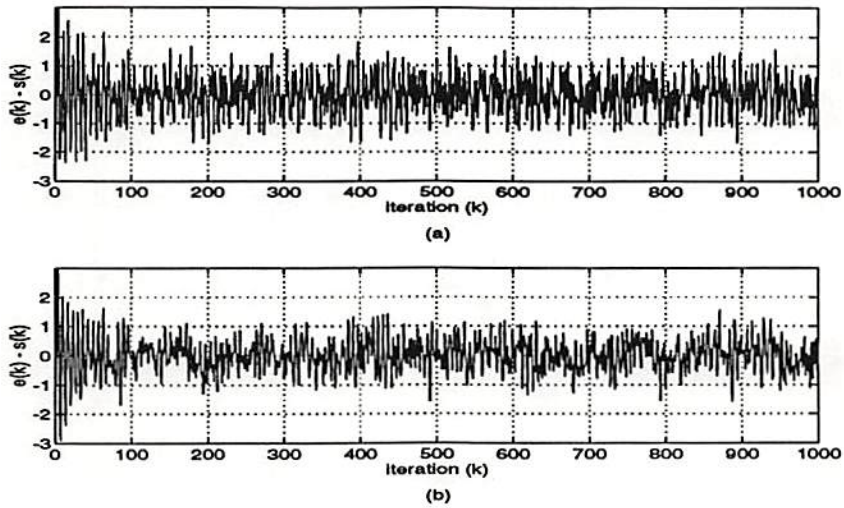

Figure 6.3: When the narrowband interference is a sinusoidal; the variance of Laplace channel noise is 0.01; and the input SINR is -10 dB, the error curves between the SOI and the system outputs obtained by (a) the one–sided linear prediction AIM method and (b) the two–sided linear prediction AIM method.

Example 6.2

Considering the same SOI in the previous example, the narrowband interference is generated by passing a white Gaussian process through the following second–order IIR filter:

$$I(k) = 1.98\, I(k-1) - 0.9801\, I(k-2) + g(k)$$

where $\{g(k)\}$ is a zero–mean, white Gaussian process.

When the variance of the Laplace channel noise is 0.01, the averaged SINR improvements over 10 independent runs obtained by the linear prediction AIM algorithms are summarized in Table 6.2. And, Fig. 6.4 illustrates typical error curves between the SOI and the system output (recovered SOI) obtained by the linear AIM methods, when the input

Table 6.2: The SINR Improvement of the One–Sided and the Two–Sided Linear Prediction AIM Methods in the Cases of AR Interferences in Laplace Noise Channel with Variance 0.01.

Input SINR (dB)	SINR Improvement (dB)	
	One–Sided Linear AIM	Two–Sided Linear AIM
−15	20.6024	23.0208
−10	16.5164	18.4622
−5	12.6351	14.0291
0	9.6269	10.4254
5	3.8602	8.2513

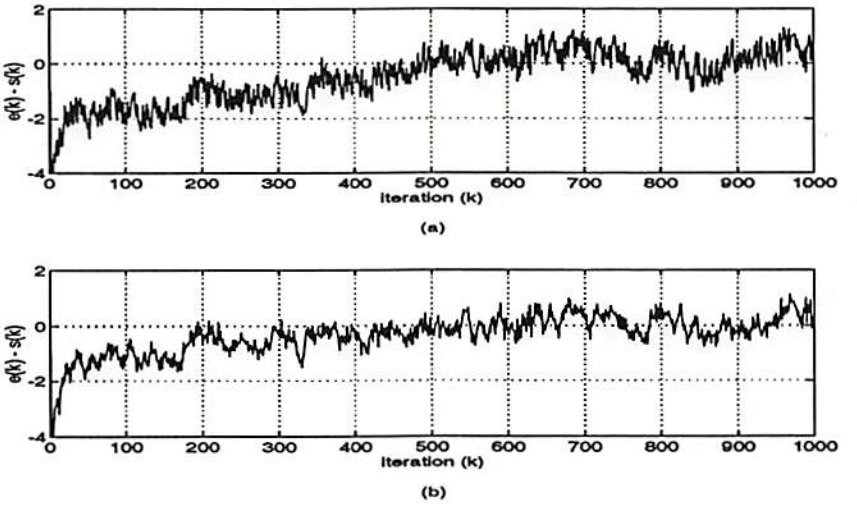

Figure 6.4: When the narrowband interference is generated through a AR model; the variance of Laplace channel noise is 0.01; and the input SINR is −10 dB, the error curves between the SOI and the system outputs obtained by (a) the one–sided linear prediction AIM method and (b) the two–sided linear prediction AIM method.

SINR is -10 dB. The same phenomenon can be observed as in Fig. 6.3. A lot of outliers whose magnitude is greater and closer to unity exist after convergence.

These two examples demonstrate that the utilization of linear AIM techniques in the applications where processes are not Gaussian is not an appropriate choice to effectively mitigate the narrowband interferences that can be represented by either a single–tone sinusoidal signal or an AR process excited by a white Gaussian process.

6.4 Nonlinear AIM Methods

Since linear AIM methods do not effectively mitigate the narrowband interferences in the DSSS communication system, it is necessary to use nonlinear AIM methods that perform better in the given broadband communication format. We describe in this section the nonlinear AIM methods based on the ACM algorithm for Gaussian and impulsive noise channels.

6.4.1 ACM–Based AIM Method

In order to suppress an interference with unknown or varying parameters, an obvious method of implementing a nonlinear AIM algorithm would be to identify the AR parameters of the interference recursively and to carry out the ACM filtering algorithm using the estimates obtained at each iteration. However, since the ACM filter using estimates of AR parameters is sensitive to variations in the parameter values, the obvious approach is not promising [Vijayan and Poor, 1990].

6.4.2 Nonlinear AIM Method for Gaussian Noise Channels

Considering the state–space representation (6.6a), suppose that the channel noise $\{n(k)\}$ is a zero–mean white Gaussian process. Then, since $\{v(k)\}$ in (6.6a) is the sum of two independent random variables, one of which is Gaussian and the other takes on values $+1$ or -1 with equal probability, its density becomes the weighted sum of two Gaussian densities given by [Vijayan and Poor, 1990]

$$f_v(v) = \frac{1}{2}\left[N_{\sigma_n^2}(v-1) + N_{\sigma_n^2}(v+1) \right] \tag{6.18a}$$

where σ_n^2 denotes the variance of the channel noise $\{n(k)\}$ and the zero–mean Gaussian probability density $N_{\sigma_n^2}(\cdot)$ is defined by

$$N_{\sigma_n^2}(v) = \frac{1}{\sqrt{2\pi}\sigma_n} e^{-v^2/2\sigma_n^2}. \tag{6.18b}$$

Since $f(\,\mathbf{u}(k) \mid \mathbf{X}(k-1)\,)$ is assumed to be Gaussian (Theorem 1), using the fact that $\{v(k)\}$ is independent of $\mathbf{X}(k-1)$, we can obtain the following expression for the observation prediction density:

$$\begin{aligned} f(\,x(k) \mid \mathbf{X}(k-1)\,) &= f(\,\mathbf{H}\mathbf{u}(k) + v(k) \mid \mathbf{X}(k-1)\,) \\ &= \frac{1}{2}\Big[N_{\mathbf{H}\mathbf{M}(k)\mathbf{H}^T+\sigma_n^2}(\,x(k) - \mathbf{H}\bar{\mathbf{u}}(k) - 1\,) \\ &\qquad + N_{\mathbf{H}\mathbf{M}(k)\mathbf{H}^T+\sigma_n^2}(\,x(k) - \mathbf{H}\bar{\mathbf{u}}(k) + 1\,)\Big]. \end{aligned} \tag{6.19}$$

Substituting the expression of the observation prediction density into (6.12b), we get

$$g_k(\,x(k)\,) = \frac{x(k) - \mathbf{H}\bar{\mathbf{u}}(k) - \tanh\left(\frac{x(k)-\mathbf{H}\bar{\mathbf{u}}(k)}{\mathbf{H}\mathbf{M}(k)\mathbf{H}^T+\sigma_n^2}\right)}{\mathbf{H}\mathbf{M}(k)\mathbf{H}^T + \sigma_n^2}. \tag{6.20}$$

Using the following relationships

$$x(k) = \hat{x}(k) + e(k) \tag{6.21a}$$

$$\hat{x}(k) = \sum_{m=1}^{M} h_m(k-1)\, x(k-m), \tag{6.21b}$$

where $\{e(k)\}$ is the system output that the sum of a Gaussian channel noise and a binary SOI, we get a nonlinear transversal filter for the prediction of $\{x(k)\}$ [Vijayan and Poor, 1990]:

$$\hat{x}(k) = \sum_{m=1}^{M} h_m(k-1) \left[\hat{x}(k-m) + \rho_{k-m}(e(k-m)) \right] \tag{6.22a}$$

where $\rho(\cdot)$ is a nonlinear term obtained from $\{g_k(x(k))\}$ appearing in the ACM algorithm and can be written as

$$\rho_k(e(k)) = e(k) - \tanh\left(\frac{e(k)}{\sigma^2(k)} \right). \tag{6.22b}$$

Note that $\sigma^2(k)$ is the variance of the Gaussian channel noise. In order to implement the nonlinear AIM algorithm, an estimate of the parameter $\sigma^2(k)$ must be obtained. A useful estimate for $\sigma^2(k)$ is given by

$$\hat{\sigma}^2(k) = \Delta(k) - 1 \tag{6.23}$$

where $\Delta(k)$ is a sample estimate of the prediction error variance. For an adaptation strategy that minimizes the squared prediction error, Vijayan and Poor [1990] have presented a nonlinear gradient algorithm that is required a considerable amount of additional computational burden. It has been found that the nonlinear AIM method using the nonlinear gradient algorithm performs only slightly better than the nonlinear AIM method using the LMS algorithm when the interference is generated through an AR model. When the interference is a single–tone sinusoidal signal, its performance is inferior to that of the nonlinear AIM method using the LMS algorithm. Therefore, the utilization of the LMS algorithm for updating the adaptive filter coefficients is preferable. Figure 6.5 illustrates the general structure of the nonlinear AIM methods based on the ACM algorithm. In Fig. 6.5, $\{\tilde{x}(k)\}$ is defined as

$$\tilde{x}(k) = \hat{x}(k-m) + \rho_{k-m}(e(k-m)) \tag{6.24}$$

for $m = 1, 2, \ldots, M$.

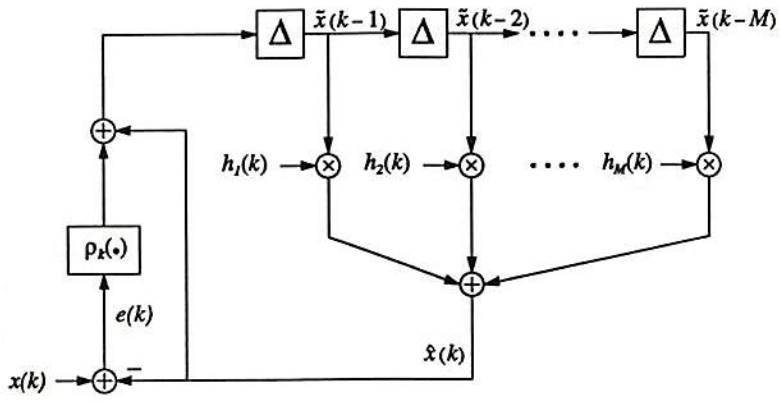

Figure 6.5: The general structure of the nonlinear prediction AIM methods based on the ACM filter.

6.4.3 Nonlinear AIM Method for Impulsive Noise Channels

Consider the $\epsilon-$mixture impulsive channel noise model using the first–order probability density described by

$$f_n(n) = (1-\epsilon)\, f_g(n) + \epsilon\, f_i(n) \tag{6.25}$$

where $\epsilon \in [0,1]$ and, $f_g(n)$ and $f_i(n)$ are probability density functions. The nominal noise density function $f_g(n)$ is usually described by a Gaussian density and the impulsive density $f_i(n)$ is taken to be a more heavy–tailed density such as a Laplacian or a Gaussian density with a large variance.

Using the $\epsilon-$mixture impulsive noise model with a Gaussian density with a large variance, Garth and Poor [1992] have obtained the nonlinear terms for AIM methods in the absence and presence of the DSSS SOI. In the absence of the DSSS signal, the prediction residual $\{e(k)\}$ is composed of the sum of two Gaussian random variables. Thus, the observation

prediction density becomes

$$f(\,x(k)\mid \mathbf{X}(k-1)\,) = (1-\epsilon)\,N_{\mathbf{HM}(k)\mathbf{H}^T+\sigma_n^2}(\,x(k)-\mathbf{H}\bar{\mathbf{u}}(k)\,) + \epsilon\,N_{\mathbf{HM}(k)\mathbf{H}^T+\sigma_n^2}(\,x(k)-\mathbf{H}\bar{\mathbf{u}}(k)\,). \quad (6.26)$$

Using this density, the nonlinear term $\{g_k(\,x(k)\,)\}$ can be obtained by

$$g_k(\,x(k)\,) = e(k)\left[\frac{\frac{1-\epsilon}{\sigma_a^3}\,e^{-e^2(k)/2\sigma_a^2}+\frac{\epsilon}{\sigma_b^3}\,e^{-e^2(k)/2\sigma_b^2}}{\frac{1-\epsilon}{\sigma_a}\,e^{-e^2(k)/2\sigma_a^2}+\frac{\epsilon}{\sigma_b}\,e^{-e^2(k)/2\sigma_b^2}}\right] \quad (6.27a)$$

where

$$\begin{aligned} e(k) &= x(k)-\mathbf{H}\bar{\mathbf{u}}(k)\\ \sigma_a^2 &= \mathbf{HM}(k)\mathbf{H}^T+\sigma_g^2\\ \sigma_b^2 &= \mathbf{HM}(k)\mathbf{H}^T+\kappa\sigma_g^2. \end{aligned} \quad (6.27b)$$

In the expression of (6.27b), the variance σ_g^2 is the variance of the nominal Gaussian noise and the parameter κ is the ratio of the impulsive noise variance and nominal Gaussian noise density defined by

$$\kappa = \frac{\sigma_i^2}{\sigma_g^2} \quad (6.28)$$

where σ_i^2 is the variance of the impulsive noise (Gaussian with a large variance). In the presence of the DSSS signal, the observation prediction density becomes

$$f(\,x(k)\mid \mathbf{X}(k-1)\,) = \frac{1}{2}\Big[(1-\epsilon)\left\{N_{\sigma_a^2}(\,e(k)-1\,)+N_{\sigma_a^2}(\,e(k)+1\,)\right\} + \epsilon\left\{N_{\sigma_b^2}(\,e(k)-1\,)+N_{\sigma_b^2}(\,e(k)+1\,)\right\}\Big] \quad (6.29)$$

where $\{e(k)\}$ is given in (6.27b). Using this density, the nonlinear term $\{g_k(\,x(k)\,)\}$ can be obtained by

$$g_k(\,x(k)\,) = \frac{N_k^g(\,x(k)\,)}{D_k^g(\,x(k)\,)} \quad (6.30a)$$

where

$$N_k^g(\,x(k)\,) = \frac{1-\epsilon}{\sigma_a^2}\Big[\,(\,e(k)-1\,)N_{\sigma_a^2}(\,e(k)-1\,) + (\,e(k)+1\,)N_{\sigma_a^2}(\,e(k)+1\,)\Big] + \frac{\epsilon}{\sigma_b^2}\Big[\,(\,e(k)-1\,)N_{\sigma_b^2}(\,e(k)-1\,) + (\,e(k)+1\,)N_{\sigma_b^2}(\,e(k)+1\,)\Big] \tag{6.30b}$$

$$D_k^g(\,x(k)\,) = (1-\epsilon)\Big[\,N_{\sigma_a^2}(\,e(k)-1\,) + N_{\sigma_a^2}(\,e(k)+1\,)\Big] + \epsilon\Big[\,N_{\sigma_b^2}(\,e(k)-1\,) + N_{\sigma_b^2}(\,e(k)+1\,)\Big]. \tag{6.30c}$$

Since we are interested in only the situation where the SOI exists, the prediction residual $\{e(k)\}$ is approximated to be composed of the sum of two Gaussian channel noises and a binary SOI. From the corresponding $\{g_k(e(k))\}$ given in (6.30a), Garth and Poor [1992] have used the following nonlinear function for AIM algorithm.

$$\rho_k(\,e(k)\,) = \Big[\,(1-\epsilon)\,\sigma_a^2 + \epsilon\,\sigma_b^2\,\Big]\;e(k)\,\frac{N_k^g(\,e(k)\,)}{D_k^g(\,e(k)\,)} \tag{6.31}$$

where $N_k^g(\,e(k)\,)$ and $D_k^g(\,e(k)\,)$ are given in (6.30b) and (6.30c), respectively. The variances σ_a^2 and σ_b^2 can be estimated using

$$\hat{\sigma}_a^2 = \Delta(k) + (1-\kappa)\,\epsilon\,\sigma_g^2 \tag{6.32a}$$

$$\hat{\sigma}_b^2 = \Delta(k) + (\kappa-1)\,(1-\epsilon)\,\sigma_g^2 \tag{6.32b}$$

where $\Delta(k)$ is the sample estimate of the prediction error variance which can be found recursively using

$$\Delta(k) = \Delta(k-1) + \gamma\Big[\,\|e(k)\|^2 - \Delta(k-1)\,\Big]. \tag{6.32c}$$

The LMS algorithm can be used to adjust the nonlinear AIM filter coefficients. Although the AIM method using the nonlinear function

given in (6.31) is supposed to work better in ϵ–mixture impulsive noise channel, Garth and Poor [1992] have found that although it requires more computations, it does not perform better than the nonlinear AIM method based on the nonlinear function (6.22b). Therefore, the nonlinear AIM method for Gaussian noise channels given in (6.22a) is preferable because of its robustness in ϵ–mixture impulsive noise channel environments and its low computational burden.

The advantages of the two nonlinear prediction AIM methods are:

- They can predict the narrowband interference better than the linear prediction algorithms.

Their disadvantages can be summarized as

- They require *a priori* information about the channel noise.
- Their computational burden is quite heavy.
- Their performance is sensitive to the choice of the step size parameters.

6.5 Nonlinear AIM Method for Generalized Gaussian Noise Channels

Assuming that the channel noise process $\{n(k)\}$ has a generalized Gaussian distribution, we describe a nonlinear AIM method based on the ACM algorithm [Shin and Nikias, 1994].

6.5.1 Description

When the channel noise process $\{n(k)\}$ is derived using a generalized Gaussian model described by (6.7a) with α and β parameters, the density

function of $\{v(k)\}$ in (6.6a) can be represented as

$$f_v(v) = \frac{1}{2}\left[\frac{\alpha}{2\beta\Gamma(\frac{1}{\alpha})}\, e^{-\left(\frac{|v-1|}{\beta}\right)^{\alpha}} + \frac{\alpha}{2\beta\Gamma(\frac{1}{\alpha})}\, e^{-\left(\frac{|v+1|}{\beta}\right)^{\alpha}}\right]. \tag{6.33}$$

To derive the ACM algorithm in a generalized Gaussian noise channel, it is necessary to calculate the density function $f(\, x(k) \mid \mathbf{X}(k-1)\,)$ which is of the form

$$f(\, x(k) \mid \mathbf{X}(k-1)\,) = f(\, \mathbf{H}\mathbf{u}(k) + v(k) \mid \mathbf{X}(k-1)\,). \tag{6.34}$$

Since $\{v(k)\}$ is independent of the observation vector $\mathbf{X}(k-1)$, $f(\, x(k) \mid \mathbf{X}(k-1)\,)$ is the density function of the sum of two independent random variables of which the first has a Gaussian density with mean $\mathbf{H}\bar{\mathbf{u}}(k)$ and variance $\mathbf{H}\mathbf{M}(k)\mathbf{H}^T$; and the second has the density described by (6.33). Since the probability density function of the sum of two statistically independent random variables is given by the convolution of their respective probability density functions, it becomes

$$\begin{aligned} f(\, x(k) \mid \mathbf{X}(k-1)\,) \;=\; & \frac{1}{2\sqrt{2\pi}\sigma_k}\left[\int_{-\infty}^{\infty} \zeta(v,\epsilon_k)\cdot gG(v-1,\alpha,\beta)\, dv\right. \\ & \left. + \int_{-\infty}^{\infty} \zeta(v,\epsilon_k)\cdot gG(v+1,\alpha,\beta)\, dv\right] \end{aligned} \tag{6.35a}$$

where

$$\begin{aligned} \sigma_k^2 &= \mathbf{H}\mathbf{M}(k)\mathbf{H}^T \\ \epsilon_k &= x(k) - \mathbf{H}\bar{\mathbf{u}}(k) \\ \zeta(v,\epsilon_k) &= \exp\left\{-\frac{(v-\epsilon_k)^2}{2\sigma_k^2}\right\}. \end{aligned} \tag{6.35b}$$

Note that $gG(x,\alpha,\beta)$ is symmetric and that the λ-th order moments of a generalized Gaussian process are satisfying [Gray, 1979]

$$\int_{-\infty}^{\infty} x^{\lambda}\, gG(x,\alpha,\beta)dx = \begin{cases} \frac{\Gamma(\frac{\lambda+1}{\alpha})}{\Gamma(\frac{1}{\alpha})}\,\beta^{\lambda}, & \text{for } \lambda = 2n \\ 0, & \text{for } \lambda = 2n+1 \end{cases} \tag{6.36}$$

where n is a nonnegative integer. By expanding the function $\zeta(v, \epsilon_k)$ using Taylor series about v_0, it becomes

$$\zeta(v, \epsilon_k) = \sum_{n=0}^{\infty} \frac{\zeta^{(n)}(v_0, \epsilon_k)}{n!}(v - v_0)^n \tag{6.37a}$$

where

$$\zeta^{(n)}(v_0, \epsilon_k) = \begin{cases} \zeta(v_0, \epsilon_k), & n = 0, \\ \frac{\partial^n \zeta(v, \epsilon_k)}{\partial v^n}\Big|_{v=v_0}, & n \geq 1. \end{cases} \tag{6.37b}$$

Since $\frac{\partial^n}{\partial v^n}\frac{(v-\epsilon_k)^2}{2\sigma_k^2} = 0$, for $n \geq 3$, the derivatives of $\zeta(v_0, \epsilon_k)$ satisfy the following recursive equation (see Appendix A):

$$\zeta^{(n)}(v_0, \epsilon_k) = -\frac{(v_0 - \epsilon_k)}{\sigma_k^2} \cdot \zeta^{(n-1)}(v_0, \epsilon_k) - \frac{(n-1)}{\sigma_k^2} \cdot \zeta^{(n-2)}(v_0, \epsilon_k). \tag{6.38}$$

When each n–th order derivative of $\zeta(v_0, \epsilon_k)$ is represented with respect to v_0, it becomes (also see Appendix A)

$$\frac{\partial^n \zeta(v_0, \epsilon_k)}{\partial v^n} = \begin{cases} \zeta(v_0, \epsilon_k) \cdot \sum_{i=1}^{m+1} \frac{a_i^{(m)} (v_0 - \epsilon_k)^{2(i-1)}}{\sigma_k^{2(m+i-1)}}, & \text{for } n = 2m \\ \zeta(v_0, \epsilon_k) \cdot \sum_{i=1}^{m+1} \frac{b_i^{(m)} (v - \epsilon_k)^{2i-1}}{\sigma_k^{2(m+i)}}, & \text{for } n = 2m+1 \end{cases} \tag{6.39}$$

where $m = 0, 1, 2, \ldots$. The coefficients $\{a_i^{(m)}\}$ and $\{b_i^{(m)}\}$ are constants with $a_{m+1}^{(m)} = 1$ and $b_{m+1}^{(m)} = -1$. Appendix A presents the values of coefficient vectors $\mathbf{a}^{(m)}$ and $\mathbf{b}^{(m)}$ for $m = 0, 1, 2, 3, 4, 5$. There seems, unfortunately, no explicit relationship between coefficients $a_i^{(m_1)}$ and $a_j^{(m_2)}$ or between coefficients $b_i^{(m_1)}$ and $b_j^{(m_2)}$ for any m_1 and m_2.

Expanding $\zeta(v, \epsilon_k)$ about 1 and -1 and using the property that odd moments of a generalized Gaussian process are identically zero, (6.35a) can be rewritten as

$$f(\, x(k) \mid \mathbf{X}(k-1)\,) = \frac{1}{2\sqrt{2\pi}\sigma_k\,\Gamma\left(\frac{1}{\alpha}\right)} \sum_{n=0}^{\infty} \frac{\Gamma\left(\frac{2n+1}{\alpha}\right)\beta^{2n}}{(2n)!} \left[\, \zeta^{(2n)}(1,\epsilon_k) + \zeta^{(2n)}(-1,\epsilon_k) \right] \quad (6.40)$$

where $\{\epsilon_k\}$ is given in (6.35b). Although the derivatives $\zeta^{(2n+1)}(\pm 1, \epsilon_k)$ are not needed for the representation of $f(\, x(k) \mid \mathbf{X}(k-1)\,)$, they are needed to get the expressions of even derivatives $\zeta^{(2n)}(\pm 1, \epsilon_k)$.

To get a nonlinear AIM scheme in a generalized Gaussian noise channel, an expression for the following nonlinear function should be found.

$$\rho_k(\, x(k)\,) = -\frac{\frac{\partial}{\partial x(k)} f(\, x(k) \mid \mathbf{X}(k-1)\,)}{f(\, x(k) \mid \mathbf{X}(k-1)\,)}. \quad (6.41)$$

After some manipulations presented in Appendix B, the nonlinear function becomes

$$\rho_k(\epsilon_k) = -\frac{\sum_{n=0}^{\infty} \frac{\Gamma\left(\frac{2n+1}{\alpha}\right)\beta^{2n}}{(2n)!} \left[\xi_n(1,\epsilon_k) + \xi_n(-1,\epsilon_k)\right]}{\sum_{n=0}^{\infty} \frac{\Gamma\left(\frac{2n+1}{\alpha}\right)\beta^{2n}}{(2n)!} \left[\zeta^{(2n)}(1,\epsilon_k) + \zeta^{(2n)}(-1,\epsilon_k)\right]} \quad (6.42a)$$

where

$$\zeta^{(2n)}(\pm 1, \epsilon_k) = \zeta(\pm 1, \epsilon_k) \sum_{i=1}^{n+1} \frac{a_i^{(n)} (\pm 1 - \epsilon_k)^{2(i-1)}}{\sigma_k^{2(n+i-1)}} \quad (6.42b)$$

$$\xi_n(\pm 1, \epsilon_k) = \zeta(\pm 1, \epsilon_k) \left[\sum_{i=1}^{n+1} \frac{a_i^{(n)} (\pm 1 - \epsilon_k)^{2i-1}}{\sigma_k^{2(n+i)}} - \sum_{i=2}^{n+1} \frac{2(i-1) a_i^{(n)} (\pm 1 - \epsilon_k)^{2i-3}}{\sigma_k^{2(n+i-1)}} \right]. \quad (6.42c)$$

Using the nonlinear function $\{\rho_k(\, e(k)\,)\}$ with a finite order in (6.42a), one can get a nonlinear transversal AIM algorithm for $\{\hat{x}(k)\}$ that is the prediction of the narrowband interference.

$$\hat{x}(k) = \sum_{m=1}^{M} h_m(k-1) \left[\, \hat{x}(k-m) + \rho_{k-m}(\, e(k-m)\,) \right] \quad (6.43)$$

where M denotes the number of taps of the AIM algorithm. In this expression $\{h_m(k-1),\ m = 1, 2, \ldots, M\}$ is the filter coefficients updated by the LMS algorithm given in (6.14a). Then, the system output (restored SOI) of the nonlinear AIM scheme $\{e(k)\}$ are given by

$$e(k) = x(k) - \hat{x}(k). \tag{6.44}$$

The structure of the nonlinear AIM scheme is illustrated in Fig. 6.5 with a different structure of the nonlinear term $\{\rho(\, e(k)\,)\}$.

6.5.2 Implementation

The nonlinear AIM scheme is implemented by using order up to eight. Note that only even-order expressions are required for implementation because the odd-order terms vanish. Then, the derivatives $\zeta^{(2n)}(\cdot)$ can be given as

$$\begin{aligned}
\zeta^{(0)}(v_0, \epsilon_k) &= e^{-\frac{(v_0-\epsilon_k)^2}{2\sigma_k^2}} \equiv \zeta(v_0, \epsilon_k) \\
\zeta^{(2)}(v_0, \epsilon_k) &= \zeta(v_0, \epsilon_k)\left(-\frac{1}{\sigma_k^2} + \frac{(v_0-\epsilon_k)^2}{\sigma_k^4}\right) \\
\zeta^{(4)}(v_0, \epsilon_k) &= \zeta(v_0, \epsilon_k)\left(\frac{3}{\sigma_k^4} - \frac{6(v_0-\epsilon_k)^2}{\sigma_k^6} + \frac{(v_0-\epsilon_k)^4}{\sigma_k^8}\right) \\
\zeta^{(6)}(v_0, \epsilon_k) &= \zeta(v_0, \epsilon_k)\left(-\frac{15}{\sigma_k^6} + \frac{45(v_0-\epsilon_k)^2}{\sigma_k^8} - \frac{15(v_0-\epsilon_k)^4}{\sigma_k^{10}}\right. \\
&\qquad \left. + \frac{(v_0-\epsilon_k)^6}{\sigma_k^{12}}\right) \\
\zeta^{(8)}(v_0, \epsilon_k) &= \zeta(v_0, \epsilon_k)\left(\frac{105}{\sigma_k^8} - \frac{420(v_0-\epsilon_k)^2}{\sigma_k^{10}} + \frac{210(v_0-\epsilon_k)^4}{\sigma_k^{12}}\right. \\
&\qquad \left. - \frac{28(v_0-\epsilon_k)^6}{\sigma_k^{14}} + \frac{(v_0-\epsilon_k)^8}{\sigma_k^{16}}\right)
\end{aligned} \tag{6.45}$$

where $v_0 = 1$ or -1. Note that the coefficients of $\zeta^{(n)}(v_0, \epsilon_k)$ are given in Appendix A. Using (6.45), an approximated expression of the denom-

inator, $D_k^g(\epsilon_k)$, of the nonlinear function $\{\rho_k(\epsilon_k)\}$ can be obtained by

$$\begin{aligned} D_k^g(\epsilon_k) &= f(x(k) \mid \mathbf{X}(k-1)) \\ &\approx \sum_{n=0}^{4} \frac{\Gamma\left(\frac{2n+1}{\alpha}\right)}{(2n)!} \beta^{2n} \left[\zeta^{(2n)}(1,\epsilon_k) + \zeta^{(2n)}(-1,\epsilon_k)\right] \end{aligned} \quad (6.46)$$

where $\{\zeta^{(2n)}(v_0,\epsilon_k),\ 0 \le n \le 4\}$ are given in (6.45). Again, with (6.45), an approximated expression of the numerator, $N_k^g(\epsilon_k)$, which is the derivatives of the denominator, of the nonlinear function $\{\rho_k(\epsilon_k)\}$ can be obtained as follows.

$$\begin{aligned} N_k^g(\epsilon_k) &= \frac{\partial}{\partial x(k)} f(x(k) \mid \mathbf{X}(k-1)) \\ &\approx \sum_{n=0}^{4} \frac{\Gamma\left(\frac{2n+1}{\alpha}\right)\beta^{2n}}{(2n)!} \left[\xi_n(1,\epsilon_k) + \xi_n(-1,\epsilon_k)\right] \end{aligned} \quad (6.47a)$$

where

$$\begin{aligned} \xi_0(v_0,\epsilon_k) &= \zeta(v_0,\epsilon_k)\left(\frac{(v_0-\epsilon_k)}{\sigma_k^2}\right) \\ \xi_1(v_0,\epsilon_k) &= \zeta(v_0,\epsilon_k)\left(-\frac{3(v_0-\epsilon_k)}{\sigma_k^4} + \frac{(v_0-\epsilon_k)^3}{\sigma_k^6}\right) \\ \xi_2(v_0,\epsilon_k) &= \zeta(v_0,\epsilon_k)\left(\frac{15(v_0-\epsilon_k)}{\sigma_k^6} - \frac{10(v_0-\epsilon_k)^3}{\sigma_k^8} + \frac{(v_0-\epsilon_k)^5}{\sigma_k^{10}}\right) \\ \xi_3(v_0,\epsilon_k) &= \zeta(v_0,\epsilon_k)\left(-\frac{105(v_0-\epsilon_k)}{\sigma_k^8} + \frac{105(v_0-\epsilon_k)^3}{\sigma_k^{10}}\right. \\ &\qquad \left. - \frac{21(v_0-\epsilon_k)^5}{\sigma_k^{12}} + \frac{(v_0-\epsilon_k)^7}{\sigma_k^{14}}\right) \\ \xi_4(v_0,\epsilon_k) &= \zeta(v_0,\epsilon_k)\left(\frac{945(v_0-\epsilon_k)}{\sigma_k^{10}} - \frac{1260(v_0-\epsilon_k)^3}{\sigma_k^{12}}\right. \\ &\qquad \left. + \frac{378(v_0-\epsilon_k)^5}{\sigma_k^{14}} - \frac{36(v_0-\epsilon_k)^7}{\sigma_k^{16}} + \frac{(v_0-\epsilon_k)^9}{\sigma_k^{18}}\right) \end{aligned} \quad (6.47b)$$

with $v_0 = 1$ or -1. The coefficients of $\{\xi_n(v_0,\epsilon_k)\}$ is obtained by the coefficient vectors $\mathbf{a}^{(n)}$ in Appendix A and the expression in (6.42c).

Then, the nonlinear function $\{\rho_k(\, e(k)\,)\}$, where $\{e(k) = \epsilon_k\}$, becomes

$$\rho_k(\, e(k)\,) = -\frac{N_k^g(\, e(k)\,)}{D_k^g(\, e(k)\,)}. \tag{6.48}$$

Although the nonlinear AIM scheme is structured here using order up to eight, one may use the nonlinear AIM scheme in a generalized Gaussian noise channel with a lower/higher order.

The advantages of the nonlinear prediction AIM method for generalized Gaussian noise channels are:

- Since it takes into account the distribution of the channel noise as well as the predictability of narrowband interferences, it can predict the narrowband interference very well.
- Its corresponding nonlinear terms in the AIM methods contribute to the improvement of its performance in mitigating narrowband interference.
- Since it has been developed for general values of α and β of the generalized Gaussian distribution, the algorithm can be applied to any channels characterized by a member distribution of the generalized Gaussian distributions family.

It shares the same disadvantages of the two nonlinear prediction AIM methods for Gaussian and ϵ–mixture impulsive noise channels described in the previous sections.

Example 6.3

Considering the same SOI and single–tone sinusoidal interference as those in Example 6.1, let us use the two nonlinear AIM methods developed for Gaussian and generalized Gaussian noise channels in order

to mitigate the single–tone narrowband interference. When the variance of the Laplace channel noise $\{n(k)\}$ is 0.01, the averaged SINR improvements over 10 independent runs obtained by the nonlinear AIM algorithms are summarized in Table 6.3. Note that the new nonlinear AIM algorithm for generalized Gaussian noise channels is implemented by using order up to four.

Comparing Table 6.3 with Table 6.1, we can easily observe that the nonlinear AIM algorithms performs much better than the linear AIM algorithms. Although the performance of the two nonlinear AIM algorithms seems similar, the nonlinear AIM algorithm for generalized Gaussian noise channels performs a little bit better than the nonlinear AIM algorithm for Gaussian noise channels in all the SINR cases. The difference between the two SINR improvement tends to increases as the input SINR becomes large. As another way to compare the performance between the two nonlinear AIM algorithms, we illustrate the error curves between the SOI and the system output (recovered SOI) obtained by each nonlinear AIM algorithm: i.e., $\{e(k) - s(k)\}$. Fig. 6.6 illustrates typical error curves when the input SINR is -10 dB. In Fig. 6.6, it can be easily observed that several large outliers whose magnitudes are greater and closer to unity after convergence exist in the system output obtained by the nonlinear AIM method for Gaussian noise channels. Note again that these outliers cause signal detection errors. On the other hand, although its SINR improvement is smaller, the error curve obtained by the nonlinear AIM algorithm for generalized Gaussian noise channels does not have large outliers that cause the detection error.

These experiments demonstrate that the nonlinear AIM algorithm for generalized Gaussian noise channel can mitigate single–tone sinusoidal narrowband interference more effectively than the nonlinear AIM algorithm developed for a Gaussian noise channel in the sense that the

Table 6.3: The SINR Improvement of the Nonlinear AIM (NAIM) Methods for Gaussian (G) and Generalized Gaussian (GG) Noise Channels in the Cases of a Single Sinusoidal Interference in Laplace Noise Channel with Variance 0.01.

Input SINR (dB)	SINR Improvement (dB)	
	NAIM for G Channel	NAIM for GG Channel
−15	29.8276	30.4248
−10	24.4617	25.5794
−5	18.8120	20.2001
0	12.6718	14.7004
5	6.5053	10.4891

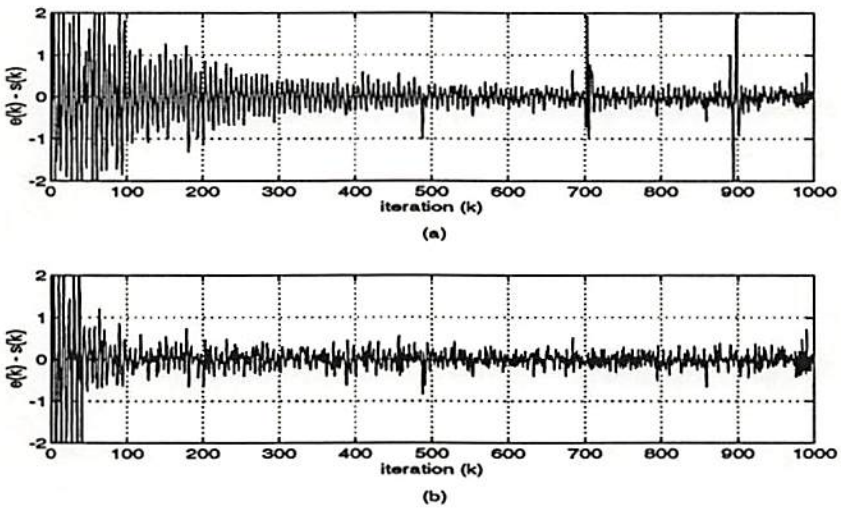

Figure 6.6: When the narrowband interference is a sinusoidal; the variance of Laplace channel noise is 0.01; and the input SINR is −10 dB, the error curves between the SOI and the system outputs obtained by (a) the nonlinear AIM method for Gaussian noise channels and (b) the new nonlinear AIM method with order four for generalized Gaussian noise channels.

former nonlinear AIM method results in similar SINR improvement and less outliers than the latter nonlinear AIM method, when the channel noise is a member of the generalized Gaussian distributions with known α.

Example 6.4

Now, let us consider the same SOI and AR interference as those in Example 6.2. When the variance of the Laplace channel noise $\{n(k)\}$ is 0.01, the averaged SINR improvements over 10 independent runs obtained by the two nonlinear AIM methods are summarized in Table 6.4. Note that the new nonlinear AIM algorithm for generalized Gaussian noise channels is implemented by using order up to four.

Again, the nonlinear AIM algorithms perform much better than the linear AIM algorithms (Table 6.2) in the AR interference case. Note that the SINR improvement obtained by the nonlinear AIM algorithm for generalized Gaussian noise channels is a little bit larger than that obtained by the nonlinear AIM algorithm for Gaussian noise channels in all five input SINR cases. Fig. 6.7 illustrates typical error curves $\{e(k) - s(k)\}$, when the input SINR is -10 dB. The same phenomenon can be observed as that in Fig. 6.6. No large outliers exist in Fig. 6.7 (b) illustrated the error curve obtained by the nonlinear AIM method for generalized Gaussian noise channels. These experiments demonstrate that the nonlinear AIM method for generalized Gaussian noise channels can effectively mitigate the narrowband interferences generated through an AR process which is considered as a more physically accurate broad class of narrowband interference.

Note that the performance of the nonlinear AIM method for generalized Gaussian noise channels in a Gaussian noise channel is very similar

Table 6.4: The SINR Improvement of the Nonlinear AIM (NAIM) Methods for Gaussian (G) and Generalized Gaussian (GG) Noise Channels in the Cases of AR Interference in Laplace Noise Channel with Variance 0.01.

Input SINR (dB)	SINR Improvement (dB)	
	NAIM for G Channel	NAIM for GG Channel
−15	29.4889	31.2914
−10	25.2374	27.2745
−5	19.6121	22.0615
0	14.6878	17.3222
5	11.1369	12.1165

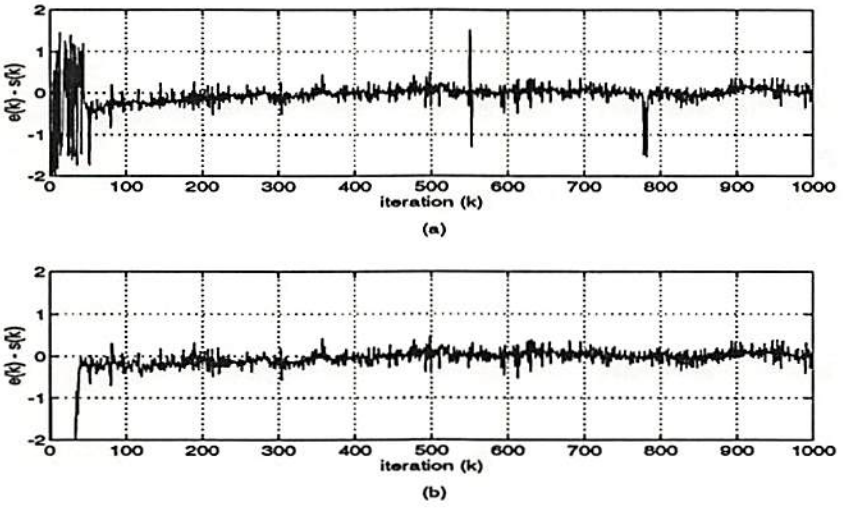

Figure 6.7: When the narrowband interference is generated through a AR model; the variance of Laplace channel noise is 0.01; and the input SINR is −10 dB, the error curves between the SOI and the system outputs obtained by (a) the nonlinear AIM method for Gaussian noise channels and (b) the new nonlinear AIM method with order four for generalized Gaussian noise channels.

to the given performance in the Laplace noise channel for the single sinusoidal and AR interferences.

Example 6.5

In the previous examples, it is assumed that the value of α is exactly known. Since this assumption is not always true, it is important to investigate the algorithm's performance dependence on the estimate $\hat{\alpha}$. Considering the previous single narrowband interference cases, the performance of the nonlinear AIM algorithm for generalized Gaussian channels is investigated with different estimated values $\hat{\alpha}$. *Note that the true value of α is unity* (Laplace noise channel). Fixing all the other parameters as in the previous two examples, we vary estimated $\hat{\alpha}$ from 0.5 to 2.0 by interval of 0.1 and calculate the nonlinear function with the value of $\hat{\alpha}$. The three input SINRs are considered: -15 dB, -10 dB, and -5 dB.

Fig. 6.8 (a) illustrates the SINR improvements with different values of $\hat{\alpha}$, when the narrowband interference is a single–tone interference whose normalized frequency is 0.15. Fig. 6.8 (b) illustrates the corresponding SINR improvements, when the narrowband interference is generated by the same AR filter. Note that the solid line corresponds to the SINR improvement when the input SINR is -15 dB and that the dashed and dash–dotted lines correspond to the SINR improvement, respectively, when the input SINRs are -10 dB and -5 dB. Each "$*$" point is obtained by averaging the results from 10 independent runs. As we can observe in Fig. 6.8, there is no large difference among the SINR improvements obtained with different values of $\hat{\alpha}$ except for the cases of $\hat{\alpha} \leq 0.6$. The nonlinear AIM algorithm has diverged with $\hat{\alpha} \leq 0.6$. These experiments illustrate that the approximate nonlinear AIM algorithm for generalized

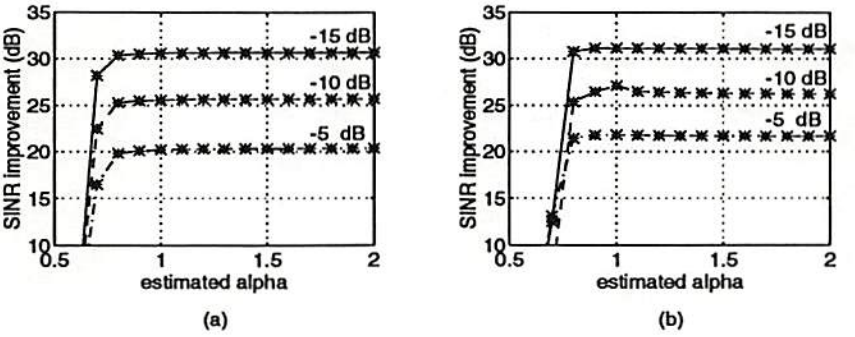

Figure 6.8: The channel noise distribution is Laplace with variance 0.01. The SINR improvements obtained by the nonlinear AIM algorithm with different estimated values of α for generalized Gaussian channels, when the narrowband interference is (a) a single–tone sinusoidal signal and (b) an AR signal.

Gaussian channels can perform well even though the estimate of α for the given channel may have a small bias.

Example 6.6

In the previous examples, we fixed the value of β at 0.15. Since the value of β is dependent on the given channel noise variance, fixing the value of β at a constant may influence the performance of the nonlinear AIM algorithm for generalized Gaussian channels. Thus, it is also important to investigate the algorithm's performance dependence on the estimated value of β. Considering the previous single narrowband interference cases, the performance of the nonlinear AIM algorithm for generalized Gaussian channels is investigated with different constant estimates $\hat{\beta}$. Fixing all the other parameters at the optimal values as in the previous two examples, we vary the value of $\hat{\beta}$ from 0.01 to 0.45 by interval of 0.02 and calculate the nonlinear function with the estimated

value of $\hat{\beta}$. The three input SINRs are considered: -15 dB, -10 dB, and -5 dB. *Note that the true value of β is about 0.07* when the variance of the Laplace channel noise is 0.01.

Fig. 6.9 (a) illustrates the SINR improvements with different values of $\hat{\beta}$, when the narrowband interference is a single–tone interference whose normalized frequency is 0.15. Fig. 6.9 (b) illustrates the corresponding SINR improvements, when the narrowband interference is generated by the AR system. Note that the solid line corresponds to the SINR improvement when the input SINR is -15 dB and that the dashed and dash–dotted lines correspond to the SINR improvement, respectively, when the input SINRs are -10 dB and -5 dB. Each "$*$" point is obtained by averaging the results from 10 independent runs. As we can observe in Fig. 6.9, the SINR improvement begins to decrease around $\hat{\beta} = 0.3$ in the both cases. Also note that when the value of $\hat{\beta}$ is greater than 0.3, the SINR improvement for an AR interference deteriorates more than that for a sinusoidal interference. These experiments illustrate that the

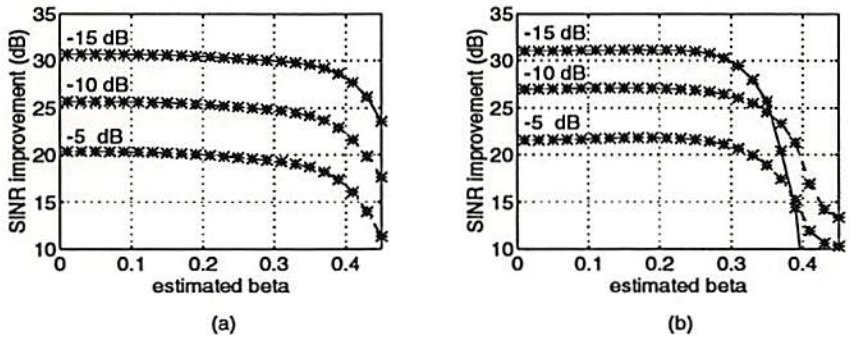

Figure 6.9: The channel noise distribution is Laplace with variance 0.01. The SINR improvements obtained by the nonlinear AIM algorithm with different values of constant β for generalized Gaussian channels, when the narrowband interference is (a) a single–tone sinusoidal signal and (b) an AR signal.

approximate nonlinear AIM algorithm for generalized Gaussian channels can still perform well even when an estimated value of β has a small bias.

6.6 Summary

This chapter described linear and nonlinear AIM schemes based on the ACM filter. Their performance was described in terms of the SINR improvement obtained by each AIM method in a Laplace noise channel environment, and when the SOI is a DSSS signal and the narrowband interference is either a single–tone sinusoidal signal or a signal through AR model. It has been observed that the nonlinear AIM methods perform much better than the linear AIM methods in both the two interference environments. Although there is no big difference between the SINR improvement obtained by the two nonlinear AIM methods, the nonlinear AIM method for generalized Gaussian noise channels can mitigate narrowband interference more effectively than the nonlinear AIM algorithm for Gaussian noise channels in the sense that the latter algorithm appears to contain more outliers whose magnitude is close to or larger than unity. These outliers can cause errors in the decoded SOIs.

Chapter 7

Impulsive Interference Environments

7.1 Introduction

Traditionally, the Gaussian assumption has dominated the engineering literature due to the fact that it can be justified by the central limit theorem and can lead to analytically tractable solutions for most given problems.There are, however, many phenomena in real world which are non–Gaussian in nature. The non–Gaussian environment may result in significant performance degradation for systems optimized under the Gaussian assumption. In addition, it has to be stated that the non–Gaussian environment usually leads to nonlinear systems which require more complexity to obtain a solution. Therefore, there is a trade–off between complexity and accuracy. When the loss of resolution or accuracy due to the Gaussian assumption in a non–Gaussian environment cannot be tolerated, more realistic non–Gaussian statistical model must be used.

The main difference between the Gaussian distribution and the non–Gaussian stable distribution is their tails. The tail of a non–Gaussian

stable density is heavier than that of a Gaussian density. This fact is one of the main reasons why the non–Gaussian stable distribution is suitable for describing impulsive phenomena. The non–Gaussian distribution can provide useful models for phenomena observed in many different fields, such as engineering, physics, hydrology, etc. [Lévy, 1925].

7.2 Stable Distributions

This section briefly reviews the definition of stable distributions and their statistical properties. For those who are interested in learning more about stable distributions, we refer them to the tutorial paper and book by [Shao and Nikias, 1993; 1995].

7.2.1 Univariate Stable Distribution

A univariate distribution function $F(x)$ is defined to be *stable* if and only if (iff) its characteristic function has the following form

$$\phi(t) = \exp\{\, jat - \gamma\,|t|^{\alpha}\ [\,1 + j\beta\,\mathrm{sign}(t)\,w(t,\alpha)\,]\,\} \tag{7.1a}$$

where

$$\mathrm{sign}(t) \;=\; \begin{cases} 1 & \text{for } t > 0 \\ 0 & \text{for } t = 0 \\ -1 & \text{for } t < 0 \end{cases} \tag{7.1b}$$

$$w(t,\alpha) \;=\; \begin{cases} \tan\left(\frac{\alpha\pi}{2}\right) & \text{for } \alpha \neq 0 \\ \frac{2}{\pi}\log|t| & \text{for } \alpha = 1. \end{cases} \tag{7.1c}$$

A stable distribution is completely determined by the following four parameters.

1. Location parameter a: It corresponds to the location of the median value of the density function and takes any real values.

2. Index of skewness β: It can take values in the interval $-1 \leq \beta \leq 1$. When $\beta = 0$, the distribution becomes symmetric about the center a. If $\alpha \neq 1$, the cases $\beta > 0$ and $\beta < 0$ correspond to left–skewness and right–skewness, respectively. The direction of skewness is reversed if $\alpha = 1$.

3. Characteristic exponent α: It is a shape parameter whose values are satisfying $0 < \alpha \leq 2$. It measures the "thickness" of the tails of the density function. The larger the value of α is, the less likely it is to observe values of the random variable which are far from its central location. The smaller α is, the heavier are the tails of the density function.

4. Scale parameter γ (also called the dispersion): It can be any positive number and behaves like the variance. When $\alpha = 2$, it becomes a half of the variance of the Gaussian distribution.

A stable distribution with characteristic exponent α is called α–stable. Symmetric stable distributions with characteristic exponent α are called *symmetric α–stable* (SαS). A stable distribution is called *standard* if $a = 0$ and $\gamma = 1$. By taking the inverse Fourier transform of the characteristic function, it is easy to show that the standard stable density function is given by

$$f(x;\alpha,\beta) = \frac{1}{\pi}\int_0^\infty \exp(-t^\alpha)\,\cos[\,xt + \beta t^\alpha w(t,\alpha)\,]\,dt. \qquad (7.2)$$

Note that $f(x;\alpha,\beta) = f(-x;\alpha,-\beta)$. It can also be shown that the probability density functions of stable distributions are bounded and have derivatives of arbitrary orders. Unfortunately, no close–form expressions exist for general density except for the special densities: Gaussian ($\alpha = 2$), Cauchy ($\alpha = 1$ and $\beta = 0$), and Pearson ($\alpha = \frac{1}{2}$ and $\beta = -1$) distributions. The standard stable density function can be expanded

into convergent series. Using power series expansion, the standard SαS density function can be given as

$$f_\alpha(x)=\begin{cases} \frac{1}{\pi}\sum_{k=1}^{\infty}\frac{(-1)^{k-1}}{k!}\Gamma(\alpha k+1)\,|x|^{-\alpha k-1}\sin\left(\frac{k\alpha\pi}{2}\right), & 0<\alpha<1 \\ \frac{1}{\pi(x^2+1)}, & \alpha=1 \\ \frac{1}{\pi\alpha}\sum_{k=0}^{\infty}\frac{(-1)^k}{2k!}\Gamma\left(\frac{2k+1}{\alpha}\right)x^{2k}, & 1<\alpha<2 \\ \frac{1}{2\sqrt{\pi}}e^{-\frac{x^2}{4}}, & \alpha=2 \end{cases} \tag{7.3a}$$

where

$$\Gamma(x)=\int_0^\infty t^{x-1}\,e^{-t}dt. \tag{7.3b}$$

The SαS densities are smooth, unimodal, symmetric with respect to the median, and bell–shaped. Most importantly, unlike the Gaussian density which has exponentially decaying tails, the stable densities have algebraic tails.

7.2.2 Basic Properties

Two of the most important properties of the stable distribution are the stability property and the generalized central limit theorem [Shao, 1993; Shao and Nikias, 1993].

Theorem 7.1 (Stability Property): *A random variable x is stable iff for any independent random variables x_1 and x_2 with the same distribution as x, and for arbitrary constants a_1 and a_2, there exist constant a and b such that*

$$a_1x_1+a_2x_2\overset{d}{=}ax+b \tag{7.4}$$

where the notation $x\overset{d}{=}y$ means that x and y have the same distribution.

More general statement is that if $x_1, x_2, \ldots, x_n$ are independent and follow stable laws with the same (α, β), then all linear combinations of the form $\sum_{k=1}^{n} a_k x_k$ are stable with the same parameters α and β.

Theorem 7.2 (Generalized Central Limit Theorem): *x is the limit in distribution of normalized sums of the form*

$$s_n = \frac{x_1 + x_2 + \cdots + x_n}{a_n} - b_n \tag{7.5}$$

where $x_1, x_2, \ldots$ are i.i.d. and $a_n \to \infty$, iff x is stable.

As a consequence of the stability property, stable distributions are the only possible limit distributions for sums of i.i.d. random variables. It can be shown that for a non–Gaussian α–stable random variable x with zero location parameter and dispersion γ,

$$\lim_{t \to \infty} t^{\alpha} \Pr\{|x| > t\} = \gamma C(\alpha) \tag{7.6}$$

where $C(\alpha)$ is a positive constant depending on α. Thus, stable density functions have algebraic tails. The important consequence is the *nonexistence* of the second–order moment of stable distributions. Let x be an α–stable random variable. If $0 < \alpha < 2$, then

$$\mathbf{E}\{\,|x|^p\,\} = \infty, \quad \text{if } p \geq \alpha \tag{7.7a}$$

and

$$\mathbf{E}\{\,|x|^p\,\} < \infty, \quad \text{if } 0 \leq p \leq \alpha. \tag{7.7b}$$

If $\alpha = 2$, then

$$\mathbf{E}\{\,|x|^p\,\} < \infty, \quad \text{for all } p \geq 0. \tag{7.8}$$

Hence for $0 < \alpha \leq 1$, α–stable distributions have no finite first or higher order moments; for $1 < \alpha < 2$, they have finite first–order moments and all the fractional moments of order p where $p < \alpha$; for $\alpha = 2$, all moments exist.

7.2.3 Fractional Lower Order Moments and Covariations

All moments of order less than α are called the fractional lower order moments (FLOM) [Shao and Nikias, 1993].

Theorem 7.3 *Let x be a SαS random variable with zero location parameter and dispersion γ. Then,*

$$\mathbf{E}\{|x|^p\} = C(p,\alpha)\,\gamma^{\frac{p}{\alpha}} \qquad \text{for } 0 < p < \alpha \tag{7.9a}$$

where

$$C(p,\alpha) = \frac{2^{p+1}\,\Gamma\left(\frac{p+1}{2}\right)\,\Gamma\left(-\frac{p}{\alpha}\right)}{\alpha\sqrt{\pi}\,\Gamma\left(-\frac{p}{2}\right)} \tag{7.9b}$$

depends only on α and p, not on x.

Let x be a SαS random variable with $a = 0$ and $\gamma > 0$. Then, the norm of x is defined as

$$\|x\|_\alpha = \begin{cases} \gamma^{\frac{1}{\alpha}} & \text{for } 1 \le \alpha \le 2 \\ \gamma & \text{for } 0 < \alpha < 1. \end{cases} \tag{7.10}$$

Let x and y be jointly SαS random variables, then the distance between x and y is defined as

$$d_\alpha(x,y) = \|x - y\|_\alpha\,. \tag{7.11}$$

The distance d_α between two SαS random variables measures the pth order moment of the difference of these two random variables for any $0 < p < \alpha$. It should be mentioned that convergence in distance is equivalent to convergence in probability.

Let x and y be jointly SαS random variables with $1 < \alpha \le 2$, zero location parameters, and dispersions γ_x and γ_y, respectively. Then the covariation of x with y is defined by

$$[x,y]_\alpha = \frac{\mathbf{E}\{xy^{<p-1>}\}}{\mathbf{E}\{|y|^p\}}\,\gamma_y \tag{7.12a}$$

where for any real number z and $a \geq 0$,

$$z^{<a>} = |z|^a \operatorname{sign}(z). \tag{7.12b}$$

The covariation coefficient of x with y is defined by

$$\lambda_{x,y} = \frac{[x,y]_\alpha}{[y,y]_\alpha} = \frac{\mathbf{E}\{xy^{<p-1>}\}}{\mathbf{E}\{|y|^p\}} \quad \text{for any } 1 \leq p < \alpha \tag{7.13a}$$

where

$$[x,x]_\alpha = \gamma_x = \|x\|_\alpha^\alpha \qquad [y,y]_\alpha = \gamma_y = \|y\|_\alpha^\alpha . \tag{7.13b}$$

7.2.4 Conditional Expectation and Linear Regression

Let $x_0, x_1, \ldots, x_n$ be jointly SαS random variables with $1 < \alpha \leq 2$. The regression of x_0 in terms of $x_1, x_2, \ldots, x_n$ is the conditional expectation $\mathbf{E}\{x_0 \mid x_1, x_2, \ldots, x_n\}$. The following theorem states a necessary and sufficient condition for the regression to be linear.

Theorem 7.4 *If $x_0, x_1, \ldots, x_n$ are jointly SαS random variables with $1 < \alpha \leq 2$. Then*

$$\mathbf{E}\{x_0 \mid x_1, x_2, \ldots, x_n\} = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n \tag{7.14a}$$

iff for all $b_1, b_2, \ldots, b_n$,

$$\left[x_0 - a_1 x_1 - \cdots - a_n x_n,\ b_1 x_1 + \cdots + b_n x_n \right]_\alpha = 0. \tag{7.14b}$$

If the regression is linear, then the coefficients $\{a_1, a_2, \ldots, a_n\}$ are unique iff $x_1, x_2, \ldots, x_n$ are linearly independent elements in the space of integrable random variables.

Theorem 7.5 *If $x_0, x_1, \ldots, x_n$ are jointly SαS random variables and if $x_1, x_2, \ldots, x_n$ are independent and nondegenerate, then*

$$\mathbf{E}\{x_0 \mid x_1, x_2, \ldots, x_n\} = \lambda_{01} x_1 + \lambda_{02} x_2 + \cdots + \lambda_{0n} x_n \tag{7.15}$$

where λ_{0i} is the covariation coefficient of x_0 with x_i, for $i = 1, 2, \ldots, n$.

It is important to have good estimators for covariations (covariation coefficients) due to the fact that they play important role in certain problems as the correlations do for second–order random variables. For the independent observations $\{(x_1, y_1), \ldots, (x_N, y_N)\}$, one method for the FLOM estimator is given by [Shao and Nikias, 1993]

$$\hat{\lambda}_{FLOM} = \frac{\sum_{n=1}^{N} x_n \ |y_n|^{p-1} \ \text{sign}(y_n)}{\sum_{n=1}^{N} |y_n|^p} \tag{7.16}$$

for some $1 \leq p < \alpha$. Note that a computationally efficient choice is $p = 1$. Another estimator is the screened ratio (SR) estimator proposed by Kanter and Steiger [1974], which is strongly consistent. It is of the form

$$\hat{\lambda}_{SR} = \frac{\sum_{n=1}^{N} x_n \, y_n^{-1} \, \chi_{y_i}}{\sum_{n=1}^{N} \chi_{y_i}} \tag{7.17a}$$

where the random variable χ_y is defined

$$\chi_y = \begin{cases} 1 & \text{if } c_1 < |y| < c_2 \\ 0 & \text{otherwise} \end{cases} \tag{7.17b}$$

for arbitrary constants $0 < c_1 < c_2 \leq \infty$. Note that $\hat{\lambda}_{SR}$ converges to λ almost surely as $N \to \infty$.

Example 7.1

Considering the standard SαS distribution (zero–mean and unit dispersion), let us demonstrate how the value of α influences a SαS process. What we can expect is that the smaller the value of α, the more will be

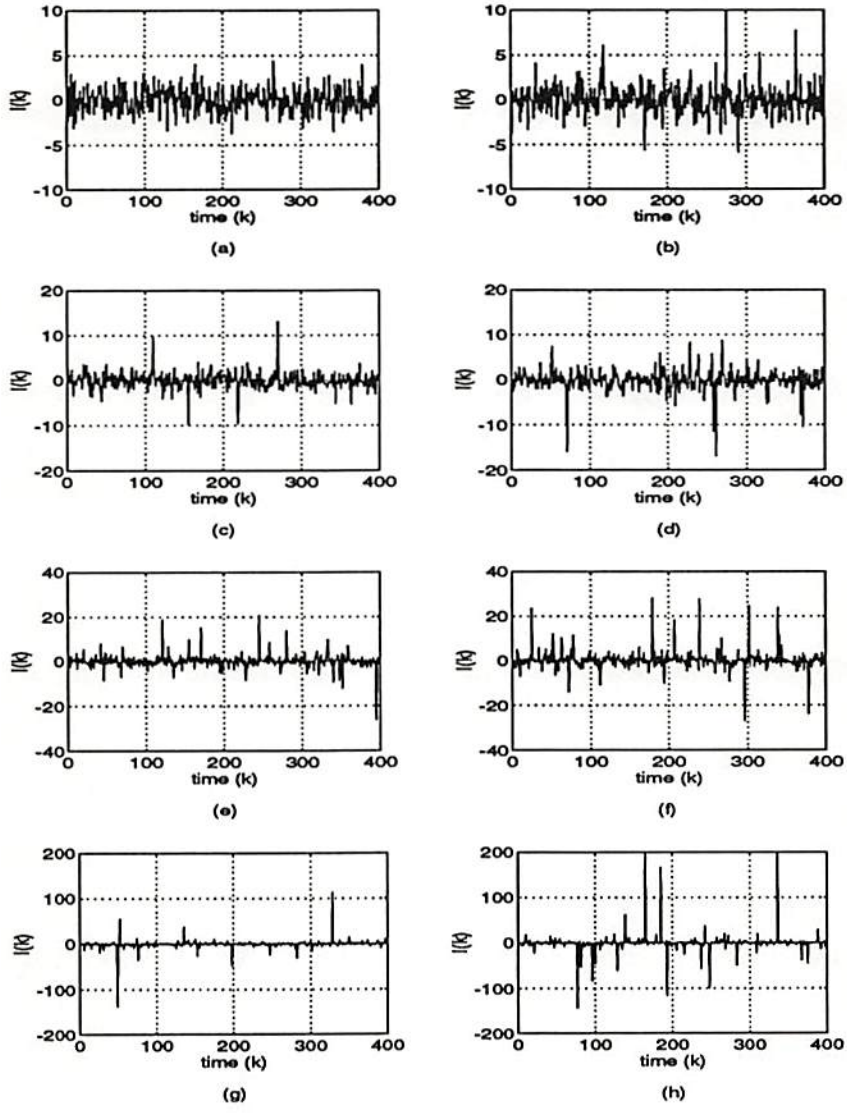

Figure 7.1: Symmetric α–stable processes with different values of α: (a) 2, (b) 1.9, (c) 1.75, (d) 1.6, (e) 1.35, (f) 1.2, (g) 1, and (h) 0.9.

the outliers due to heavier tail of the distribution. Fig. 7.1 illustrates typical SαS processes with different values of α: $\alpha = 2$ (Gaussian), 1.9, 1.75, 1.6, 1.35, 1.2, 1 (Cauchy), and 0.9. Note that each figure presents one realization and that each figure has different scales for easy looking plots.

From the Fig. 7.1, it is easy to observe that processes with $\alpha < 2$ (Fig. 7.1 (b) – (h)) have larger and more outliers than a Gaussian process with $\alpha = 2$ (Fig. 7.1 (a)). As the value of α decreases, SαS process becomes more impulsive.

7.3 AIM Methods for SαS Interferences

In this section, we describe AIM methods for mitigating interferences whose distributions are characterized by SαS distributions. The model of additive interference mitigation with a reference signal is the same as shown in Fig. 3.1.

7.3.1 AIM Method Based on LMS Algorithm

The LMS filter update using an instant and simple gradient estimate is expressed by [Widrow and Stearns, 1985]

$$\mathbf{H}_{lms}(k+1) = \mathbf{H}_{lms}(k) + 2\mu\, e_{lms}(k)\, \mathbf{z}(k). \tag{7.18}$$

where μ is a constant that regulates the speed and stability of adaptation. The process $\{e_{lms}(k)\}$ and the vector $\mathbf{z}(k)$ denote the AIM system output and the filter input vector consisting of reference signals $\{z_m(k),\ m = 0, 1, \ldots, M-1\}$, respectively. The AIM method based on the LMS algorithm is described in Chapter 3.

In theory, the LMS algorithm is not suitable for mitigating SαS interferences because its corresponding criterion is the minimum mean square

error (MMSE). Under this criterion, the best estimate is the one that minimizes the variance of the estimation error. For stable processes with $\alpha < 2$, however, finite variances do not exist. Although its statistical analysis is not possible, one can still use the AIM method based on LMS algorithm because of the fact that the LMS algorithm uses an instant estimate of the gradient which is finite.

Example 7.2

Assuming that the SOI is a BPSK signal satisfying $\Pr\{s(k) = 1\} = \Pr\{s(k) = -1\} = 0.5$ and that the reference signal $\{w(k)\}$ has a SαS distribution with $1 \leq \alpha < 2$. Since the variance of the reference signal does not exist, we define the signal power to interference dispersion ratio (SPIDR) as follows:

$$\text{SPIDR (dB)} = 10 \log_{10} \frac{\sigma_s^2}{\gamma_w} \tag{7.19}$$

where σ_s^2 is the power (variance) of the SOI and γ_w is the dispersion of the reference signal. We assume that the interference $\{I(k)\}$ is generated by a MA system excited by the reference signal. The MA system is given by [1, 0.75, −0.2, 0.1].

When the SPIDR is 0 dB, Fig. 7.2 illustrates the averaged interferences over 50 independent trials with $\alpha = 2$, 1.75, 1.5, and 1.2. Due to its impulsiveness, the averaged interferences contain more and larger outliers as the value of α decreases. Applying the AIM method based on the LMS algorithm for mitigating the interferences shown in Fig. 7.2, we obtain the averaged error curves between the system outputs and the SOI over 50 independent runs and illustrate them in Fig. 7.3. The number of filter taps is four and the step size constant μ is 0.0075 for the filter update equations. The results obtained by the AIM method based

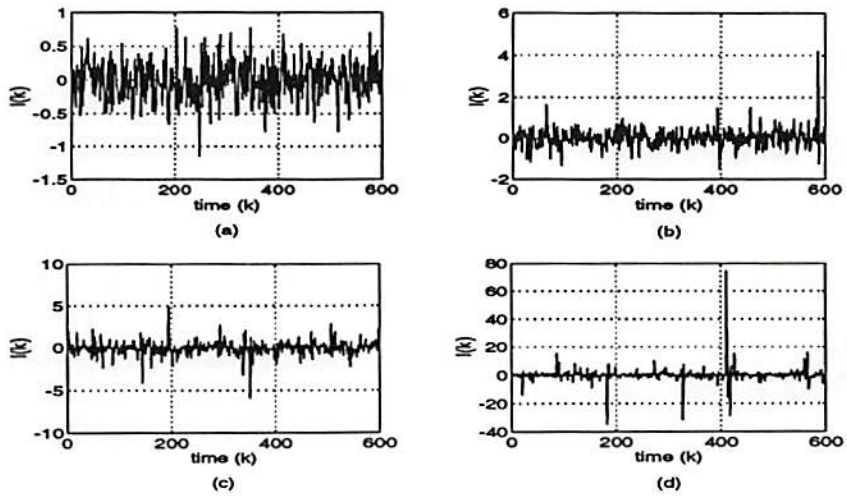

Figure 7.2: Averaged interference $\{I(k)\}$ when its SαS distribution is of (a) $\alpha = 2$, (b) $\alpha = 1.75$, (c) $\alpha = 1.5$, and (d) $\alpha = 1.2$.

on the LMS algorithm demonstrate its poor performance except for the case of $\alpha = 2$ (Gaussian interferences). When the value of α is less than two, we can observe much larger outliers exist than the interferences. This effect becomes more severe as the value of α decreases. The poor performance of the AIM method based on the LMS algorithm for interferences with $\alpha < 2$ is due to the filter input vector $\mathbf{z}(k)$ which contains impulsive signals. Since large signal values of the filter input vector directly affect the adaptive filter coefficients, large variation of the filter coefficients results in poor performance. Thus, we can conclude that it is not appropriate to use the LMS–based AIM method for mitigating the impulsive interferences ($\alpha < 2$) characterized by SαS distributions.

Previously, assuming that we had *a priori* knowledge about the relationship between the interference and the reference signal, the number of filter taps was chosen as four. In practice, we do not have such information. Thus, the number of taps has to be estimated and is usually

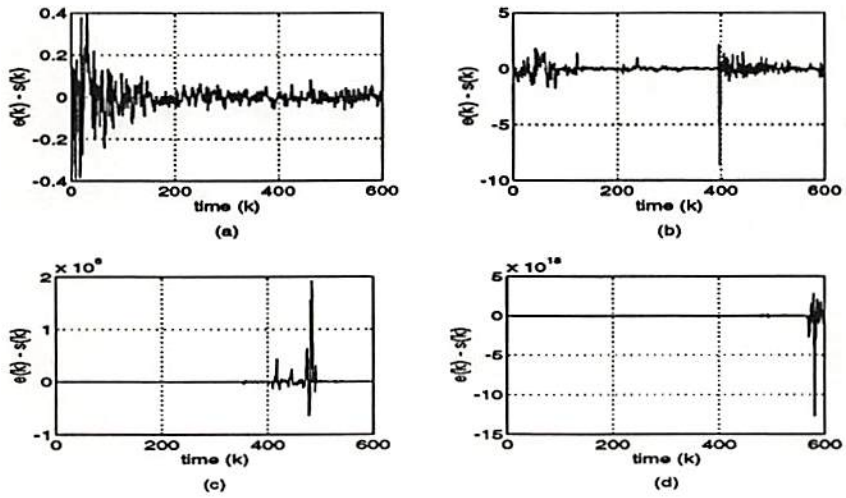

Figure 7.3: Averaged error curves $\{e_{lms}(k) - s(k)\}$ between the SOI and the outputs obtained by the AIM method based on the LMS algorithm ($M = 4$) when the interference has SαS distribution with (a) $\alpha = 2$, (b) $\alpha = 1.75$, (c) $\alpha = 1.5$, and (d) $\alpha = 1.2$.

chosen greater than the true number of weights for interference. Now, let us consider the AIM method based on the LMS algorithm whose number of filter tabs is $M = 8$ or 12. When the SPIDR is 0 dB and $M = 8$, Fig. 7.4 shows the averaged error curves $\{e_{lms}(k) - s(k)\}$ obtained from the AIM method based on LMS algorithm for mitigating the impulsive interferences with $\alpha = 1.75$ or 1.5. Its step size parameter is $\mu = 0.007$. Also, its performance with $M = 12$ and $\mu = 0.006$ is given in Fig. 7.4. Comparing the results shown in Fig. 7.3 with those in Fig. 7.4, we can conclude that an overestimated number of filter tabs can degrade the performance of the LMS algorithm more severely. As the overestimated tab number increases, its performance becomes worse. Also note that a smaller value the characteristic exponent of the interference results in its worse performance. Since overestimated numbers of filter tabs deteriorate the performance of the LMS algorithm more severely, the utilization

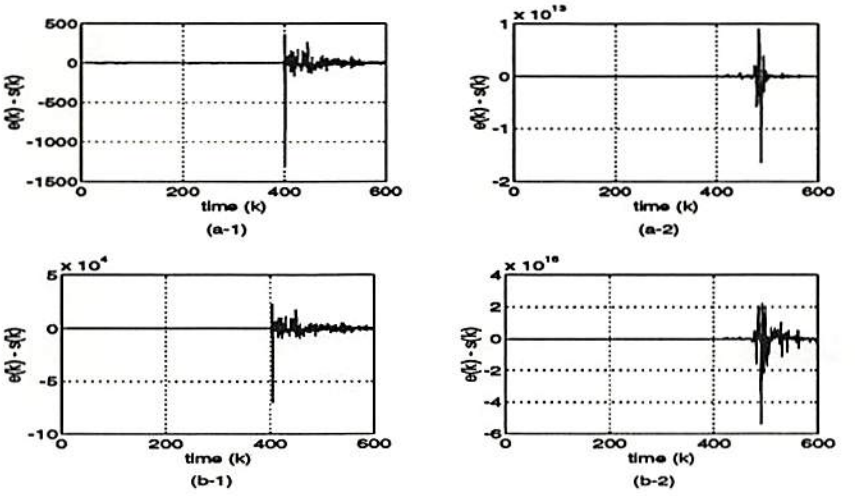

Figure 7.4: Averaged error curves $\{e_{lms}(k) - s(k)\}$ between the SOI and the outputs obtained by the AIM method based on the LMS algorithm when (a–1) $\alpha = 1.75$ and $M = 8$, (a–2) $\alpha = 1.5$ and $M = 8$, (b–1) $\alpha = 1.75$ and $M = 12$, and (b–2) $\alpha = 1.5$ and $M = 12$.

of the LMS algorithm should be avoided when the interference is assumed to be impulsive.

7.3.2 AIM Method Based on NLMS Algorithm

To reduce the variation effect of the filter input vector on the filter coefficient update equation, we can consider the AIM method using the normalized LMS (NLMS) algorithm. The NLMS–based AIM method is described in Chapter 5 and its update equation is given by

$$\mathbf{H}_{nlms}(k+1) = \mathbf{H}_{nlms}(k) + \mu \frac{e_{nlms}(k)\,\mathbf{z}(k)}{\|\mathbf{z}(k)\|^2 + \beta} \tag{7.20}$$

where $\beta, \mu > 0$ are appropriately chosen update parameters. The parameter β is used to prevent the denominator of the update equation from being zero (or very close to zero). Note that the usual statistical analysis

for the performance of the NLMS algorithm is not possible due to the lack of finite variance of the interference.

Example 7.3

Considering the same signals as those in Example 7.2, we apply the AIM method based on NLMS algorithm. The averaged error curves between the SOI and the system outputs obtained by the NLMS–based AIM method are illustrated in Fig. 7.5 when the interference distribution is of $\alpha = 2$, 1.75, 1.5, and 1.2. Note that $M = 4$, $\mu = 0.075$ and $\beta = 3.5$.

Surprisingly, the AIM method based on the NLMS algorithm shows consistently good performance even when the value of α is less than two. Since it becomes optimal (in MMSE sense) when $\alpha = 2$ and mitigates impulsive interferences very effectively, the AIM method based on the NLMS algorithm looks promising for SαS distribution processes even though its usual statistical performance analysis is not possible.

Now let us investigate the performance of the NLMS–based AIM algorithm for mitigating SαS interferences when the number of filter tabs is overestimated. When the characteristic exponent is 1.75 or 1.5, Fig. 7.6 illustrates the averaged error curves $\{e_{nlms}(k) - s(k)\}$ between the SOI and the system outputs obtained by the AIM methods based on the NLMS algorithm. The overestimated number of filter tabs is $M = 8$ or $M = 12$. For the NLMS algorithm with $M = 8$ the parameters are chosen as $\mu = 0.105$ and $\beta = 3.5$. For $M = 12$, the parameters are chosen as $\mu = 0.145$ and $\beta = 3.5$. Comparison between the results in Fig. 7.5 and Fig. 7.6 indicates that as the overestimated number of filter tabs increases, its performance only slightly deteriorates in the sense that the averaged errors increase just a little bit. We can conclude that the utilization of the NLMS algorithm is practical and appropriate to cancel

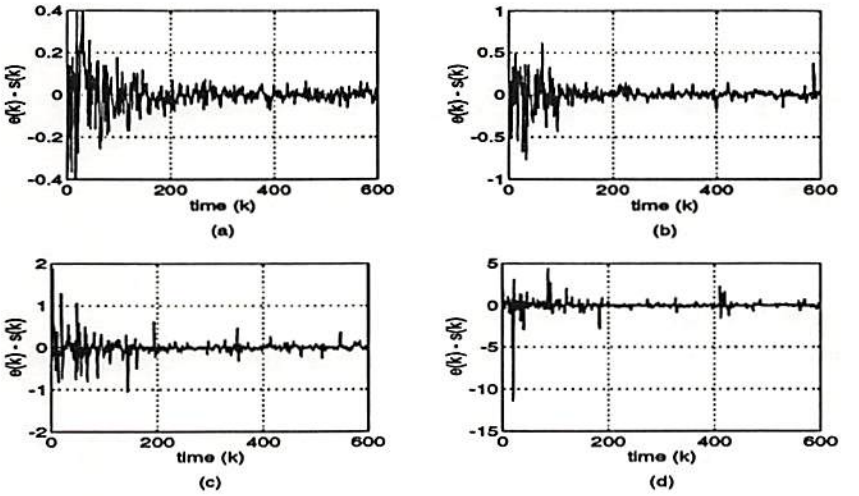

Figure 7.5: Averaged error curves $\{e_{nlms}(k) - s(k)\}$ between the SOI and the outputs obtained by the AIM method based on the normalized LMS algorithm ($M = 4$) when the interference has SαS distribution with (a) $\alpha = 2$, (b) $\alpha = 1.75$, (c) $\alpha = 1.5$, and (d) $\alpha = 1.2$.

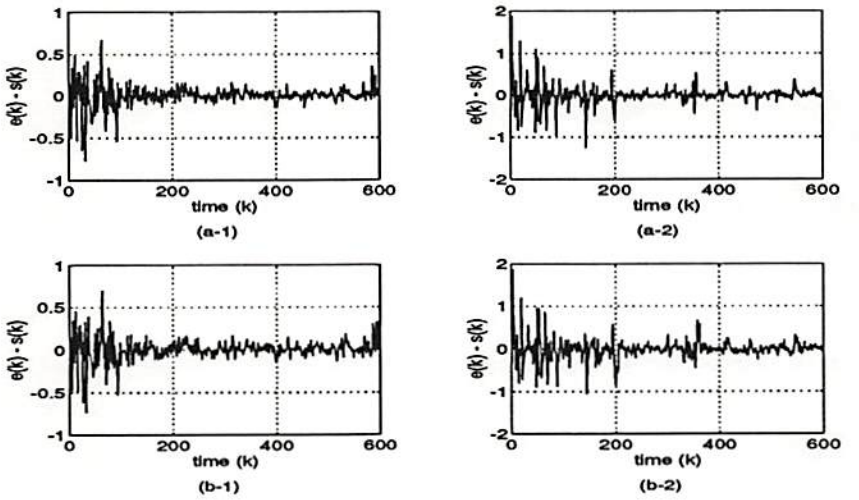

Figure 7.6: Averaged error curves $\{e_{nlms}(k) - s(k)\}$ between the SOI and the outputs obtained by the AIM method based on the normalized LMS algorithm when (a–1) $\alpha = 1.75$ and $M = 8$, (a–2) $\alpha = 1.5$ and $M = 8$, (b–1) $\alpha = 1.75$ and $M = 12$, and (b–2) $\alpha = 1.5$ and $M = 12$.

impulsive interferences based on the facts that it can effectively mitigate interferences which are highly impulsive and that it still performs well with overestimated numbers of filter tabs.

7.3.3 AIM Method Based on LMP Algorithm

Suppose that the interference has a SαS distribution with $1 \leq \alpha < 2$ and that the value of α is known in advance. Since the MMSE criterion is not available, the minimum dispersion criterion can be considered as a natural and mathematically meaningful choice for a measure of optimality in stable signal processing. What we would like to do is to adjust the filter coefficients so that the dispersion of the system output $\{e(k)\}$ is minimized. Thus, the cost function is given by

$$J = \| e(k) \|_\alpha = \| x(k) - \sum_{m=0}^{M-1} h_m(k)\, z_m(k) \|_\alpha \tag{7.21}$$

where $\| \cdot \|_\alpha$ is defined in (7.10). This cost function is quite intractable in general. So, we can use an equivalent cost function given by

$$J = \mathbf{E}\{ | x(k) - \sum_{m=0}^{M-1} h_m(k)\, z_m(k) |^p \} \tag{7.22}$$

where $1 \leq p < \alpha$. Using the given equivalent cost function, we can obtain a stochastic gradient method to adjust the filter coefficients in a similar way as the LMS algorithm does. The algorithm is called the least mean p–norm (LMP) [Shao and Nikias, 1993]. When we fix p, the tap weight update equation is of the following form

$$\mathbf{H}_{lmp}(k+1) = \mathbf{H}_{lmp}(k) + \mu\, \mathbf{z}(k)\ |e_{lmp}(k)|^{p-1}\ \text{sign}(\, e_{lmp}(k)\,) \tag{7.23}$$

where $\mathbf{H}_{lmp}(k)$ is the adaptive filter coefficient vector and $\mu > 0$ is the step size. The adaptive filter output is obtained by

$$y_{lmp}(k) = \mathbf{H}_{lmp}^T(k)\, \mathbf{z}(k) \tag{7.24}$$

and the system output is given by

$$e_{lmp}(k) = x(k) - y_{lmp}(k). \tag{7.25}$$

The disadvantages of the LMP–based AIM method are:

- The value of α for the interference (reference signal) must be known or estimated in advance.
- The computational burden may be heavy to calculate $|e_{lmp}(k)|^{p-1}$ with $1 \leq p < \alpha$.

Example 7.4

Considering the same signals as those in Example 7.2, we apply the AIM method based on LMP algorithm. The averaged error curves between the SOI and the system outputs obtained by the LMP–based AIM method ($M = 4$ and $\mu = 0.0075$) are illustrated in Fig. 7.7 when the characteristic exponent of the interference $\alpha = 2$, 1.75, 1.5, and 1.2. Note that the update equation (7.22) is calculated using $p = \alpha - 0.05$.

When the value of α decreases, the performance of the AIM method based on the LMP algorithm deteriorates. When $\alpha = 1.5$ and 1.2, the error curves contain some enlarged outliers rather than reducing impulsive interferences. Although its performance is better than that of the AIM method based on the LMS algorithm, the AIM method based on the LMP algorithm cannot be used for mitigating impulsive interferences with a small value of α.

Fig. 7.8 demonstrates the performance of the LMP algorithm when the number of filter tabs are overestimated as $M = 8$ and $M = 12$. The corresponding step size parameters are $\mu = 0.007$ and 0.006, respectively.

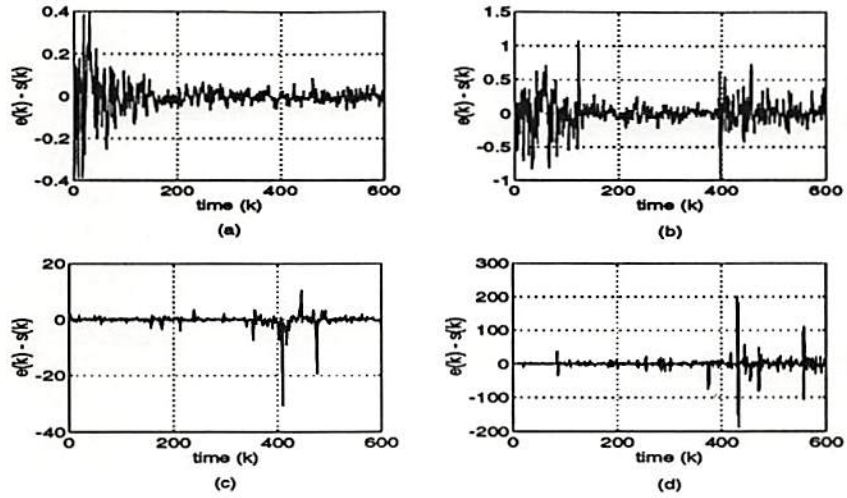

Figure 7.7: Averaged error curves $\{e_{lmp}(k) - s(k)\}$ between the SOI and the outputs obtained by the AIM method based on the LMP algorithm ($M = 4$) when the interference has SαS distribution with (a) $\alpha = 2$, (b) $\alpha = 1.75$, (c) $\alpha = 1.5$, and (d) $\alpha = 1.2$.

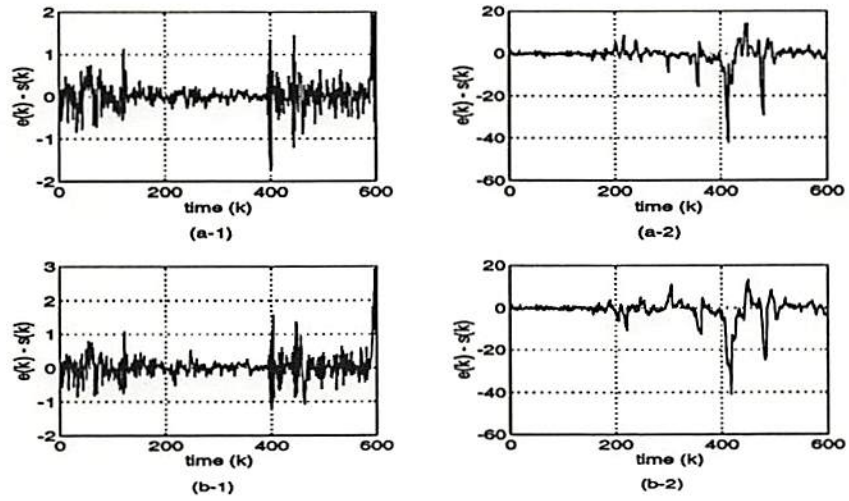

Figure 7.8: Averaged error curves $\{e_{lmp}(k) - s(k)\}$ between the SOI and the outputs obtained by the AIM method based on the LMP algorithm when (a–1) $\alpha = 1.75$ and $M = 8$, (a–2) $\alpha = 1.5$ and $M = 8$, (b–1) $\alpha = 1.75$ and $M = 12$, and (b–2) $\alpha = 1.5$ and $M = 12$.

As the overestimated number of filter tabs increases, the performance of the LMP algorithm slightly deteriorates in the sense that the averaged errors increase and that the error curves are smoothed (slow convergence due to impulsive signals).

7.3.4 AIM Method Based on NLMP Algorithm

Recently, a variation of the LMP algorithm has been suggested with the motivation of the NLMS [Arikan *et al.*, 1994], which is called the normalized LMP (NLMP) algorithm using the following update equation

$$\mathbf{H}_{nlmp}(k+1) = \mathbf{H}_{nlmp}(k) + \mu \frac{|e_{nlmp}(k)|^{p-1} \ \text{sign}(e_{nlmp}(k))}{\|\mathbf{z}(k)\|_p^p + \beta} \mathbf{z}(k) \quad (7.26a)$$

where

$$\|\mathbf{z}(k)\|_p = \left(\sum_{m=0}^{M} |z_m(k)|^p \right)^{1/p} \quad (7.26b)$$

and β, $\mu > 0$ are appropriately chosen update parameters. The parameter β is used to avoid excessively large updates in case of an occasionally small filter tap inputs. Note that p should be less than the value of α. When $p = 2$, the NLMP algorithm becomes the NLMS algorithm. Again, we must know or estimate, in advance, the value of α for the interference.

The disadvantages of the NLMP–based AIM method share those of the LMP–based AIM method. In addition, it requires additional computational burden to compute $\|\mathbf{z}(k)\|_p^p$.

Example 7.5

Considering the same signal environments as in the previous examples, we apply the AIM method based on NLMP algorithm. The averaged error curves $\{e_{nlmp}(k) - s(k)\}$ between the SOI and the system outputs obtained by the NLMP–based AIM method ($M = 4$, $\mu = 0.075$, and

$\beta = 3.5$) are shown in Fig. 7.9 when the interference distribution is of $\alpha = 2$, 1.75, 1.5, and 1.2. Note that $p = \alpha - 0.05$ has been used for the update equation.

Note that the performance of the AIM method based on the NLMP algorithm is very similar to that of the AIM method based on the NLMS algorithm. When $\alpha = 1.5$ and 1.2, the latter AIM method performs better than the former one in the sense of smaller errors after convergence. Moreover, in spite of its similar performance, the AIM method based on the NLMP algorithm is less attractive than the AIM method based on the NLMS algorithm. This is due to facts that the value of α must be known or estimated in advance and that the computational complex is much heavier because of non–integer values of $p < \alpha$. The computational complexity will be considered later.

Now, consider overestimated numbers of filter tabs: $M = 8$ and $M =$

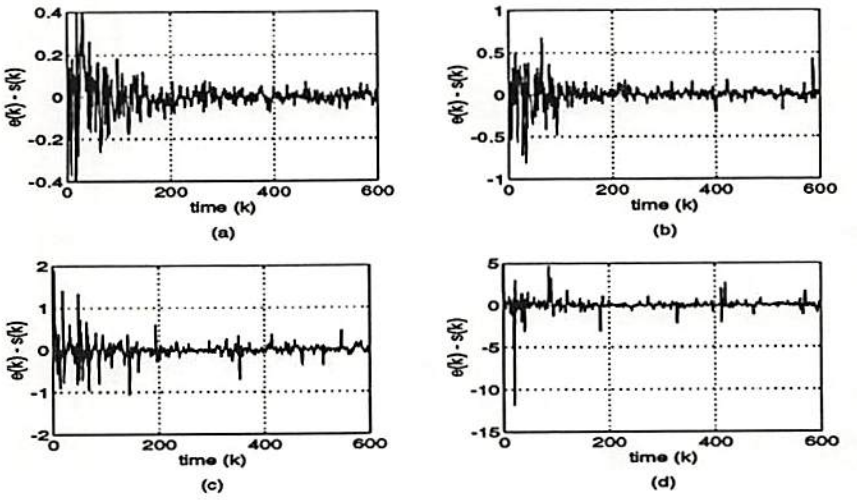

Figure 7.9: Averaged error curves $\{e_{nlmp}(k) - s(k)\}$ between the SOI and the outputs obtained by the AIM method based on the normalized LMP algorithm ($M = 4$) when the interference has SαS distribution with (a) $\alpha = 2$, (b) $\alpha = 1.75$, (c) $\alpha = 1.5$, and (d) $\alpha = 1.2$.

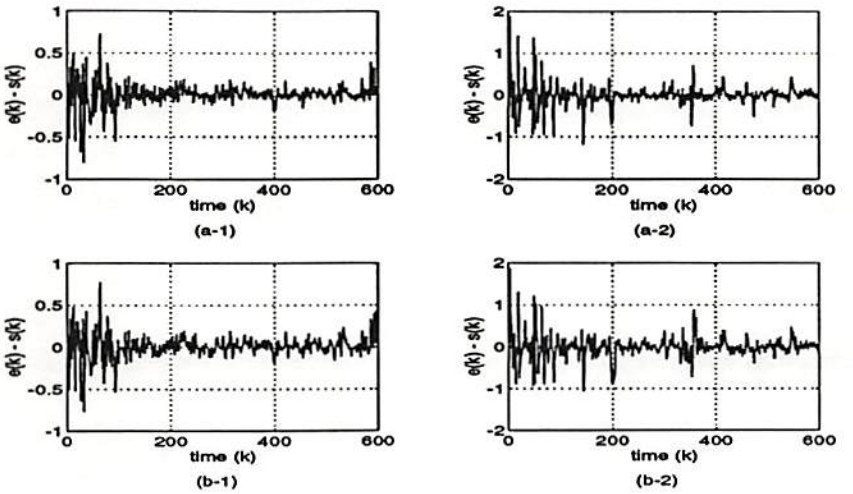

Figure 7.10: Averaged error curves $\{e_{nlmp}(k) - s(k)\}$ between the SOI and the outputs obtained by the AIM method based on the normalized LMP algorithm when (a–1) $\alpha = 1.75$ and $M = 8$, (a–2) $\alpha = 1.5$ and $M = 8$, (b–1) $\alpha = 1.75$ and $M = 12$, and (b–2) $\alpha = 1.5$ and $M = 12$.

12. Fig. 7.10 illustrates the performance of the NLMP algorithm with $M = 8$, $\mu = 0.105$, and $\beta = 3.5$ when the characteristic exponent of the impulsive interferences is 1.75 or 1.5. Its performance with $M = 12$, $\mu = 0.145$, and $\beta = 3.5$ is also given in Fig. 7.10. Again, we can observe that its performance with overestimated filter tabs deteriorates as much as that of the NLMS–based AIM method shown in Fig. 7.6. Note that the performance of the latter algorithm is still better than that of the former one.

7.3.5 AIM Method Based on LMAD Algorithm

When $p = 1$, the LMP algorithm will be called the least mean absolute deviation (LMAD) algorithm [Shao and Nikias, 1993]. Note that the LMAD algorithm is actually the familiar signed LMS algorithm (pilot LMS, signed error, or sign algorithm) [Clarkson, 1993]. The adaptive

filter update equation based on the LMAD algorithm is given by

$$\mathbf{H}_{lmad}(k+1) = \mathbf{H}_{lmad}(k) + \mu\, \mathbf{z}(k)\, \text{sign}(\, e_{lmad}(k)\,) \tag{7.27}$$

where $\mathbf{H}_{lmad}(k)$ is the adaptive filter coefficient vector and $\mu > 0$ is the step size. The LMAD algorithm is simpler than the LMS algorithm because of the "sign" operator on the system output. Note that a sign algorithm usually suffers from slow convergence rate. Another advantage of the LMAD–based AIM method is that it is not necessary to estimate or know the characteristic exponent of the interferences. The LMAD algorithm is useful when the true value of the characteristic exponent α is sure to be greater than unity, but either exactly unknown or varying in time.

Example 7.6

Let us apply the AIM method based on LMAD algorithm to the problem in the previous examples. The averaged error curves between the SOI and the system outputs obtained by the AIM method using LMAD algorithm are shown in Fig. 7.11 when $\alpha = 2$, 1.75, 1.5, and 1.2. The number of filter tap is fixed at four and the step size parameter is $\mu = 0.0075$.

Although its implementation is simpler and its performance is better than the AIM method based on the LMS algorithm, the AIM method based on the LMAD algorithm is not promising for SαS interference mitigation because the error curves still contains large values. Compared with the performance of the aforementioned normalized algorithms, its performance is not good enough. That means the LMAD–based AIM method cannot effectively mitigate SαS impulsive interferences. This phenomenon becomes more severe as the value of α decreases.

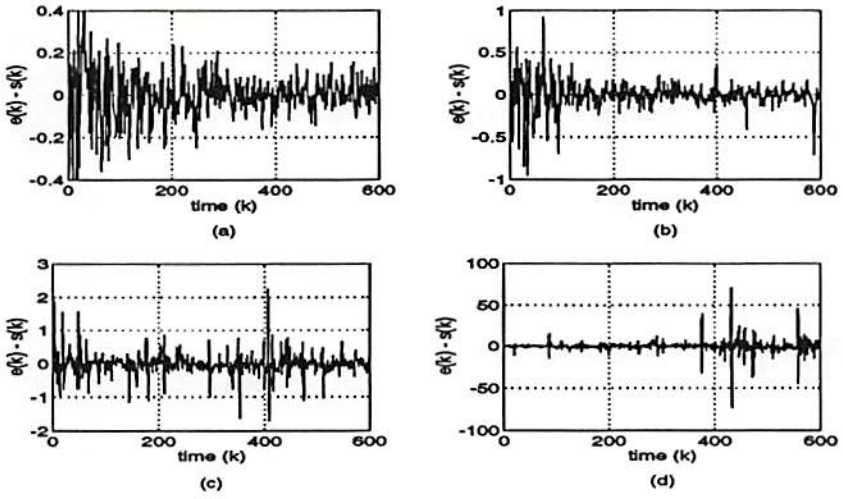

Figure 7.11: Averaged error curves $\{e_{lmad}(k)-s(k)\}$ between the SOI and the outputs obtained by the AIM method based on the LMAD algorithm ($M = 4$) when the interference has SαS distribution with (a) $\alpha = 2$, (b) $\alpha = 1.75$, (c) $\alpha = 1.5$, and (d) $\alpha = 1.2$.

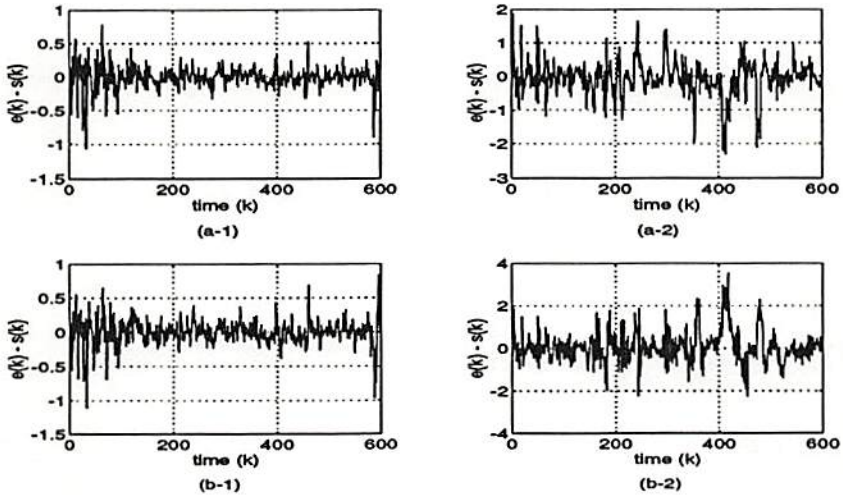

Figure 7.12: Averaged error curves $\{e_{lmad}(k)-s(k)\}$ between the SOI and the outputs obtained by the AIM method based on the LMAD algorithm when (a–1) $\alpha = 1.75$ and $M = 8$, (a–2) $\alpha = 1.5$ and $M = 8$, (b–1) $\alpha = 1.75$ and $M = 12$, and (b–2) $\alpha = 1.5$ and $M = 12$.

Let us consider the performance of the LMAD algorithm with overestimated number of filter tabs as before. Again, two different numbers of filter tabs are considered: $M = 8$ and $M = 12$. The corresponding step size parameters are $\mu = 0.00875$ and $\mu = 0.0095$, respectively. The performance of the LMAD–based AIM method with the filter tabs is shown in Fig. 7.12 when the characteristic exponent of the impulsive interferences are 1.75 and 1.5. The results given in Fig. 7.11 and Fig. 7.12 indicate that the performance of the LMAD algorithm is not greatly affected by the number of filter tabs and that an overestimated filter tabs causes slightly enlarged and smoothed error curves (slow cancellation of impulsive signals).

7.3.6 AIM Method Based on NLMAD Algorithm

When $p = 1$, the NLMP algorithm becomes the normalized LMAD (NLMAD) algorithm [Arikan *et al.*, 1994]. The adaptive filter update equation based on the NLMAD algorithm is

$$\mathbf{H}_{nlmad}(k+1) = \mathbf{H}_{nlmad}(k) + \mu \frac{\text{sign}(\, e_{nlmad}(k)\,)}{\|\mathbf{z}(k)\|_1 + \beta}\, \mathbf{z}(k) \qquad (7.28)$$

where $\mathbf{H}_{nlmad}(k)$ is the adaptive filter coefficient vector and $\mu > 0$ is the step size. Again, the NLMAD algorithm can be utilized when the true value of the characteristic exponent α is either exactly unknown or varying in time (assuming $\alpha > 1$). The advantages of the NLMAD algorithm are:

- Its computational complexity is lower than that of the previous other normalized algorithms because of the sign operator and the use of L^1 norm.
- The estimation or *a priori* knowledge of the characteristic exponent of the interferences is not necessary.

Example 7.7

Applying the AIM method based on NLMAD algorithm to the same problem as that in Example 7.2, the averaged error curves $\{e_{nlmad}(k) - s(k)\}$ between the SOI and the system outputs obtained by the NLMAD–based AIM method are shown in Fig. 7.13, when $\alpha = 2$, 1.75, 1.5, and 1.2. The number of filter tabs is four. The parameters μ and β are given as 0.075 and 1.5, respectively.

Fig. 7.13 illustrates that the normalization procedure inherent in the NLMAD algorithm clearly improves its performance. Compared with the results obtained from the AIM methods based on the LMAD algorithm, this improved performance can be observed when the value of α is 1.5 and 1.2 (see Fig. 7.13 (c) and (d)). We can notice, however, that its performance does not improve much when $\alpha = 2$ and 1.75. Although

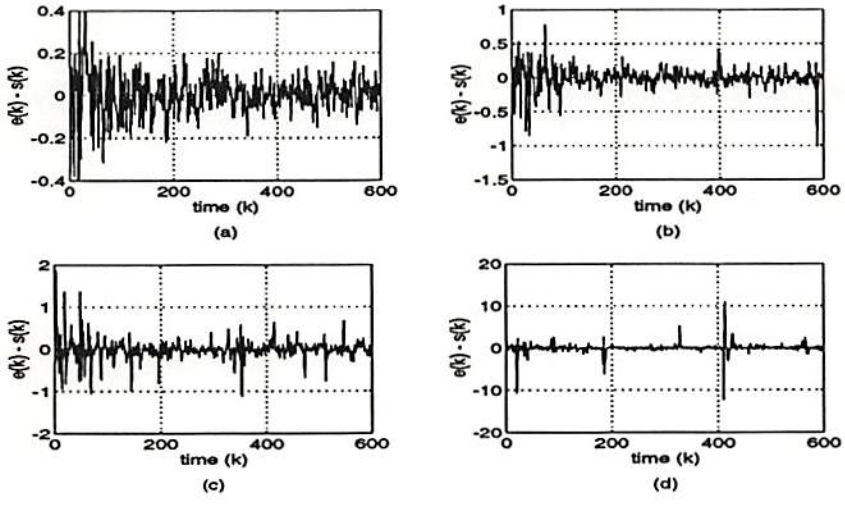

Figure 7.13: Averaged error curves $\{e_{nlmad}(k) - s(k)\}$ between the SOI and the outputs obtained by the AIM method based on the normalized LMAD algorithm ($M = 4$) when the interference has SαS distribution with (a) $\alpha = 2$, (b) $\alpha = 1.75$, (c) $\alpha = 1.5$, and (d) $\alpha = 1.2$.

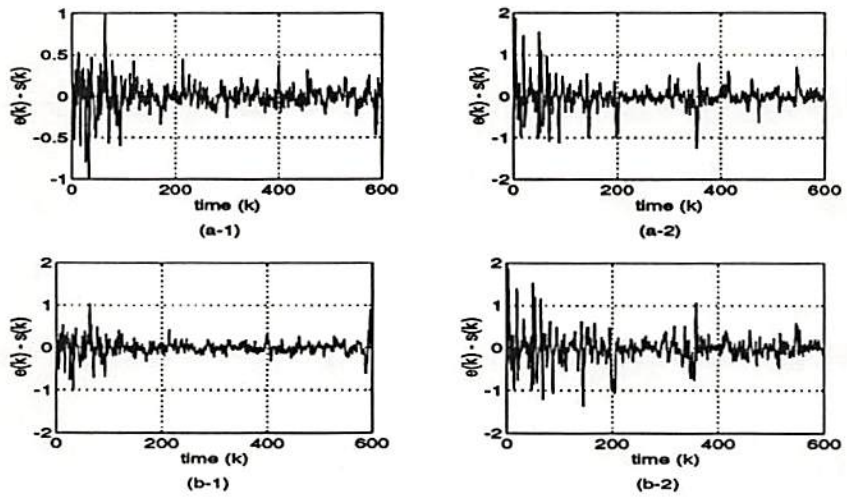

Figure 7.14: Averaged error curves $\{e_{nlmad}(k) - s(k)\}$ between the SOI and the outputs obtained by the AIM method based on the normalized LMAD algorithm when (a–1) $\alpha = 1.75$ and $M = 8$, (a–2) $\alpha = 1.5$ and $M = 8$, (b–1) $\alpha = 1.75$ and $M = 12$, and (b–2) $\alpha = 1.5$ and $M = 12$.

the NLMAD algorithm is computationally simpler and faster than the NLMS algorithm, its performance for impulsive interferences with all the values of α is not as good as that of the latter algorithm. When the value of α is 2 or 1.75, its errors are larger. When $\alpha = 1.5$ or 1.2, it cannot effectively mitigate impulsive components of the interferences.

When the number of tabs is overestimated, Fig. 7.14 shows the performance of the NLMAD–based AIM method when $\alpha = 1.75$ or 1.5. When the number of tabs is $M = 8$, the parameters for the update equation are $\mu = 0.085$ and $\beta = 1.5$. When $M = 12$, they are fixed at 0.105 and 1.5, respectively. From Fig. 7.14, we can observe that the overestimated number of filter tabs causes slight performance deterioration. Based on the results obtained all the normalized algorithms, we can conclude that an overestimated number of filter tabs does not greatly degrade their performance.

7.3.7 AIM Method Based on Clipped LMS Algorithm

A modification of the LMAD algorithm is so called clipped LMS (CLMS) algorithm (or signed regressor) [Clarkson, 1993] described by

$$\mathbf{H}_{clms}(k+1) = \mathbf{H}_{clms}(k) + \mu \,\mathrm{sign}(\,\mathbf{z}(k)\,)\, e_{clms}(k) \tag{7.29}$$

where $\mathbf{H}_{clms}(k)$ is the adaptive filter coefficient vector and $\mu > 0$ is the step size. The only difference between the CLMS algorithm and the LMAD algorithm is that the "sign" operator is applied on the tap input vector instead on the system output. The system output is given by

$$e_{clms}(k) = x(k) - \mathbf{H}^T_{clms}(k)\,\mathbf{z}(k). \tag{7.30}$$

Example 7.8

Considering the same signal environment as in the previous examples, we apply the AIM method based on CLMS algorithm. The averaged error curves $\{e_{clms}(k) - s(k)\}$ between the SOI and the system outputs obtained by the CLMS–based AIM method ($M = 4$ and $\mu = 0.0075$) are shown in Fig. 7.15 when $\alpha = 2$, 1.75, 1.5, and 1.2.

When $\alpha = 2$, the AIM method based on the CLMS algorithm converges faster and has smaller errors than the LMAD algorithm. On the other hand, when $\alpha = 1.5$ and 1.2, the performance of the CLMS algorithm is very poor. Although it is simpler than the LMAD algorithm, the AIM method based on the CLMS algorithm cannot be utilized to cancel SαS interferences.

Fig. 7.16 shows the performance of the CLMS–based AIM methods with the overestimated numbers of filter tabs: $M = 8$ or 12. The corresponding step size parameters for the update equation are respectively

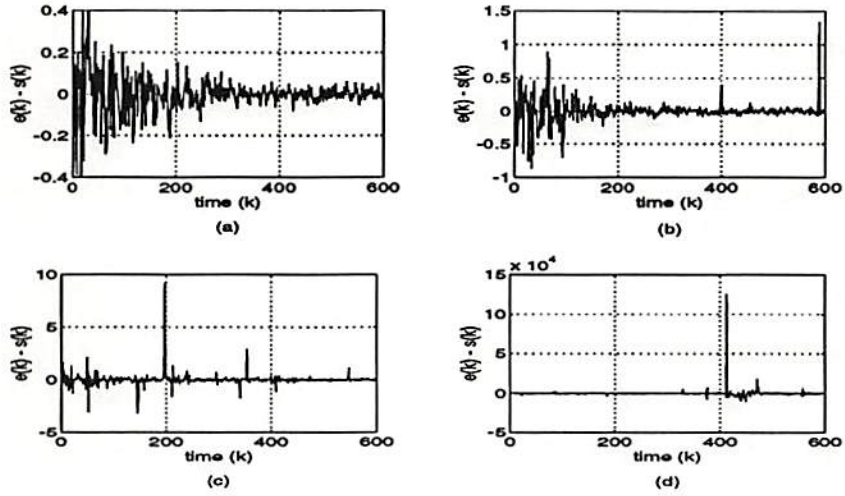

Figure 7.15: Averaged error curves $\{e_{clms}(k)-s(k)\}$ between the SOI and the outputs obtained by the AIM method based on the CLMS algorithm ($M = 4$) when the interference has SαS distribution with (a) $\alpha = 2$, (b) $\alpha = 1.75$, (c) $\alpha = 1.5$, and (d) $\alpha = 1.2$.

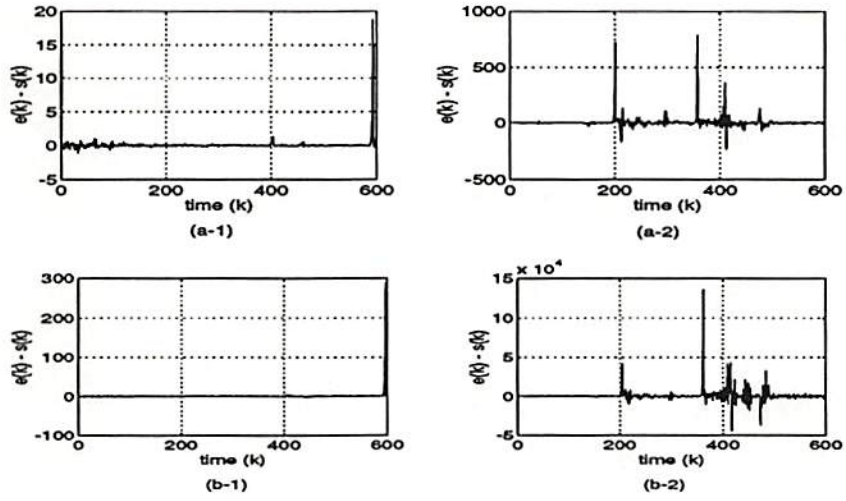

Figure 7.16: Averaged error curves $\{e_{clms}(k)-s(k)\}$ between the SOI and the outputs obtained by the AIM method based on the CLMS algorithm when (a–1) $\alpha = 1.75$ and $M = 8$, (a–2) $\alpha = 1.5$ and $M = 8$, (b–1) $\alpha = 1.75$ and $M = 12$, and (b–2) $\alpha = 1.5$ and $M = 12$.

fixed at 0.00775 and 0.0075. As we can observe from Fig. 7.16, an overestimated number of tabs may cause very severe performance degradation. This phenomenon becomes more severe as the number of filter tabs increases.

7.3.8 AIM Method Based on ZFLMS Algorithm

The simplest LMS–type algorithm is the zero–forcing LMS (ZFLMS) algorithm (or sign–sign algorithm) given by [Clarkson, 1993]

$$\mathbf{H}_{zflms}(k+1) = \mathbf{H}_{zflms}(k) + \mu \,\text{sign}(\,\mathbf{z}(k)\,)\,\text{sign}(\,e_{zflms}(k)\,) \qquad (7.31)$$

where $\mathbf{H}_{zflms}(k)$ is the adaptive filter coefficient vector and $\mu > 0$ is the step size. Note that the "sign" operator is applied to both the filter tap input vector and the system output. When we consider the AIM method based on the ZFLMS algorithm, the system output is given by

$$e_{zflms}(k) = x(k) - \mathbf{H}^T_{zflms}(k)\,\mathbf{z}(k). \qquad (7.32)$$

The major disadvantages of the ZFLMS algorithm is its slow convergence and large excess errors.

Example 7.9

Considering the same signal environment as in the previous examples, we apply the AIM method based on ZFLMS algorithm. The averaged error curves $\{e_{zflms}(k) - s(k)\}$ between the SOI and the system outputs obtained by the ZFLMS–based AIM method ($M = 4$ and $\mu = 0.025$) are shown in Fig. 7.17 when $\alpha = 2$, 1.75, 1.5, and 1.2.

When $\alpha = 2$ and 1.75, the AIM method based on ZFLMS algorithm suffers from slow convergence rate and large excess errors. However, for a

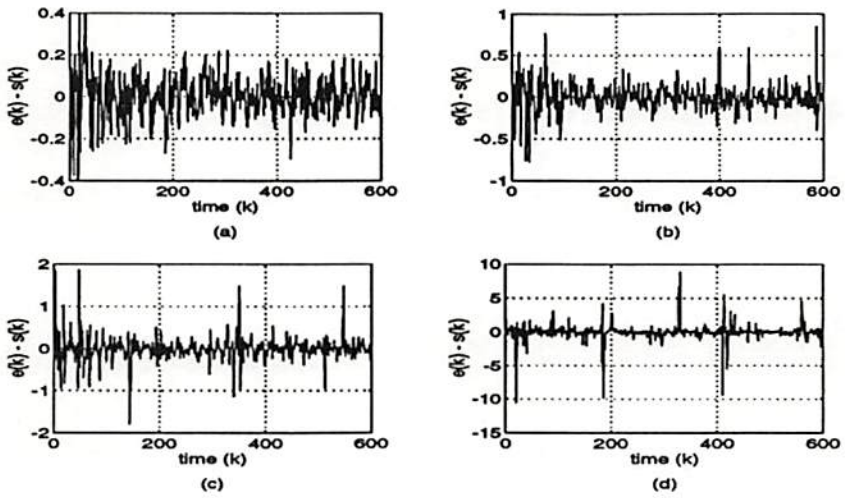

Figure 7.17: Averaged error curves $\{e_{zflms}(k) - s(k)\}$ between the SOI and the outputs obtained by the AIM method based on the ZFLMS algorithm ($M = 4$) when the interference has SαS distribution with (a) $\alpha = 2$, (b) $\alpha = 1.75$, (c) $\alpha = 1.5$, and (d) $\alpha = 1.2$.

smaller values of α, its performance is similar to that of the NLMS algorithm. When the interference becomes very impulsive, utilization of the ZFLMS algorithm should be considered due to its simple structure and better performance than the other algorithms using the sign operator.

As before, let us consider the performance of the ZFLMS algorithm with the overestimated numbers of filter tabs: $M = 8$ or 12. The corresponding step size parameters are respectively given as $\mu = 0.0235$ and 0.0225. Then, Fig. 7.18 illustrates its performance when the characteristic exponent is 1.75 or 1.5. We can observe that an overestimated number of filter tabs results in slight performance deterioration. However, it is interesting observation that its performance with overestimated filter tabs does not deteriorate as much as that of the CLMS algorithm and that overestimated filter tabs does not cause smoothed error curves as much as in the LMAD algorithm.

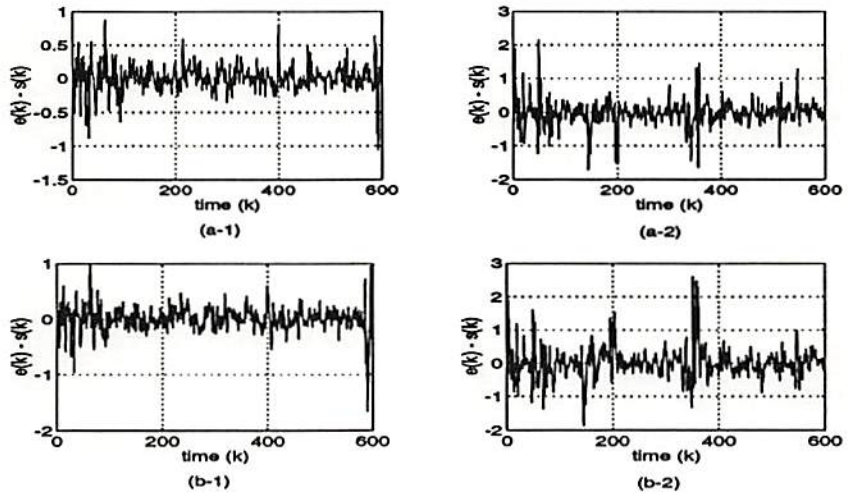

Figure 7.18: Error curves $\{e_{zflms}(k) - s(k)\}$ between the SOI and the outputs obtained by the AIM method based on the ZFLMS algorithm when (a–1) $\alpha = 1.75$ and $M = 8$, (a–2) $\alpha = 1.5$ and $M = 8$, (b–1) $\alpha = 1.75$ and $M = 12$, and (b–2) $\alpha = 1.5$ and $M = 12$.

7.3.9 Algorithmic Assessments

The algorithmic description and the computational complexity of each AIM method for mitigating a SαS interference are summarized in Table 7.1. The computational complexity of each algorithm is considered as the required number of multiplications per iteration (M/I) when the number of filter tabs is M. Note that the number of M/I of each algorithm is the sum of the number of M/I for filter update equation (algorithm description in Table 7.1) and the number of M/I for system output which is M. We also assume that there is no multiplication required for obtaining the multiplication between a signed value (sign$(\cdot)$) and a scalar (or vector), because the sign of the corresponding scalar (or vector) is only changed. Even though there may be many other methods for computing it, the value of a^b, where $a > 0$ and $|b| < \infty$, is assumed

Table 7.1 : Summary of AIM Methods for SαS Interferences

Algorithms	Description	M/I*
LMS	$\mathbf{H}(k+1) = \mathbf{H}(k) + 2\,\mu\, e(k)\, \mathbf{z}(k)$	2M + 1
NLMS	$\mathbf{H}(k+1) = \mathbf{H}(k) + \mu\, e(k)\, \mathbf{z}(k) / (\|\| \mathbf{z}(k) \|\|^2 + \beta)$	3M + 2
LMP	$\mathbf{H}(k+1) = \mathbf{H}(k) + \mu\, \mathbf{z}(k) \mid e(k) \mid^{p-1} \text{sign}(\, e(k)\,)$	2M + 8
NLMP	$\mathbf{H}(k+1) = \mathbf{H}(k) + \mu\, \mathbf{z}(k) \mid e(k) \mid^{p-1} \text{sign}(\, e(k)\,) / (\|\| \mathbf{z}(k) \|\|_p^p + \beta)$	9M + 9
LMAD	$\mathbf{H}(k+1) = \mathbf{H}(k) + \mu\, \mathbf{z}(k) \text{ sign}(\, e(k)\,)$	2M
NLMAD	$\mathbf{H}(k+1) = \mathbf{H}(k) + \mu\, \mathbf{z}(k) \text{ sign}(\, e(k)\,) / (\|\|\mathbf{z}(k) \|\| + \beta)$	2M + 1
CLMS	$\mathbf{H}(k+1) = \mathbf{H}(k) + \mu\, e(k) \text{ sign}(\, \mathbf{z}(k)\,)$	M + 1
ZFLMS	$\mathbf{H}(k+1) = \mathbf{H}(k) + \mu \text{ sign}(\, e(k)\,) \text{ sign}(\, \mathbf{z}(k)\,)$	M

*. M/I: Number of multiplications per iteration

$\mathbf{H}(k)$: Adaptive filter coefficient vector
$e(k)$: System output
$\mathbf{z}(k)$: Filter tap input vector

M: Number of filter tabs
μ : Step size
β : Normalized parameter
$p < \alpha$

LMS : Least Mean Squares
NLMS : Normalized LMS
LMP : Least Mean *p*th Norm
NLMP : Normalized LMP
LMAD : Least Mean Absolute Derivation
NLMAD : Normalized LMAD
CLMS : Clipped LMS
ZFLMS : Zero-Forcing LMS

to be calculated using the following exponential series:

$$a^b = 1 + b\log a + \frac{(b\log a)^2}{2!} + \frac{(b\log a)^3}{3!} + \cdots, \tag{7.33}$$

for the computational burden of the LMP and NLMP algorithms. To calculate (7.33) another series expansion may be needed to calculate the value of logarithm, but let us assume, for simplicity, that logarithm values are not required any multiplications by looking up a logarithm table rather than using a series expansion. Then, when the value of a^b is assumed here to be approximated using order up to three, the required number of multiplications becomes seven. The higher orders of the expansion are used, the more accurate value can be obtained, but the more number of multiplications is required. Also note that (7.33) is considered only for figuring out the required computational complexity of the LMP and the NLMP algorithms. The expansion (7.33) was not used in the examples of the LMP and the NLMP algorithms to calculate the corresponding values. Since all the results in the examples are obtained by using the MATLAB version 4.1, all the values of a^b are calculated by the algorithm provided in the MATLAB.

7.4 Summary

We have briefly reviewed the SαS distributions that can be considered as useful tools to describe impulsive interferences. Sα distributions have some very important properties such as the stability property and generalized central limit theorem. However, since variances of the SαS processes ($\alpha < 2$) are not finite, the usual second–order statistics–based analysis or algorithmic development is inappropriate. Instead of variance, the fractional lower order moments (FLOM), which are all moments of order less than $\alpha < 2$, should be used.

In this chapter, we assumed that the interference to be mitigated is characterized by the SαS distributions. Some of described AIM methods are based on FLOM criteria and the rest on the LMS algorithm or its modifications, which are derived from second–order statistics. Although usual statistical analysis of the LMS–based methods is not possible, LMS–type methods can be utilized for SαS processes due to the fact that the LMS algorithm uses an instant gradient estimate.

Appendix A

Derivatives of $\zeta\left(v, \epsilon_k\right)$

Letting $\epsilon_k = x(k) - \mathbf{H}\bar{\mathbf{u}}(k)$, $\sigma_k^2 = \mathbf{H}\mathbf{M}(k)\mathbf{H}^T$, and $f_{v,\epsilon_k} = \frac{(v-\epsilon_k)^2}{2\sigma_k^2}$, we define the function ζ as

$$\zeta = \exp\{-f_{v,\epsilon_k}\}. \tag{A.1}$$

Note that the function ζ is used without variables v and ϵ_k for simplicity. Then, the first derivative of ζ with respect to v is given by

$$\begin{aligned} \zeta^{(1)} &= -\, f_{v,\epsilon_k}^{(1)} \exp\{-f_{v,\epsilon_k}\} \\ &= -\, f_{v,\epsilon_k}^{(1)} \zeta. \end{aligned} \tag{A.2a}$$

where

$$f_{v,\epsilon_k}^{(n)} = \frac{\partial^n}{\partial v^n} f_{v,\epsilon_k} \tag{A.2b}$$

for $n = 1, 2, 3, \ldots$. Its second derivative becomes

$$\zeta^{(2)} = -\, f_{v,\epsilon_k}^{(2)} \zeta - f_{v,\epsilon_k}^{(1)} \zeta^{(1)}. \tag{A.3}$$

And, the third derivative of the function ζ becomes

$$\begin{aligned} \zeta^{(3)} &= -f_{v,\epsilon_k}^{(3)} \zeta - f_{v,\epsilon_k}^{(2)} \zeta^{(1)} - f_{v,\epsilon_k}^{(2)} \zeta^{(1)} - f_{v,\epsilon_k}^{(1)} \zeta^{(2)} \\ &= -\, f_{v,\epsilon_k}^{(3)} \zeta - 2 f_{v,\epsilon_k}^{(2)} \zeta^{(1)} - f_{v,\epsilon_k}^{(1)} \zeta^{(2)}. \end{aligned} \tag{A.4}$$

Its fourth derivative is given by

$$\begin{aligned}\zeta^{(4)} &= -f^{(4)}_{v,\epsilon_k}\zeta - f^{(3)}_{v,\epsilon_k}\zeta^{(1)} - 2f^{(3)}_{v,\epsilon_k}\zeta^{(1)} - 2f^{(2)}_{v,\epsilon_k}\zeta^{(2)} - f^{(2)}_{v,\epsilon_k}\zeta^{(2)} - f^{(1)}_{v,\epsilon_k}\zeta^{(3)} \\ &= -f^{(4)}_{v,\epsilon_k}\zeta - 3f^{(3)}_{v,\epsilon_k}\zeta^{(1)} - 3f^{(2)}_{v,\epsilon_k}\zeta^{(2)} - f^{(1)}_{v,\epsilon_k}\zeta^{(3)}. \qquad \text{(A.5)}\end{aligned}$$

And, the fifth derivative of the function ζ becomes

$$\begin{aligned}\zeta^{(5)} &= -f^{(5)}_{v,\epsilon_k}\zeta - f^{(4)}_{v,\epsilon_k}\zeta^{(1)} - 3f^{(4)}_{v,\epsilon_k}\zeta^{(1)} - 3f^{(3)}_{v,\epsilon_k}\zeta^{(2)} - 3f^{(3)}_{v,\epsilon_k}\zeta^{(2)} \\ &\quad - 3f^{(2)}_{v,\epsilon_k}\zeta^{(3)} - f^{(2)}_{v,\epsilon_k}\zeta^{(3)} - f^{(1)}_{v,\epsilon_k}\zeta^{(4)} \\ &= -f^{(5)}_{v,\epsilon_k}\zeta - 4f^{(4)}_{v,\epsilon_k}\zeta^{(1)} - 6f^{(3)}_{v,\epsilon_k}\zeta^{(2)} - 4f^{(2)}_{v,\epsilon_k}\zeta^{(3)} - f^{(1)}_{v,\epsilon_k}\zeta^{(4)}. \qquad \text{(A.6)}\end{aligned}$$

In general, the n–th order derivatives of ζ can be represented by

$$\zeta^{(n)} = -\sum_{i=1}^{n} a_i\, f^{(i)}_{v,\epsilon_k}\zeta^{(n-i)} \qquad \text{for } n = 1, 2, 3, \ldots \qquad \text{(A.7)}$$

where $\zeta^{(0)} = \zeta$ and the coefficients $\{a_i,\ i = 1, 2, \ldots, n\}$ are the constant coefficients of the binomial expansion $(x+1)^{n-1}$. Since $f^{(n)}_{v,\epsilon_k} = 0$ for $n \geq 3$, the representation of general n–th derivatives of ζ in (A.7) can be of the following recursive form:

$$\begin{aligned}\zeta^{(n)} &= -f^{(1)}_{v,\epsilon_k}\zeta^{(n-1)} - (n-1)f^{(2)}_{v,\epsilon_k}\zeta^{(n-2)} \\ &= -\frac{(v-\epsilon_k)}{\sigma_k^2}\zeta^{(n-1)} - \frac{(n-1)}{\sigma_k^2}\zeta^{(n-2)} \qquad \text{(A.8a)}\end{aligned}$$

where

$$\zeta^{(n)} = \begin{cases} 0, & n < 0 \\ \zeta, & n = 0 \\ \frac{\partial^n \zeta}{\partial v^n}, & n \geq 1. \end{cases} \qquad \text{(A.8b)}$$

In stead of the recursive form, each n–th order derivative of ζ with respect to v can be represented as follows

$$\frac{\partial^n \zeta}{\partial v^n} = \begin{cases} \zeta \sum_{i=1}^{m+1} \frac{a_i^{(m)}(v-\epsilon_k)^{2(i-1)}}{\sigma_k^{2(m+i-1)}}, & \text{for } n = 2m \\ \zeta \sum_{i=1}^{m+1} \frac{b_i^{(m)}(v-\epsilon_k)^{2i-1}}{\sigma_k^{2(m+i)}}, & \text{for } n = 2m+1 \end{cases} \qquad \text{(A.9)}$$

where m is a nonnegative integer. The coefficients $\{a_i^{(m)},\ 1 \le i \le m+1\}$ and $\{b_i^{(m)},\ 1 \le i \le m+1\}$ are constants with $a_{m+1}^{(m)} = 1$ and $b_{m+1}^{(m)} = -1$. There seems, unfortunately, no explicit relationship either between $a_i^{(m_1)}$ and $a_j^{(m_2)}$ or between $b_i^{(m_1)}$ and $b_j^{(m_2)}$ for any m_1 and m_2. After expanding each term, these coefficients can be obtained. The coefficients $\{a_i^{(m)}\}$ and $\{b_i^{(m)}\}$ for $m = 0, 1, 2, 3, 4, 5$ can be obtained by

$$\mathbf{a}^{(m)} = \begin{cases} [\,1\,]^T, & m = 0 \\ [\,-1, 1\,]^T, & m = 1 \\ [\,3, -6, 1\,]^T, & m = 2 \\ [\,-15, 45, -15, 1\,]^T, & m = 3 \\ [\,105, -420, 210, -28, 1\,]^T, & m = 4 \\ [\,-945, 4725, -3150, 630, -45, 1\,]^T, & m = 5 \end{cases} \tag{A.10a}$$

$$\mathbf{b}^{(m)} = \begin{cases} [\,-1\,]^T, & m = 0 \\ [\,3, -1\,]^T, & m = 1 \\ [\,-15, 10, -1\,]^T, & m = 2 \\ [\,105, -105, 21, -1\,]^T, & m = 3 \\ [\,-945, 1260, -378, 36, -1\,]^T, & m = 4 \\ [\,10395, -17325, 6930, -990, 55, -1\,]^T, & m = 5 \end{cases} \tag{A.10b}$$

where $\mathbf{a}^{(m)}$ and $\mathbf{b}^{(m)}$ are corresponding coefficient vectors given by

$$\begin{aligned} \mathbf{a}^{(m)} &= \left[\, a_1^{(m)}, a_2^{(m)}, \cdots, a_{m+1}^{(m)} \,\right]^T \\ \mathbf{b}^{(m)} &= \left[\, b_1^{(m)}, b_2^{(m)}, \cdots, b_{m+1}^{(m)} \,\right]^T . \end{aligned} \tag{A.10c}$$

Appendix B

Derivations of $\rho_k(\epsilon_k)$

For the expression of the nonlinear function $\{\rho_k(\epsilon_k)\}$ given in (6.41), its denominator and numerator have to be obtained. Since the denominator of $\{\rho_k(\epsilon_k)\}$ is the same as $f(x(k) \mid \mathbf{X})$ given in (6.40), what has to be obtained is the expression of the numerator of $\{\rho_k(\epsilon_k)\}$ that is the derivative of the denominator with respect to $\{x(k)\}$ (or $\{\epsilon_k\}$). Let $N_k^g(\epsilon_k)$ be the numerator of $\{\rho_k(\epsilon_k)\}$. Then, it becomes

$$\begin{aligned} N_k^g(x(k)) &= \frac{\partial}{\partial x(k)} f(x(k) \mid \mathbf{X}(k-1)) \\ &= \frac{1}{2\sqrt{2}\sigma_k \Gamma\left(\frac{1}{\alpha}\right)} \sum_{n=0}^{\infty} \frac{\Gamma\left(\frac{2n+1}{\alpha}\right) \beta^{2n}}{(2n)!} \left[\xi_n(1, \epsilon_k) + \xi_n(-1, \epsilon_k) \right] \end{aligned} \quad \text{(B.1a)}$$

where

$$\xi_n(\pm 1, \epsilon_k) = \frac{\partial}{\partial \epsilon_k} \zeta^{(2n)}(\pm 1, \epsilon_k). \quad \text{(B.1b)}$$

The expression of $\zeta^{(2n)}(\pm 1, \epsilon_k)$ is given in (A.9). Using the expression, the function $\xi_n(\pm 1, \epsilon_k)$ becomes

$$\xi_n(\pm 1, \epsilon_k) = \zeta^{(1)}(\pm 1, \epsilon_k) \sum_{i=1}^{n+1} \frac{a_i^{(n)} (\pm 1 - \epsilon_k)^{2(i-1)}}{\sigma_k^{2(n+i-1)}} + \zeta(\pm 1, \epsilon_k) \frac{\partial}{\partial \epsilon_k} \sum_{i=1}^{n+1} \frac{a_i^{(n)} (\pm 1 - \epsilon_k)^{2(i-1)}}{\sigma_k^{2(n+i-1)}}. \quad \text{(B.2)}$$

Using (A.2a), it can be rewritten as

$$\xi_n(\pm 1, \epsilon_k) = \zeta(\pm 1, \epsilon_k) \left[\sum_{i=1}^{n+1} \frac{a_i^{(n)} (\pm 1 - \epsilon_k)^{2i-1}}{\sigma_k^{2(n+i)}} - \sum_{i=2}^{n+1} \frac{2(i-1) a_i^{(n)} (\pm 1 - \epsilon_k)^{2i-3}}{\sigma_k^{2(n+i-1)}} \right] \quad \text{(B.3)}$$

with the coefficients $\{a_i^{(n)},\ 1 \le i \le n+1\}$ given in Appendix A. Using this expression of the numerator and the denominator expression in (6.40), the nonlinear function $\{\rho_k(\epsilon_k)\}$ becomes

$$\rho_k(\epsilon_k) \triangleq - \frac{\frac{\partial}{\partial x(k)} f(x(k) \mid \mathbf{X}(k-1))}{f(x(k) \mid \mathbf{X}(k-1))} = \frac{\sum_{n=0}^{\infty} \frac{\Gamma(\frac{2n+1}{\alpha}) \beta^{2n}}{(2n)!} \left[\xi_n(1, \epsilon_k) + \xi_n(-1, \epsilon_k) \right]}{\sum_{n=0}^{\infty} \frac{\Gamma(\frac{2n+1}{\alpha}) \beta^{2n}}{(2n)!} \left[\zeta^{(2n)}(1, \epsilon_k) + \zeta^{(2n)}(-1, \epsilon_k) \right]} \quad \text{(B.4)}$$

where $\epsilon_k = x(k) - \mathbf{H}\bar{\mathbf{u}}(k)$. The expressions of $\zeta^{(2n)}(\pm 1, \epsilon_k)$ and $\xi_n(\pm 1, \epsilon_k)$ are given in (A.9) and (B.3), respectively.

Bibliography

[Abu-El-Haija and Peterson, 1979] A. I. Abu–El–Haija and A. M. Peterson, "A Structure for Digital Notch Filters," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–27, no. 2, pp. 193–195, April 1979.

[Agarwal and Burrus, 1975] R. C. Agarwal and C. S. Burrus, "New Recursive Digital Filter Structures Having Very Low Sensitivity and Roundoff Noise," *IEEE Trans. Circuits Syst.*, vol. CAS–22, no. 12, pp. 921–927, Dec. 1975.

[Antoniou, 1979] A. Antoniou, *Digital Filters: Analysis and Design,* McGraw–Hill Series in Electrical Engineering, McGraw–Hill Book Company, New York, New York, 1979.

[Antoniou, 1983] A. Antoniou, "New Improved Method for the Design of Weighted–Chebyshev, Nonrecursive, Digital Filters," *IEEE Trans. Circuits Syst.*, vol. CAS–30, no. 10, pp. 740–749, Oct. 1983.

[Arikan *et al.*, 1994] O. Arikan, A. Enis Çetin, and E. Erzin, "Adaptive Filtering for Non–Gaussian Stable Processes," to be published in *IEEE Signal Process. Letter*, 1994.

[Benveniste and Goursat, 1984] A. Benveniste and M. Goursat, "Blind Equalizers," *IEEE Trans. Commun.*, vol. COM–32, no. 7. pp. 871–883, 1984.

[Bhaskar Rao and Kung, 1984] D. V. Bhaskar Rao and S. Kung, "Adaptive Notch Filtering for the Retrieval of Sinusoids in Noise," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–32, no. 4, pp. 791–802, Aug. 1984.

[Bird, 1980] G. J. A. Bird, *Design of Continuous and Digital Electronic Systems,* McGraw–Hill Book Company (UK) Limited, Berkshire, England, 1980.

[Boctor, 1975a] S. A. Boctor, "Design of a Third–Order Single Amplifier Filter," *IEEE Trans. Circuits Syst.*, vol. CAS–22, no. 4, pp. 329–334, April 1975.

[Boctor, 1975b] S. A. Boctor, "A Novel Second–Order Canonical *RC*–Active Realization of High–Pass Notch Filter," *IEEE Trans. Circuits Syst.*, vol. CAS–22, no. 5, pp. 397–404, May 1975.

[Boctor, 1975c] S. A. Boctor, "Single Amplifier Functionally Tunable Low–Pass–Notch Filter," *IEEE Trans. Circuits Syst.*, vol. CAS–22, no. 11, pp. 875–881, Nov. 1975.

[Boll and Pulsipher, 1980] S. F. Boll and D. C. Pulsipher, "Suppression of Acoustic Noise in Speech Using Two Microphone Adaptive Noise Cancellation," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–28, no. 6, pp. 752–753, Dec. 1980.

[Cadzow, 1973] J. A. Cadzow, *Discrete–Time Systems: An Introduction with Interdisciplinary Applications,* Prentice–Hall, Inc., Englewood Cliffs, New Jersey, 1973.

[Cadzow, 1974] J. A. Cadzow, "Digital Notch Filter Design Procedure," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–22, no. 1, pp. 10–15, Feb. 1974.

[Carney, 1963] R. Carney, "Design of a Digital Notch Filter with Tracking Requirements," *IEEE Trans. Space Electronics and Telemetry,* vol. SET–9, pp. 109–114, Dec. 1963.

[Chan *et al.*, 1989] C. K. Chan, M. R. Petraglia, and J. J. Shynk, "Frequency–Domain Implementation of the Constant Modulus Algorithm," *Proc. 23rd Asilomar Conference on Signal, Systems, & Computer,* Pacific Grove, CA, pp. 663–669, Oct. 1989.

[Chen *et al*, 1991] Y. Chen, C. L. Nikias, J. G. Proakis, "CRIMNO: Criterion with Memory Nonlinearity for Blind Equalization," *Proc. Intern. Signal Processing Workshop on Higher–Order Statistics,* Chamrousse, France, pp. 87–90, July 1991.

[Chen, *et al.*, 1992] Y. Chen, C. L. Nikias, and J. G. Proakis", "Blind Equalization with Criterion with Memory Nonlinearity," *Optical Engineering,* vol. 31, pp. 1200–1210, June 1992.

[Chen and Nikias, 1992] Y. Chen and C. L. Nikias, "Fractionally–Spaced Blind Equalization with CRIMNO Algorithm," *Proc. IEEE MILCOM'92 Conference,* San Diego, CA, pp. 9.3.1–9.3.5, Oct. 1992.

[Chiang and Nikias, 1988] H. Chiang and C. L. Nikias, "Cumulant–Based Adaptive Time Delay Estimation," *Proc. 22nd Asilomar Conference on Signal, Systems, & Computer,* Pacific Grove, CA, pp. 15–19, Oct. 1988.

[Chiang and Nikias, 1990a] H. Chiang and C. L. Nikias, "Adaptive Deconvolution and Identification of Nonminimum Phase FIR Systems Based on Cumulants," *IEEE Trans. Automat. Contr.,* vol. AC–35, no. 1, pp. 36–47, Jan. 1990.

[Chiang and Nikias, 1990b] H. Chiang and C. L. Nikias, "A New Method for Adaptive Time Delay Estimation for Non-Gaussian Signals," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-38, pp. 209–219, Feb. 1990.

[Clarkson, 1993] P. M. Clarkson, *Optimal and Adaptive Signal Processing,* CRC Press, Boca Ranton, Florida, 1993.

[Coker and Simkins, 1980] M. J. Coker and D. N. Simkins, "A Nonlinear Adaptive Noise Canceller," *Proc. ICASSP'80*, Denver, CO, pp. 470–473, April 1980.

[Compton, 1988] R. T. Compton, Jr., *Adaptive Antennas,* Prentice–Hall, Inc., Englewood Cliffs, New Jersey, 1988.

[Conolly and Su, 1986] S. Conolly and Y. Su, "State–Space Analysis of the Adaptive Notch Filter," *Proc. IEEE,* vol. 74, no. 1, pp. 219–221, Jan. 1986.

[Cummiskey *et al.*, 1973] P. Cummiskey, N. S. Jayant, and J. L. Flanagan, "Adaptive Quantization in Differential PCM Coding of Speech," *Bell Syst. Tech. J.*, vol. 52, no. 7, pp. 1105–1118, Sept. 1973.

[Cutler, 1952] C. C. Cutler, "Differential Quantization of Communication Signals," U.S. Patent 2 605 361, July 1952.

[Dandawate and Giannakis, 1989] A. V. Dandawate and G. B. Giannakis, "A Triple Cross-Correlation Approach for Enhancing Noisy Signals," *Proc. Workshop on Higher–Order Spectral Analysis*, Vail, CO, pp. 212–216, June 1989.

[Das and Milstein, 1978] P. K. Das and L. B. Milstein, "Variable Format Radar Receiver Using a SAW Convolver," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-14, no. 6, pp. 843–852, Nov. 1978.

[Datta, 1986] S. Datta, *Surface Acoustic Wave Devices,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1986.

[Davenport and Root, 1987] W. B. Davenport, Jr. and W. L. Root, *An Introduction to the Theory of Random Signals and Noise,* IEEE Press, New York, 1987.

[Davidovici and Kanterakis, 1989] S. Davidovici and E. G. Kanterakis, "Narrow–Band Interference Rejection Using Real–Time Fourier Transforms," *IEEE Trans. Commun.,* vol. COM–37, no. 7. pp. 713–722, July, 1989.

[Ding and Yu, 1986] H. Ding and C. Yu, "Adaptive Lattice Noise Canceller and Optimal Step Size," *Proc. ICASSP'86,* Tokyo, Japan, pp. 2939–2942, ??? 1986.

[Douglas and Meng, 1990] S. C. Douglas and T. H.–Y. Meng, "Optimum Error Nonlinearities for LMS Adaptation," *Proc. ICASSP'90,* Albuquerque, NM, pp. 1421–1424, April 1990.

[Douglas and Meng, 1991a] S. C. Douglas and T. H.–Y. Meng, "Optimum Error Quantization for LMS Adaptation," *IEEE Pacific Rim Conf. on Communications, Computers, and Signal Processing,* Victoria, B. C., Canada, pp. 704–708, May, 1991.

[Douglas and Meng, 1991b] S. C. Douglas and T. H.–Y. Meng, "A Nonlinear Error Adaptive Notch Filter for Separating Two Sinusoidal Signals," *Proc. 25th Asilomar Conference on Signal, Systems, & Computer,* Pacific Grove, CA, pp. 673–677, Nov. 1991.

[Douglas and Meng, 1992] S. C. Douglas and T. H.–Y. Meng, "The Optimum Scalar Data Nonlinearity in LMS Adaptation for Arbitrary

I. I. D. Inputs," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–40, no. 6, pp. 1566–1570, June 1992.

[Duttweiler, 1982] D. L. Duttweiler, "Adaptive Filter Performance with Nonlinearities in the Correlation Multiplier," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–30, no. 4, pp. 578–586, Aug. 1982.

[Falconer and Ljung, 1978] D. D. Falconer and L. Ljung, "Application of Fast–Kalman Estimation to Adaptive Equalization," *IEEE Trans. Commun.*, vol. COM–26, no. 10, pp. 1439–1446, Oct. 1978.

[Franks, 1970] L. E. Franks, "Evaluation of the Effects of Notch Filters on Digital Data Transmission," *IEEE Trans. Commun. Technology*, pp. 447–449, Aug. 1970.

[Friedlander, 1982a] B. Friedlander, "A Modified Prefilter for Some Recursive Parameter Estimation Algorithms," *IEEE Trans. Automat. Contr.*, vol. AC–27, no. 1, pp. 232–235, Feb. 1982.

[Friedlander, 1982b] B. Friedlander, "Lattice Methods for Adaptive Processing," *Proc. IEEE*, vol. 70, no. 8, pp. 829–867, Aug. 1982.

[Friedlander, 1982c] B. Friedlander, "Lattice Methods for Spectral Estimation," *Proc. IEEE*, vol. 70, no. 9, pp. 990–1017, Sept. 1982.

[Friedlander and Smith, 1984] B. Friedlander and J. O. Smith, "Analysis and Performance Evaluation of an Adaptive Notch Filter," *IEEE Trans. Inform. Theory*, vol. IT–30, no. 2, pp. 283–295, March 1984.

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

[Garth and Poor, 1992] L. M. Garth and H. V. Poor, "Narrowband Interference Suppression in Impulsive Channels," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES–28, no. 1, pp. 15–34, Jan. 1992

[Gentleman and Kung, 1981] W. M. Gentleman and H. T. Kung, "Matrix Triangularization by Systolic Arrays," *Proc. SPIE,* vol. 298, Real Time Signal Processing IV, pp. 298–303, 1981.

[Gevargiz *et al.*, 1989] J. Gevargiz and P. K. Das and L. B. Milstein, "Adaptive Narrow–band Interference Rejection in a DS Spread–Spectrum Intercept Receiver Using Transform Domain Signal Processing Techniques," *IEEE Trans. Commun.,* vol. COM–37, no. 12, pp. 1359–1365, Dec. 1989.

[Glover, 1977] J. R. Glover, Jr., "Adaptive Noise Canceling Applied to Sinusoidal Interferences," *IEEE Trans. Acoust. Speech Signal Process.,* vol. ASSP–25, pp. 484–491, Dec. 1977.

[Godard, 1974] D. Godard, "Channel Equalization Using a Kalman Filter for Fast Data Transmission," *IBM J. Res. Develop.,* vol. 18, pp. 267–273, 1974.

[Godard, 1980] D. N. Godard, "Self–Recovering Equalization and Carrier Tracking in Two–Dimensional Data Communication Systems," *IEEE Trans. Commun.,* vol. COM–28, no. 11, pp. 1867–1875, Nov. 1980.

[Goldberg and Iltis, 1988] S. H. Goldberg and R. A. Iltis, "Joint Channel Equalization and Interference Rejection Using a Modified Constant Modulus Algorithm," *Proc. IEEE MILCOM'88 Conference,* San Diego, CA, pp. 4.4.1–4.4.5, Oct. 1988.

[Golden, 1968] R. M. Golden, "Digital Filter Synthesis by Sampled–Data Transformation, *IEEE Trans. Audio Electroacoust.*, vol. AU–16, no. 3, pp. 321–329, Sept. 1968.

[Gooch and Daellenbach, 1989] R. P. Gooch and B. Daellenbach, "Prevention of Interference Capture in a Blind (CMA–Based) Adaptive Receive Filter," *Proc. 23rd Asilomar Conference on Signal, Systems, & Computer*, Pacific Grove, CA, pp. 898–902, Oct. 1989.

[Goodwin and Payne, 1977] G. C. Goodwin and R. L. Payne, *Dynamic System Identification: Experiment Design and Data Analysis,* Academic Press, New York, 1977.

[Gray and Markel, 1973] A. H. Gray, Jr. and J. D. Markel, "Digital Lattice and Ladder Filter Synthesis," *IEEE Trans. Audio Electroacoust.*, vol. AU–21, no. 6, pp. 491–500, Dec. 1973.

[Gray, 1979] W. C. Gray, *Variable Norm Deconvolution*, Ph.D thesis, Stanford University, Stanford, CA, 1979.

[Griffiths, 1977] L. J. Griffiths, "A Continuously–Adaptive Filter Implemented as a Lattice Structure," *Proc. ICASSP'77,* Hartford, CT, pp. 683–686, May, 1977.

[Griffiths, 1978] L. J. Griffiths, "An Adaptive Lattice Structure for Noise–Cancelling Applications," *Proc. ICASSP'78,* Tulsa, OK, pp. 87–90, April, 1978.

[Haimovich and Bar-Ness, 1991] A. M. Haimovich and Y. Bar–Ness, "An Eigenanalysis Interference Canceler," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–39, pp. 76–84, Jan. 1991.

[Harris, 1978] F. J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," *Proc. IEEE,* vol. 66, pp. 51–83, Jan. 1978.

[Hatzinakos and Nikias, 1990] D. Hatzinakos and C. L. Nikias, "Blind Equalization Using a Tricepstrum Based Algorithm," *IEEE Trans. Commun.*, vol. COM–39, no. 5, pp. 669–682, May, 1991.

[Haykin, 1991] S. Haykin, *Adaptive Filter Theory*, second ed., Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1991.

[Hays and Hartmann, 1976] R. M Hays and C. S. Hartmann, "Surface–Acoustic–Wave Devices for Communications," *Proc. IEEE*, vol. 64, no. 5, pp. 652–671, May, 1976.

[Helstrom, 1968] C. W. Helstrom, *Statistical Theory of Signal Detection,* 2nd ed., Pergamon Press, London, 1968.

[Hirano *et al.*, 1974] K. Hirano, S. Nishimura, and S. K. Mitra, "Design of Digital Notch Filters," *IEEE Trans. Circuits Syst.,* vol. CAS–21, no. 4, pp. 540–546, July, 1974.

[Hodgkiss and Presley, 1980] W. S. Hodgkiss, Jr. and J. A. Presley, Jr., "Adaptive Tracking of Multiple Sinusoids Whose Power Levels are Widely Separated," *IEEE Trans. Circuits Syst.,* vol. CAS–28, no. 6, pp. 550–561, June, 1981.

[Honig and Messerschmitt, 1981] M. L. Honig and D. G. Messerschmitt, "Convergence Properties of an Adaptive Digital Lattice Filter, *IEEE Trans. Acoust. Speech Signal Process.,* vol. ASSP–29, pp. 642–653, June, 1981.

[Hsu and Giordano, 1978] F. M. Hsu and A. A. Giordano, "Digital Whitening Techniques for Improving Spread Spectrum Communications Performance in the Presence of Narrowband Jamming and Interference," *IEEE Trans. Commun.*, vol. COM–26, no. 2, pp. 209–216, Feb. 1978.

[Huelsman and Raghunath, 1974] L. P. Huelsman and S. Raghunath, "The Values of the Parameters of Some Multilayer Distributed *RC* Null Networks," *IEEE Trans. Circuits Syst.*, vol. CAS–21, no. 6, pp. 804–805, Nov. 1974.

[Hush *et al.*, 1986] D. R. Hush *et al.*, "An Adaptive IIR Structure for Sinusoidal Enhancement, Frequency Estimation, and Detection," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–34, no. 6, pp. 1380–1390, Dec. 1986.

[Ibukun, 1966] O. Ibukun, "Structural Aspects of Atmospheric Radio Noise in the Tropics," *Proc. IEEE,* vol. 54, pp. 361–367, 1966.

[Iltis and Milstein, 1984] R. A. Iltis and L. B. Milstein, "Performance Analysis of Narrow–band Interference Rejection Techniques in DS Spread–Spectrum Systems," *IEEE Trans. Commun.*, vol. COM–32, no. 11, pp. 1169–1177, Nov. 1984.

[Iltis and Goldberg, 1987] R. A. Iltis and S. H. Goldberg, "Joint Interference Rejection/Channel Equalization in DS Spread–Spectrum Receivers Using the CMA Equalizer and Maximum–Likelihood Techniques," *Proc. IEEE MILCOM'87 Conference*, Washington D. C., pp. 4.2.1–4.2.5, Oct. 1987.

[Jack *et al.*, 1980] M. A. Jack, P. M. Grant, and J. H. Colins, "The Theory, Design, and Applications of Surface Acoustic Wave Fourier–

Transform Processors," *Proc. IEEE,* vol. 68, no. 4, pp. 450–468, April 1980.

[Jamali and Wood, 1990] H. Jamali and S. L. Wood, "Error Surface Analysis for the Complex Constant Modulus Adaptive Algorithm," *Proc. 24th Asilomar Conference on Signal, Systems, & Computer,* Pacific Grove, CA, pp. 248–252, Nov. 1990.

[Johnson, 1984] C. R. Johnson, Jr., "Adaptive IIR Filtering: Current Results and Open Issues," *IEEE Trans. Inform. Theory,* vol. IT–30, no. 2, pp. 237–250, March, 1984.

[Kailath, 1974] T. Kailath, "A View of Three Decades of Linear Filtering Theory," *IEEE Trans. Information Theory,* vol. IT–20, no. 2, pp. 146–181, March 1974.

[Kailath, 1981] T. Kailath, *Lectures on Wiener and Kalman Filtering,* International Center for Mechanical Science, Springer–Verlag, Wien–New York, 1981.

[Kalman, 1960] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Trans. ASME, J. Basic Eng. Series D,* vol. 82, pp. 35–46, March 1960.

[Kalman and Bucy, 1961] R. E. Kalman and R. Bucy, "New Results in Linear Filtering and Prediction Theory," *Trans. ASME, J. Basic Eng. Series D,* vol. 83, pp. 95–108, March 1961.

[Kanter and Steiger, 1974] M. Kanter and W. L. Steiger, "Regression and Autoregression with Infinite Variance," *Adv. Appl. Prob.* vol. 6, pp. 768–783, 1974.

[Kasparis, 1991] T. Kasparis, "Frequency Independent Sinusoidal Suppression Using Median Filters," *Proc. ICASSP'91,* Toronto, Canada, pp. 1969–1972, May 1991.

[Kasparis *et al.*, 1991] T. Kasparis, M. Georgiopoulos, and E. Payne, "Non–Linear Filtering Techniques for Narrow–Band Interference Rejection in Direct Sequence Spread–Spectrum Systems," *Proc. IEEE MILCOM'91 Conference,* ??, pp. 17.1.1–17.1.5, Nov. 1991.

[Kay, 1988] S. M. Kay, *Modern Spectral Estimation*, Prentice–Hall, Inc., Englewood Cliffs, New Jersey, 1988.

[Knapp adn Carter, 1976] C. H. Knapp and G. C. Carter, "The Generalized Correlation Method for Estimation of Time Delay," *IEEE Trans. Acoust. Speech Signal Process.,* vol. ASSP–24", no. 4, pp. 320–327, Aug. 1976.

[Kowalski, 1977] A. M. Kowalski, "Adaptive Filterfor Interference Suppression," *IEEE Proc. Nat. Telecommun. Conf.,* pp. 04:5-1–04:5-6, 1977.

[Kung and Bhaskar Rao, 1982] S. Kung and D. V. Bhaskar Rao, "Analysis and Implementation of the Adaptive Notch Filter for Frequency Estimation," *Proc. ICASSP'82*, Paris, pp. 663–666, May 1982.

[Kwan and Martin, 1989] T. Kwan and K. Martin, "Adaptive Detection and Enhancement of Multiple Sinusoids Using a Cascade IIR Filter," *IEEE Trans. Circuits Syst.,* vol. CAS–36, no. 7, pp. 937–947, July 1989.

[Lam, 1979] H. Y-F. Lam, *Analog and Digital FIlters: Design and Realization*, Prentice–Hall Inc., Englewood Cliffs, New Jersey, 1979.

[Lev–Ari *et al.*, 1984] H. Lev–Ari, T. Kailath, and J. Cioffi, "Least–Squares Adaptive Lattice and Transversal Filters: A Unified Geometric Theory," *IEEE Trans. Inform. Theory,* vol. IT–30, no. 2, pp. 222–236, March 1984.

[Lévy, 1925] P. Lévy, *Calcul des Probabilités,* Gauthier–Villars, Paris, 1925.

[Li and Milstein, 1982] L. Li and L. B. Milstein, "Rejection of Narrow–Band Interference in PN Spread–Spectrum Systems Using Transversal Filters," *IEEE Trans. Commun.*, vol. COM–30, no. 11, pp. 925–928, May 1982.

[Li and Milstein, 1983] L. Li and L. B. Milstein, "Rejection of Pulsed CW Interference in PN Spread–Spectrum Systems Using Complex Adaptive Filters," *IEEE Trans. Commun.*, vol. COM–31, no. 1, pp. 10–20, Jan. 1983.

[Ljung, 1981] L. Ljung, "Analysis of a General Recursive Prediction Error Identification Algorithm," *Automatica*, vol. 17, no. 1, pp. 89–100, Jan. 1981.

[Lucky, 1965] R. W. Lucky, "Automatic Equalization for Digital Communication," *Bell Syst. Tech. J.,* vol. 44, pp. 547–588, Apr. 1965.

[Lucky, 1966] R. W. Lucky, "Techniques for Adaptive Equalization of Digital Communication Systems," *Bell Syst. Tech. J.,* vol. 45 pp. 255–286, 1966.

[Manines and Paige, 1976] J. D. Maines and E. G. S. Paige, "Surface–Acoustic–Wave Devices for Signal Processing Applications," *Proc. IEEE*, vol. 64, no. 5, pp. 639–652, May 1976.

[Makhoul, 1977] J. Makhoul, "Stable and Efficient Lattice Lattice Methods for Linear Prediction," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–25, no. 5, pp. 423–428, Oct. 1977.

[Makhoul, 1978] J. Makhoul, "A Class of All–Zero Lattice Digital FIlters: Properties and Applications," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–26, no. 4, pp. 304–314, Aug. 1978.

[Marple, 1987] S. L. Marple, *Digital Spectral Analysis with Applications*, Prentice–Hall, Inc., Englewood Cliffs, New Jersey, 1987.

[Martin and Sun, 1986] K. W. Martin and M. T. Sun, "Adaptive Filters Suitable for Real–Time Spectral Analysis," *IEEE Trans. Circuits Syst.*, vol. CAS–33, no. 2, pp. 218–229, Feb. 1986.

[Masreliez, 1975] C. J. Masreliez, "Approximate Non-Gaussian Filtering with Linear State and Observation Relations," *IEEE Trans. Automat. Contr.*, vol. AC–20, pp. 107–110, Feb. 1975.

[Masry, 1984] E. Masry, "Closed–Form Analytical Results for the Rejection of Narrow–Band Interference in PN Spread–Spectrum Systems–Part I: Linear Prediction Filters," *IEEE Trans. Commun.*, vol. COM–32, no. 8, pp. 888–896, Aug. 1984.

[Masry, 1985] E. Masry, "Closed–Form Analytical Results for the Rejection of Narrow–Band Interference in PN Spread–Spectrum Systems–Part II: Linear Interpolation Filters," *IEEE Trans. Commun.*, vol. COM–33, no. 1, pp. 10–19, Jan. 1985.

[McNally, 1970] R. L. McNally, "Active Filter Realization Using Finite–Gain Voltage Amplifiers," *IEEE Trans. Circuit Theory,* vol. CT–17, pp. 445–448, Aug. 1970.

[Mendel, 1987] J. M. Mendel, *Lessons in Digital Estimation Theory,* Prentice–Hall, Inc., Englewood Cliffs, New Jersey, 1987.

[Mendel, 1991] J. M. Mendel, "Tutorial on Higher–Order Statistics (Spectra) in Signal Processing and System Theory: Theoretical Results and Some Applications," *Proc. IEEE,* vol. 79, no. 3, pp. 278–305, March 1991.

[Mendoza *et al.*, 1989a] R. Mendoza, J. H. Reed, and T. C. Hsia, "Interference Rejection Using a Hybrid of a Constant Modulus Algorithm and the Spectral Correlation Discriminator," *Proc. IEEE MILCOM'89 Conference,* Boston, MA, pp. 28.2.1–28.2.7, Oct. 1989.

[Mendoza *et al.*, 1989b] , R. Mendoza, J. H. Reed, T. C. Hsia, and B. G. Agee, "Interference Rejection Using Time–Dependent Constant Modulus Algorithms," *Proc. 23rd Asilomar Conference on Signal, Systems, & Computer,* Pacific Grove, CA, pp. 273–278, Oct. 1989.

[Middleton, 1960] D. Middleton, *An Introduction to Statistical Communication Theory,* McGraw–Hill, New York, 1960.

[Middleton, 1972] D. Middleton, "Statistical–Physical Models of Urban Radio–Noise Environments, Part I: Foundations," *IEEE Trans. Electromagn. Compat.,* vol. EMC–14, pp. 38–56, May 1972.

[Middleton, 1973] D. Middleton, "Man–Made Noise in Urban Environments and Transportation Systems," *IEEE Trans. Commun.,* vol. COM–21, pp. 1232–1241, Nov. 1973.

[Middleton, 1977] D. Middleton, "Statistical–Physical Models of Electromagnetic Interference," *IEEE Trans. Electromagn. Compat.,* vol. EMC–19, no. 3, pp. 106–127, Aug. 1977.

[Middleton, 1979a] D. Middleton, "Procedures for Determining the Parameters of the First–Order Canonical Models of Class A and Class B Electromagnetic Interference," *IEEE Trans. Electromagn. Compat.*, vol. EMC–21, pp. 190–208, Aug. 1979.

[Middleton, 1979b] D. Middleton, "Canonical non–Gaussian Noise Models: Their Implications for Measurement and for Prediction of Receiver Performance," *IEEE Trans. Electromagn. Compat.*, vol. EMC–21, pp. 209–220, Aug. 1979.

[Miller and Thomas, 1970] J. H. Miller and J. B. Thomas, "The Detection of Signals in Impulsive Noise Modeled as a Mixture Process," *IEEE Trans. Commun.*, vol. COM–24, no. 5, pp. 559–563, May 1976.

[Miller and Thomas, 1972] J. H. Miller and J. B. Thomas, "Detector for Discrete–Time Signals in Non–Gaussian Noise," *IEEE Trans. Information Theory*, vol. IT–18, no. 2, pp. 241–250, March 1972.

[Milstein and Das, 1980] L. B. Milstein and P. K. Das, "An Analysis of a Real–Time Transform Domain Filtering Digital Communication System – Part I: Narrow–Band Interference Rejection," *IEEE Trans. Commun.*, vol. COM–28, no. 6, pp. 816–824, June 1980.

[Milstein and Das, 1983] L. B. Milstein and P. K. Das, "An Analysis of a Real–Time Transform Domain Filtering Digital Communication System – Part II: Wide–Band Interference Rejection," *IEEE Trans. Commun.*, vol. COM–31, no. 1, pp. 21–27, Jan. 1983.

[Milstein, 1988] L. B. Milstein, "Interference Rejection Techniques in Spread Spectrum Communications," *Proc. IEEE*, vol. 76, no. 6, pp. 657–671, June 1988.

[Mitra and Sherwood, 1973] S. K. Mitra and R. J. Sherwood, "Digital Ladder Networks," *IEEE Trans. Audio Electroacoust.*, vol. AU–21, no. 1, pp. 30–36, Feb. 1973.

[Morf *et al.*, 1976] M. Morf, T. Kailath, and L. Ljung, "Fast Algorithms for Recursive Identification," *Proc. 1976 Conf. Decision Contr.*, Clearwater, FL, pp. 916–921, Dec. 1976.

[Mostafa *et al.*, 1983] A. B. Mostafa, M. Abdel–Dader, and A. El–Osmany, "Improvements of Antijam Performance of Spread–Spectrum Systems," *IEEE Trans. Commun.*, vol. COM–31, no. 6, pp. 803–808, June 1983.

[Nehorai, 1985] A. Nehorai, "A Minimal Parameter Adaptive Notch Filter with Constrained Poles and Zeros," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–33, no. 4, pp. 983–996, Aug. 1985.

[Nehorai and Malah, 1980] A. Nehorai and D. Malah, "On the Stability and Performance of the Adaptive Line Enhancer," *Proc. ICASSP'78*, Denver, CO, pp. 478–481, April 1980.

[Ng, 1987] T. S. Ng, "Some Aspects of an Adaptive Digital Notch Filter with Constrained Poles and Zeros," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–35, no. 2, pp. 158–161, Feb. 1987.

[Nikias and Raghuver, 1987] C. L. Nikias and M. R. Raghuver, "Bispectrum Estimation: A Digital Signal Processing Framework," *Proc. IEEE*, vol. 75, pp. 869–892, July 1987.

[Nikias and Pan, 1988] C. L. Nikias and R. Pan, "Time Delay Estimation in Unknown Gaussian Spatially Correlated Noise, *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–36, pp. 1706–1714, Nov. 1988.

[Nikias and Liu, 1990] C. L. Nikias and F. Liu, "Bicepstrum Computation Based on Second– and Third–Order Statistics with Applications," *Proc. ICASSP'90*, Albuquerque, New Mexico, pp. 2381–2386, April 1990.

[Nikias and Mendel, 1993] C. L. Nikias and J. M. Mendel, "Signal Processing with Higher–Order Spectra," *IEEE Signal Processing Magazine* vol. 10, no. 3, pp. 10–37, July, 1993.

[Nikias and Petropulu, 1993] C. L. Nikias and A. P. Petropulu, *Higher–Order Spectra Analysis: A Nonlinear Signal Processing Framework,* Prentice–Hall Inc., Oppenheim Series of Signal Processing, 1993.

[Nishimura *et al.*, 1989] S. Nishimura, J. K. Kim, and K. Hirano, "Mean–Squared Error Analysis of an Adaptive Notch Filter," *IEEE Intern. Symp. Circuits Syst.,* Poretland, OR, pp. 732–735, May 1989.

[Noll, 1975] P. Noll, "A Comparative Study of Various Schemes for Speech Encoding," *Bell Syst. Tech. J.,* vol. 54, no. 9, pp. 1597–1614, Nov. 1975.

[North *et al.*, 1992] R. C. North *et al.*, "Comparison of Adaptive Lattice Filters to LMS Transversal Filters for Sinusoidal Cancellation," *Proc. ICASSP'92*, San Francisco, CA, pp. IV-33–IV-36, March 1992.

[Oppenheim and Schafer, 1975] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing,* Prentice–Hall Inc., Englewood Cliffs, New Jersey, 1975.

[Papoulis, 1984] A. Papoulis, "Probability, Random Variables, and Stochastic Processes," second ed., McGraw–Hill Publishing Company, New York, New York, 1984.

[Parkin, 1969] R. E. Parkin, "Triple Distributed Notch Filters and Some Active Filter Applications," *IEEE Trans. Circuit Theory,* vol. CT–16, pp. 400–404, Aug. 1969.

[Parikh and Ahmed, 1980] D. D. Parikh and N. Ahmed, "Sequential Regression Considerations of Adaptive Notch Filters," *IEEE Trans. Acoust. Speech Signal Process.,* vol. ASSP–28, no. 3, pp. 313–317, June 1980.

[Petraglia *et al.*, 1990a] M. R. Petraglia, S. K. Mitra, and J. Szczpak, "Adaptive Sinusoid Detection Using IIR Notch FIlters and Multirate Techniques," *IEEE Intern. Symp. Circuits Syst.,* New Orleans, LA, pp. 271–274, May 1990.

[Petraglia *et al.*, 1990b] M. R. Petraglia, J. J. Shynk, and S. K. Mitra, "Stability Bounds for an Adaptive IIR Notch Filter," *IEEE Intern. Symp. Circuits Syst.,* New Orleans, LA, pp. 1963–1966, May 1990.

[Pitas and Venetsanopoulos, 1990] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications,* Kluwer Academic Publishers, Boston, MA, 1990.

[Plackett, 1950] R. L. Plackett, "Some Theorems in Least Squares," *Biometrika,* vol. 37, p. 149, 1950.

[Poor, 1992] H. V. Poor, "Signal Processing for Wideband Communications," *IEEE Inform. Theory Society Newsletter,* vol. 42, no. 2, pp. 1–10, June 1992.

[Proakis, 1989] J. G. Proakis, *Digital Communications,* second ed., McGraw–Hill Publishing Company, New York, New York, 1989.

[Rabiner and Schafer, 1978] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals,* Prentice–Hall, Inc., Englewood Cliffs, New Jersey, 1978.

[Rife and Boorstyn, 1974] D. C. Rife and R. R. Boorstyn, "Single–Tone Parameter Estimation from Discrete–Time Observations," *IEEE Trans. Inform. Theory,* vol. IT–20, no. 5, pp. 591–598, Sept. 1974.

[Robbins and Monro, 1951] H. Robbins and S. Monro, "A Stochastic Approximation method," *Ann. Math. Statis.,* vol. 22, pp. 400–407, 1951.

[Rosenberger and Thomas, 1971] J. Rosenberger and E. Thomas, "Performance of an Adaptive Echo Canceller Operating in a Noisy, Linear, Time–Invariant Environment," *Bell Syst. Tech. J.,* vol. 50, pp. 785–813, March 1971.

[Roy and Lowenschuss, 1970] R. Roy and O. Lowenschuss, "Design of MTI Detection Filters with Nonuniform Interpulse Periods," *IEEE Trans. Circuit Theory,* vol. CT–17, no. 4, pp. 604–612, Nov. 1970.

[Rude and Griffiths, 1989] M. J. Rude and L. J. Griffiths, "Incorporation of Linear Constraints into the Constant Modulus Algorithm," *Proc. ICASSP'89,* Glasgow, Scotland, pp. 968–971, May 1989.

[Rude and Griffiths, 1990] M. J. Rude and L. J. Griffiths, "An Untrained, Fractionally–Spaced Equalizer for Co–Channel Interference Environments," *Proc. 24th Asilomar Conference on Signal, Systems, & Computer,* Pacific Grove, CA, pp. 468–472, Nov. 1990.

[Satorius *et al.*, 1979] E. H. Satorius, J. D. Smith, and P. M. Reeves, "Adaptive Noise Cancelling of a Sinusoidal Interference Using a Lattice Structure," *Proc. ICASSP'79,* Washington D. C., pp. 929–932, April 1979.

[Satorius *et al.* 1988] E. H. Satorius *et al.*, "Suppression of Narrowband Interference via Single Channel Adaptive Preprocessing," *Proc. 22nd Asilomar Conference on Signal, Systems, & Computer*, Pacific Grove, CA, pp. 270–273, Oct. 1988.

[Scholtz, 1977] R. A. Scholtz, "The Spread Spectrum Concept," *IEEE Trans. Commun.*, vol. COM–25, no. 8, pp. 748–755, Aug. 1977.

[Shao and Nikias, 1993] M. Shao and C. L. Nikias, "Signal Processing with Fractional Lower Order Moments: Stable Processes and Their Applications," *Proc. IEEE*, vol. 81, no. 7, pp. 986–1010, July 1993.

[Shao 1993] M. Shao, *Signal Processing Based on Stable Distributions*, Ph.D thesis, University of Southern California, Los Angeles, CA, 1993.

[Shao and Nikias, 1995] M. Shao and C. L. Nikias, *Signal Processing with Alpha–Stable Distributions and Applications*, John Wiley & Sons, 1995.

[Shensa, 1980] M. J. Shensa, "Non–Wiener Solution of the Adaptive Noise Canceller with a Noisy Reference," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–28, pp. 468–473, Aug. 1980.

[Shin and Mendel, 1992] D. C. Shin and J. M. Mendel, "Assessment of Cumulant–Based Approaches to Harmonic Retrieval," *Proc. ICASSP'92*, San Francisco, CA, pp. V-205–V-208, March 1992.

[Shin and Nikias, 1993] D. C. Shin and C. L. Nikias, "Adaptive Noise Canceler for Narrowband/Wideband Interferences Using Higher–Order Statistics," *Proc. ICASSP'93*, Minneapolis, MN, pp. III-364–III-367, April 1993.

[Shin and Nikias, 1994] D. C. Shin and C. L. Nikias, "Adaptive Interference Canceler for Narrowband and Wideband Interferences Using Higher–Order Statistics," *IEEE Trans. Signal Process.*, vol. SP–42, Oct. 1994.

[Shin and Nikias, 1994] D. C. Shin and C. L. Nikias, "Nonlinear Adaptive Interference Mitigation in Generalized Gaussian Channels," *Seventh SP Workshop on Statistical Signal & Array Process.*, Quebec City, Canada, June 1994.

[Simaan, 1985] M. Simaan, *Advances in Geophysical Data Processing*, vol. 2, Jai Press Inc., Greenwich, Connecticut, 1985.

[Sklar,1988] B. Sklar, *Digital Communications*, Prentice–Hall Inc., Englewood Cliffs, New Jersey, 1988.

[Soderstrand *et al.*, 1991] M. A. Soderstrand, H. H. Loomis, and K. V. Rangarao, "Elimination of Narrow–Band Interference in BPSK–Modulated Signal Reception," *IEEE Intern. Symp. Circuits Syst.*, Singapore, pp. 2798—2801, June 1991.

[Soderstrom *et al.*, 1978] T. Soderstrom, L. Ljung, and I. Gustavsson, "A Theoretical Analysis of Recursive Identification Methods," *Automatica*, vol. 14, pp. 231–244, 1978.

[Soldan and Ahmed, 1978] D. Soldan and N. Ahmed, "On a Sequential Regression Predictor," *Electron. Lett.*, vol. 14, no. 4, pp. 118–119, Feb. 1978.

[Solo, 1979] V. Solo, "The Convergence of AML," *IEEE Trans. Automat. Contr.*, vol. AC–24, no. 6, pp. 958–962, Dec. 1979.

[Sondhi, 1967] M. Sondhi, "An Adaptive Echo Canceller," *Bell Syst. Tech. J.*, vol. 46, pp. 497–511, March 1967.

[Sorenson, 1970] H. W. Sorenson, "Least–Squares Estimation: from Gauss to Kalman," *IEEE Spectrum*, vol. 7, pp. 63–68, July 1970.

[Sorenson and Alspach, 1971] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian Estimation Using Gaussian Sums," *Automatica*, vol. 7, pp. 465–479, July 1971.

[Spaulding and Middleton, 1977] A. D. Spaulding and D. Middleton, "Optimum Reception in an Impulsive Interference Environment – Part I: Coherent Detection," *IEEE Trans. Commun.*, vol. COM–25, pp. 910–923, Sept. 1977.

[Stein, 1981] S. Stein, "Algorithms for Ambiguity Function Processing," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–29, no. 3, pp. 588–599, June 1981.

[Stoica and Nehorai, 1988] P. Stoica and A. Nehorai, "Performance Analysis of Adaptive Notch Filter with Constrained Poles and Zeros," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–36, no. 6, pp. 911–919, June 1988.

[Strandberg *et al.*, 1992] R. H. Strandberg, M. A. Soderstrand, and H. H. Loomis, "Elimination of Narrow–Band Interference Using Adaptive Sampling Rate Notch Filters," *Proc. 26th Asilomar Conference on Signal, Systems, & Computer,* Pacific Grove, CA, pp. 861–865, Oct. 1992.

[Swami and Mendel, 1991] A. Swami and J. M. Mendel, "Cumulant–based Approach to the Harmonic Retrieval and Related Problems," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–39, pp. 1099–1109, May 1991.

[Swaminathan and Tugnait, 1993] R. Swaminathan and J. K. Tugnait, "On Improving the Convergence of Constant Modulus Algorithm Adaptive Filters," *Proc. ICASSP'93*, Minneapolis, MN, pp. III-340–III-343, April 1993.

[Tow, 1978] J. Tow, "Some Results on Two–Section Generalized FLF Active Filters," *IEEE Trans. Circuits Syst.*, vol. CAS–25, no. 4, pp. 181–184, April 1978.

[Treichler and Agee, 1983] J. R. Treichler and B. G. Agee, "A New Approach to Multipath Correction of Constant Modulus Signals," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–31, no. 2, pp. 459–472, April 1983.

[Treichler and Larimore, 1985] J. R. Treichler and M. G. Larimore, "The Tone Capture Properties of CMA–Based Interference Suppressors," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–33, no. 4, pp. 946–958, Aug. 1985.

[Treichler *et al.*, 1991] J. R. Treichler, V. Wolff, and C. R. Johnson, Jr., "Observed Misconvergence in the Constant Modulus Adaptive Algorithm," *Proc. 25th Asilomar Conference on Signal, Systems, & Computer*, Pacific Grove, CA, pp. 663–667, Nov. 1991.

[Trunk, 1970] G. V. Trunk, "Small– and Large–Sample Behavior of Two Detectors Against Envelope–Detected Sea Clutter," *IEEE Trans. Information Theory*, vol. IT–16, no. 1, pp. 95–99, Jan. 1970.

[Tsakalides and Nikias, 1993] P. Tsakalides and C. L. Nikias, "A New Criterion for Blind Deconvolution of Colored Input Signals," *Proc. 27th Asilomar Conference on Signal, Systems, & Computer,* Pacific Grove, CA, Nov. 1993.

[Van Trees, 1968] H. L. Van Trees, *Detection, Estimation, and Modulation Theory,* John Wiley and Sons, Inc., New York, 1968.

[Vastola, 1984] K. S. Vastola, "Threshold Detection in Narrow–Band Non–Gaussian Noise," *IEEE Trans. Commun.,* vol. COM–32, no. 2, pp. 134–139, Feb. 1984.

[Vijayan and Poor, 1990] R. Vijayan and H. V. Poor, "Nonlinear Techniques for Interference Suppression in Spread–Spectrum Systems," *IEEE Trans. Commun.,* vol. COM–38, no. 7, pp. 1060–1065, July 1990.

[Walach and Widrow, 1984] E. Walach and B. Widrow, "The Least Mean Fourth (LMF) Adaptive Algorithm and Its Family," *IEEE Trans. Inform. Theory,* vol. IT–30, no. 2, pp. 275–283, March 1984.

[Watt and Maxwell, 1957] A. D. Watt and E. L. Maxwell, "Characteristics of Atmospheric Noise from 1 to 100 Kc," *Proc. IRE,* vol. 45, pp. 787–794, June 1957.

[Weaver, 1962] C. S. Weaver, "Adaptive Communication Filtering," *IRE Trans. Inform. Theory,* vol. IT–8, pp. S169–S178, Sept. 1962.

[Weaver *et al.*, 1968] C. S. Weaver *et al.*, "Digital Filtering with Applications to Electrocardiogram Processing," *IEEE Trans. Audio Electroacoust.,* vol. AU–16, no. 3, pp. 350–391, Sept. 1968.

[Widrow *et al.*, 1967] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive Antenna System," *Proc. IEEE,* vol. 55, pp. 2143–2159, Dec. 1967.

[Widrow *et al.*, 1975] B. Widrow *et al.*, "Adaptive Noise Cancelling: Principle and Applications," *Proc. IEEE,* vol. 63, pp. 1692–1716, Dec. 1975.

[Widrow *et al.*, 1976] B. Widrow *et al.*, "Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter," *Proc. IEEE*, vol. 64, pp. 1151–1162, Aug. 1976.

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

[Wolff *et al.*, 1988] V. Wolff, R. Gooch, and J. Treichler, "Specification and Development of an Equalizer–Demodulator for Wideband Digital Microwave Radio Signals," *Proc. IEEE MILCOM'88 Conference*, San Diego, CA, pp. 461–467, Oct. 1988.

[Xue and Liu, 1986] P. Xue and B. Liu, "Adaptive Equalizer Using Finite–Bit Power–of–Two Quantizer," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–34, no. 6, pp. 1603–1611, Dec. 1986.

[Yan, 1984] G. Yan, "New Digital Notch Filter Structures with Low Coefficient Sensitivity," *IEEE Trans. Circuits Syst.*, vol. CAS–31, no. 9, pp. 825–828, Sept. 1984.

[Zeidler *et al.*, 1978] J. R. Zeidler *et al.*, "Adaptive Enhancement of Multiple Sinusoids in Uncorrelated Noise," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP–26, no. 3, pp. 240–254, June 1978.

[Zeidler *et al.*, 1991] J. R. Zeidler *et al.*, "Frequency Tracking Performance of Adaptive Lattice Filters," *Proc. 25th Asilomar Conference on Signal, Systems, & Computer*, Pacific Grove, CA, pp. 643–649, Nov. 1991.