This book is organised in 15 chapters.

Chapter 1 begins with an introduction to signal processing, and provides a brief review of signal processing methodologies and applications. The basic operations of sampling and quantisation are reviewed in this chapter.

Chapter 2 provides an introduction to noise and distortion. Several different types of noise, including thermal noise, shot noise, acoustic noise, electromagnetic noise and channel distortions, are considered. The chapter concludes with an introduction to the modelling of noise processes.

Chapter 3 provides an introduction to the theory and applications of probability models and stochastic signal processing. The chapter begins with an introduction to random signals, stochastic processes, probabilistic models and statistical measures. The concepts of stationary, non-stationary and ergodic processes are introduced in this chapter, and some important classes of random processes, such as Gaussian, mixture Gaussian, Markov chains and Poisson processes, are considered. The effects of transformation of a signal on its statistical distribution are considered.

Chapter 4 is on Bayesian estimation and classification. In this chapter the estimation problem is formulated within the general framework of Bayesian inference. The chapter includes Bayesian theory, classical estimators, the estimate–maximise method, the Cramér–Rao bound on the minimum−variance estimate, Bayesian classification, and the modelling of the space of a random signal. This chapter provides a number of examples on Bayesian estimation of signals observed in noise.

Chapter 5 considers hidden Markov models (HMMs) for non-stationary signals. The chapter begins with an introduction to the modelling of non-stationary signals and then concentrates on the theory and applications of hidden Markov models. The hidden Markov model is introduced as a Bayesian model, and methods of training HMMs and using them for decoding and classification are considered. The chapter also includes the application of HMMs in noise reduction.

Chapter 6 considers Wiener Filters. The least square error filter is formulated first through minimisation of the expectation of the squared error function over the space of the error signal. Then a block-signal formulation of Wiener filters and a vector space interpretation of Wiener filters are considered. The frequency response of the Wiener filter is derived through minimisation of mean square error in the frequency domain. Some applications of the Wiener filter are considered, and a case study of the Wiener filter for removal of additive noise provides useful insight into the operation of the filter.

Chapter 7 considers adaptive filters. The chapter begins with the state-space equation for Kalman filters. The optimal filter coefficients are derived using the principle of orthogonality of the innovation signal. The recursive least squared (RLS) filter, which is an exact sample-adaptive implementation of the Wiener filter, is derived in this chapter. Then the steepest–descent search method for the optimal filter is introduced. The chapter concludes with a study of the LMS adaptive filters.

Chapter 8 considers linear prediction and sub-band linear prediction models. Forward prediction, backward prediction and lattice predictors are studied. This chapter introduces a modified predictor for the modelling of the short–term and the pitch period correlation structures. A maximum a posteriori (MAP) estimate of a predictor model that includes the prior probability density function of the predictor is introduced. This chapter concludes with the application of linear prediction in signal restoration.

Chapter 9 considers frequency analysis and power spectrum estimation. The chapter begins with an introduction to the Fourier transform, and the role of the power spectrum in identification of patterns and structures in a signal process. The chapter considers non–parametric spectral estimation.

Chapter 10 considers interpolation of a sequence of unknown samples. This chapter begins with a study of the ideal interpolation of a band-limited signal, a simple model for the effects of a number of missing samples, and the factors that affect interpolation. Interpolators are divided into two

categories: polynomial and statistical interpolators. A general form of polynomial interpolation as well as its special forms (Lagrange, Newton, Hermite and cubic spline interpolators) are considered. Statistical interpolators in this chapter include maximum a posteriori interpolation, least squared error interpolation based on an autoregressive model, time–frequency interpolation, and interpolation through search of an adaptive codebook for the best signal.

Chapter 11 considers spectral subtraction. A general form of spectral subtraction is formulated and the processing distortions that result form spectral subtraction are considered. The effects of processing-distortions on the distribution of a signal are illustrated. The chapter considers methods for removal of the distortions and also non-linear methods of spectral subtraction. This chapter concludes with an implementation of spectral subtraction for signal restoration.

Chapters 12 and 13 cover the modelling, detection and removal of impulsive noise and transient noise pulses. In Chapter 12, impulsive noise is modelled as a binary–state non-stationary process and several stochastic models for impulsive noise are considered. For removal of impulsive noise, median filters and a method based on a linear prediction model of the signal

Chapter 11 considers spectral subtraction. A general form of spectral subtraction is formulated and the processing distortions that result form spectral subtraction are considered. The effects of processing-distortions on the distribution of a signal are illustrated. The chapter considers methods for removal of the distortions and also non-linear methods of spectral subtraction. This chapter concludes with an implementation of spectral subtraction for signal restoration.

Chapters 12 and 13 cover the modelling, detection and removal of impulsive noise and transient noise pulses. In Chapter 12, impulsive noise is modelled as a binary–state non-stationary process and several stochastic models for impulsive noise are considered. For removal of impulsive noise, median filters and a method based on a linear prediction model of the signal process are considered. The materials in Chapter 13 closely follow Chapter 12. In Chapter 13, a template-based method, an HMM-based method and an AR model-based method for removal of transient noise are considered.

Chapter 14 covers echo cancellation. The chapter begins with an introduction to telephone line echoes, and considers line echo suppression and adaptive line echo cancellation. Then the problem of acoustic echoes and acoustic coupling between loudspeaker and microphone systems are considered. The chapter concludes with a study of a sub-band echo cancellation system

Chapter 15 is on blind deconvolution and channel equalisation. This chapter begins with an introduction to channel distortion models and the ideal channel equaliser. Then the Wiener equaliser, blind equalisation using the channel input power spectrum, blind deconvolution based on linear predictive models, Bayesian channel equalisation, and blind equalisation for digital communication channels are considered. The chapter concludes with equalisation of maximum phase channels using higher-order statistics.

FREQUENTLY USED SYMBOLS AND ABBREVIATIONS

AWGN	Additive white Gaussian noise
ARMA	Autoregressive moving average process
AR	Autoregressive process
$\boldsymbol{A}$	Matrix of predictor coefficients
a_k	Linear predictor coefficients
$\boldsymbol{a}$	Linear predictor coefficients vector
a_{ij}	Probability of transition from state i to state j in a Markov model
$\alpha_i(t)$	Forward probability in an HMM
bps	Bits per second
$b(m)$	Backward prediction error
$b(m)$	Binary state signal
$\beta_i(t)$	Backward probability in an HMM
$c_{xx}(m)$	Covariance of signal $x(m)$
$c_{XX}(k_1, k_2, \cdots, k_N)$	k^{th} order cumulant of $x(m)$
$C_{XX}(\omega_1, \omega_2, \cdots, \omega_{k-1})$	k^{th} order cumulant spectra of $x(m)$
$\boldsymbol{D}$	Diagonal matrix
$e(m)$	Estimation error
$\mathcal{E}[x]$	Expectation of x
f	Frequency variable
$f_X(x)$	Probability density function for process X
$f_{X,Y}(x, y)$	Joint probability density function of X and Y
$f_{X\|Y}(x\|y)$	Probability density function of X conditioned on Y
$f_{X;\Theta}(x;\theta)$	Probability density function of X with θ as a parameter
$f_{X\|S,\mathcal{M}}(x\|s,\mathcal{M})$	Probability density function of X given a state sequence s of an HMM $\mathcal{M}$ of the process X

$\boldsymbol{\Phi}(m,m-1)$	State transition matrix in Kalman filter
$\boldsymbol{h}$	Filter coefficient vector, Channel response
$\boldsymbol{h}_{\text{max}}$	Maximum–phase channel response
$\boldsymbol{h}_{\text{min}}$	Minimum–phase channel response
$\boldsymbol{h}^{\text{inv}}$	Inverse channel response
$H(f)$	Channel frequency response

$H^{\text{inv}}(f)$	Inverse channel frequency response
$\boldsymbol{H}$	Observation matrix, Distortion matrix
$\mathbf{I}$	Identity matrix
$\boldsymbol{J}$	Fisher's information matrix
$\lvert \boldsymbol{J} \rvert$	Jacobian of a transformation
$\boldsymbol{K}(m)$	Kalman gain matrix
LSE	Least square error
LSAR	Least square AR interpolation
λ	Eigenvalue
$\boldsymbol{\Lambda}$	Diagonal matrix of eigenvalues
MAP	Maximum a posterior estimate
MA	Moving average process
ML	Maximum likelihood estimate
MMSE	Minimum mean squared error estimate
m	Discrete time index
m_k	k^{th} order moment
$\mathcal{M}$	A model, e.g. an HMM
μ	Adaptation convergence factor
μ_x	Expected mean of vector x
$n(m)$	Noise

$n(m)$	Noise
$\boldsymbol{n}(m)$	A noise vector of N samples
$n_i(m)$	Impulsive noise
$N(f)$	Noise spectrum
$N^*(f)$	Complex conjugate of $N(f)$
$\overline{N(f)}$	Time-averaged noise spectrum
$\mathcal{N}(\boldsymbol{x}, \boldsymbol{\mu}_{xx}, \boldsymbol{\Sigma}_{xx})$	A Gaussian pdf with mean vector $\boldsymbol{\mu}_{xx}$ and covariance matrix $\boldsymbol{\Sigma}_{xx}$
$O(\cdot)$	In the order of $(\cdot)$
P	Filter order (length)
pdf	Probability density function
pmf	Probability mass function
$P_X(x_i)$	Probability mass function of x_i
$P_{X,Y}(x_i, y_j)$	Joint probability mass function of x_i and y_j
$P_{X\|Y}(x_i \| y_j)$	Conditional probability mass function of x_i given y_j
$P_{NN}(f)$	Power spectrum of noise $n(m)$
$P_{XX}(f)$	Power spectrum of the signal $x(m)$

$P_{XY}(f)$	Cross–power spectrum of signals $x(m)$ and $y(m)$
$\boldsymbol{\theta}$	Parameter vector
$\hat{\boldsymbol{\theta}}$	Estimate of the parameter vector $\boldsymbol{\theta}$
r_k	Reflection coefficients

$r_{xx}(m)$	Autocorrelation function
$\boldsymbol{r}_{xx}(m)$	Autocorrelation vector
$\boldsymbol{R}_{xx}$	Autocorrelation matrix of signal $\boldsymbol{x}(m)$
$\boldsymbol{R}_{xy}$	Cross–correlation matrix
s	State sequence
s^{ML}	Maximum–likelihood state sequence
SNR	Signal-to-noise ratio
SINR	Signal-to-impulsive noise ratio
σ_n^2	Variance of noise $n(m)$
$\boldsymbol{\Sigma}_{nn}$	Covariance matrix of noise $\boldsymbol{n}(m)$
$\boldsymbol{\Sigma}_{xx}$	Covariance matrix of signal $\boldsymbol{x}(m)$
σ_x^2	Variance of signal $x(m)$
σ_n^2	Variance of noise $n(m)$
$x(m)$	Clean signal
$\hat{x}(m)$	Estimate of clean signal
$\boldsymbol{x}(m)$	Clean signal vector
$X(f)$	Frequency spectrum of signal $x(m)$
$X^*(f)$	Complex conjugate of $X(f)$
$\overline{X(f)}$	Time-averaged frequency spectrum of $x(m)$
$X(f,t)$	Time-frequency spectrum of $x(m)$
$\boldsymbol{X}$	Clean signal matrix
$\boldsymbol{X}^{\mathrm{H}}$	Hermitian transpose of $\boldsymbol{X}$
$y(m)$	Noisy signal
$\boldsymbol{y}(m)$	Noisy signal vector
$\hat{y}(m\|m-i)$	Prediction of $y(m)$ based on observations up to time m–i
$\boldsymbol{Y}$	Noisy signal matrix
$\boldsymbol{Y}^{\mathrm{H}}$	Hermitian transpose of $\boldsymbol{Y}$
Var	Variance
w_k	Wiener filter coefficients
$\boldsymbol{w}(m)$	Wiener filter coefficients vector
$W(f)$	Wiener filter frequency response
z	z-transform variable

INTRODUCTION

Signal processing is concerned with the modelling, detection, identification and utilisation of patterns and structures in a signal process. Applications of signal processing methods include audio hi-fi, digital TV and radio, cellular mobile phones, voice recognition, vision, radar, sonar, geophysical exploration, medical electronics, and in general any system that is concerned with the communication or processing of information. Signal processing theory plays a central role in the development of digital telecommunication and automation systems, and in efficient and optimal transmission, reception and decoding of information. Statistical signal processing theory provides the foundations for modelling the distribution of random signals and the environments in which the signals propagate. Statistical models are applied in signal processing, and in decision-making systems, for extracting information from a signal that may be noisy, distorted or incomplete. This chapter begins with a definition of signals, and a brief introduction to various signal processing methodologies. We consider several key applications of digital signal processing in adaptive noise reduction, channel equalisation, pattern classification/recognition, audio signal coding, signal detection, spatial processing for directional reception of signals, Dolby noise reduction and radar. The chapter concludes with an introduction to sampling and conversion of continuous-time signals to digital signals.

1.1 Signals and Information

A signal can be defined as the variation of a quantity by which information is conveyed regarding the state, the characteristics, the composition, the trajectory, the course of action or the intention of the signal source. *A signal is a means to convey information.* The information conveyed in a signal may be used by humans or machines for communication, forecasting, decision-making, control, exploration etc. Figure 1.1 illustrates an information source followed by a system for signalling the information, a communication channel for propagation of the signal from the transmitter to the receiver, and a signal processing unit at the receiver for extraction of the information from the signal. In general, there is a mapping operation that maps the information $I(t)$ to the signal $x(t)$ that carries the information, this mapping function may be denoted as $T[\cdot]$ and expressed as

$$x(t)=T[I(t)] \qquad (1.1)$$

information $I(t)$ to the signal $x(t)$ that carries the information, this mapping function may be denoted as $T[\cdot]$ and expressed as

$$x(t)=T[I(t)] \tag{1.1}$$

For example, in human speech communication, the voice-generating mechanism provides a means for the talker to map each word into a distinct acoustic speech signal that can propagate to the listener. To communicate a word w, the talker generates an acoustic signal realisation of the word; this acoustic signal x(t) may be contaminated by ambient noise and/or distorted by a communication channel, or impaired by the speaking abnormalities of the talker, and received as the noisy and distorted signal $y(t)$. In addition to conveying the spoken word, the acoustic speech signal has the capacity to convey information on the speaking characteristic, accent and the emotional state of the talker. The listener extracts these information by processing the signal $y(t)$.

In the past few decades, the theory and applications of digital signal processing have evolved to play a central role in the development of modern telecommunication and information technology systems.

Signal processing methods are central to efficient communication, and to the development of intelligent man/machine interfaces in such areas as

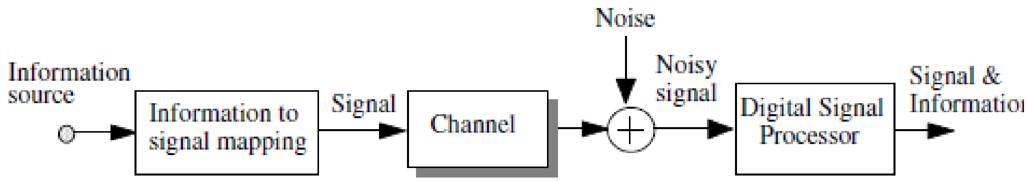


Figure 1.1 Illustration of a communication and signal processing system.

1.2 Signal Processing Methods

Signal processing methods have evolved in algorithmic complexity aiming for optimal utilisation of the information in order to achieve the best performance. In general the computational requirement of signal processing methods increases, often exponentially, with the algorithmic complexity. However, the implementation cost of advanced signal processing methods has been offset and made affordable by the consistent trend in recent years of a continuing increase in the performance, coupled with a simultaneous decrease in the cost, of signal processing hardware.

Depending on the method used, digital signal processing algorithms can be categorised into one or a combination of four broad categories. These are non–parametric signal processing, model-based signal processing, Bayesian statistical signal processing and neural networks. These methods are briefly described in the following.

1.2.1 Non–parametric Signal Processing

Non–parametric methods, as the name implies, do *not* utilise a parametric model of the signal generation or a model of the statistical distribution of the signal. The signal is processed as a waveform or a sequence of digits. Non–parametric methods are not specialised to any particular class of signals, they are broadly applicable methods that can be applied to any signal regardless of the characteristics or the source of the signal. The drawback of these methods is that they do not utilise the distinct characteristics of the signal process that may lead to substantial

1.2.2 Model-Based Signal Processing

Model-based signal processing methods utilise a parametric model of the signal generation process. The parametric model normally describes the predictable structures and the expected patterns in the signal process, and can be used to forecast the future values of a signal from its past trajectory. Model-based methods normally outperform non-parametric methods, since they utilise more information in the form of a model of the signal process. However, they can be sensitive to the deviations of a signal from the class of signals characterised by the model. The most widely used parametric model is the linear prediction model, described in Chapter 8. Linear prediction models have facilitated the development of advanced signal processing methods for a wide range of applications such as low-bit-rate speech coding in cellular mobile telephony, digital video coding, high-resolution spectral analysis, radar signal processing and speech recognition.

1.2.3 Bayesian Statistical Signal Processing

The fluctuations of a purely random signal, or the distribution of a class of random signals in the signal space, cannot be modelled by a predictive equation, but can be described in terms of the statistical average values, and modelled by a probability distribution function in a multidimensional signal space. For example, as described in Chapter 8, a linear prediction model driven by a random signal can model the acoustic realisation of a spoken word. However, the random input signal of the linear prediction model, or the variations in the characteristics of different acoustic realisations of the same word across the speaking population, can only be described in statistical terms and in terms of probability functions. Bayesian inference theory provides a generalised framework for statistical processing of random signals, and for formulating and solving estimation and decision-making problems. Chapter 4 describes the Bayesian inference methodology and the estimation of random processes observed in noise.

1.2.4 Neural Networks

Neural networks are combinations of relatively simple non-linear adaptive processing units, arranged to have a structural resemblance to the transmission and processing of signals in biological neurons. In a neural network several layers of parallel processing elements are interconnected with a hierarchically structured connection network. The connection weights are trained to perform a signal processing function such as prediction or classification. Neural networks are particularly useful in non-linear partitioning of a signal space, in feature extraction and pattern recognition, and in decision-making systems. In some hybrid pattern recognition systems neural networks are used to complement Bayesian inference methods. Since the main objective of this book is to provide a coherent presentation of the theory and applications of statistical signal processing, neural networks are not discussed in this book.

1.3 Applications of Digital Signal Processing

In recent years, the development and commercial availability of increasingly powerful and affordable digital computers has been accompanied by the development of advanced digital signal processing algorithms for a wide variety of applications such as noise reduction, telecommunication, radar, sonar, video and audio signal processing, pattern recognition, geophysics explorations, data forecasting, and the processing of large databases for the identification extraction and organisation of unknown underlying structures and patterns. Figure 1.2 shows a broad categorisation of some DSP applications. This section provides a review of several key applications of digital signal processing methods.

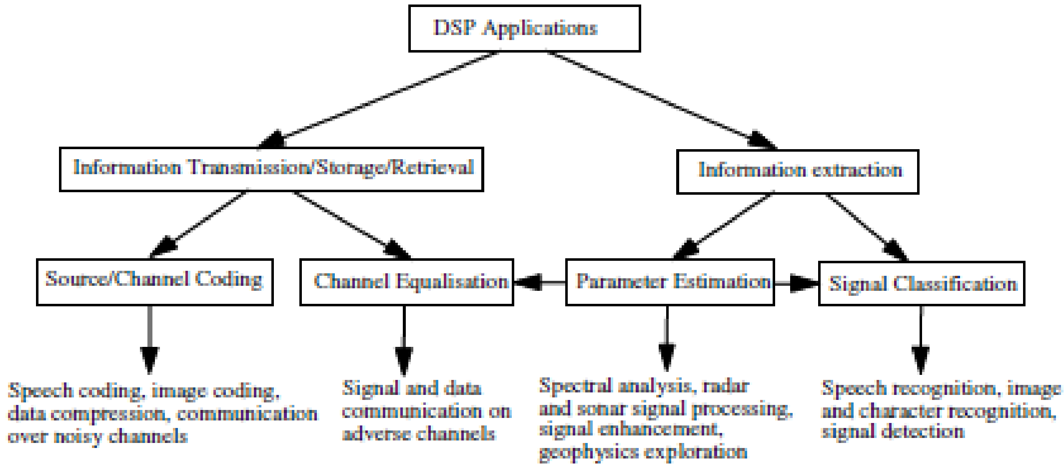


Figure 1.2 A classification of the applications of digital signal processing.

where $x(m)$ and $n(m)$ are the signal and the noise, and m is the discrete-time index. In some situations, for example when using a mobile telephone in a moving car, or when using a radio communication device in an aircraft cockpit, it may be possible to measure and estimate the instantaneous amplitude of the ambient noise using a directional microphone. The signal $x(m)$ may then be recovered by subtraction of an estimate of the noise from the noisy signal.

Figure 1.3 shows a two-input adaptive noise cancellation system for enhancement of noisy speech. In this system a directional microphone takes

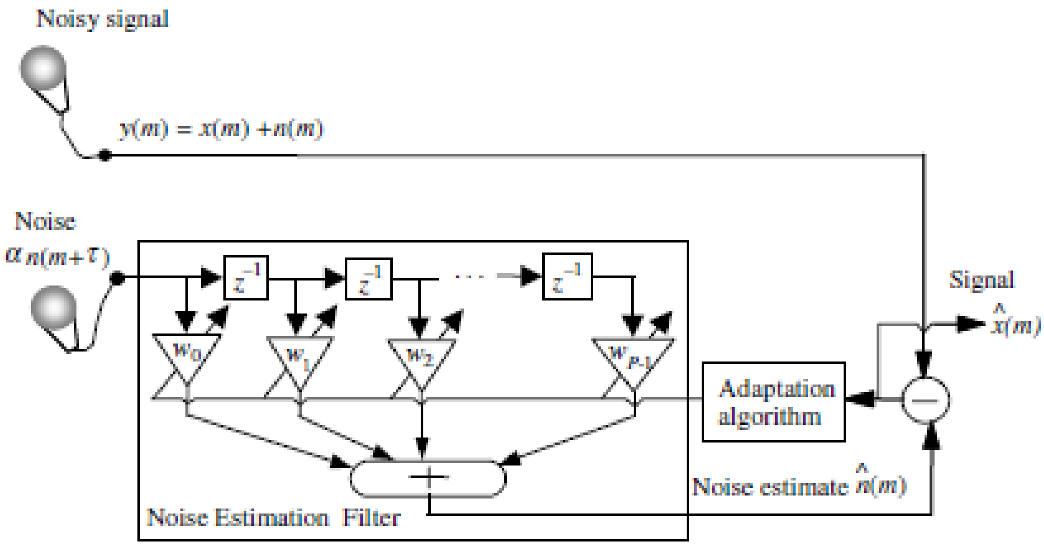


Figure 1.3 Configuration of a two-microphone adaptive noise canceller.

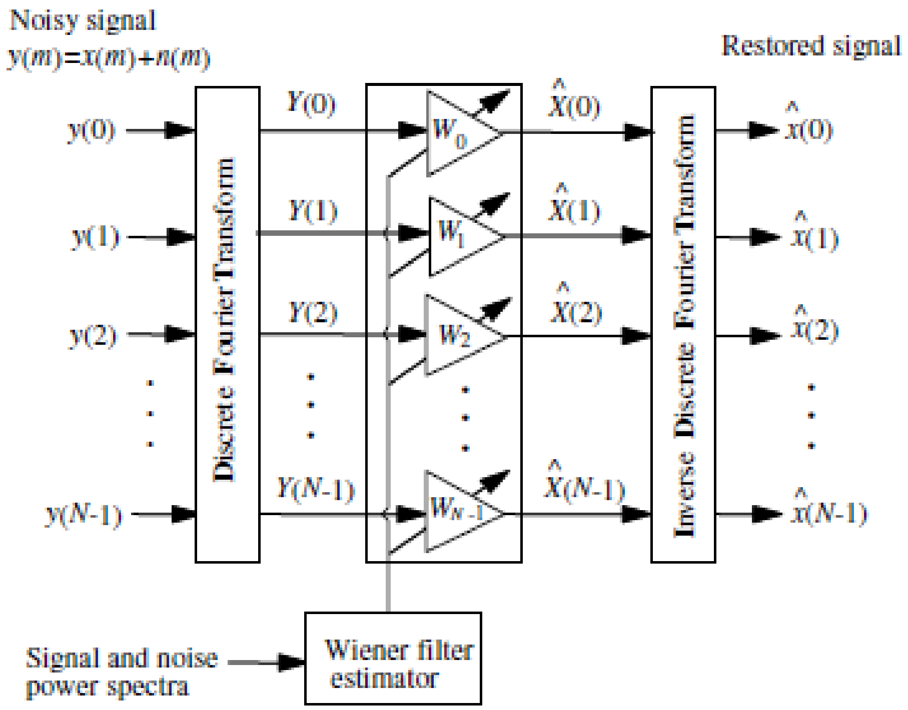


Figure 1.4 A frequency–domain Wiener filter for reducing additive noise.

1.3.3 Signal Classification and Pattern Recognition

Signal classification is used in detection, pattern recognition and decision-making systems. For example, a simple binary-state classifier can act as the detector of the presence, or the absence, of a known waveform in noise. In signal classification, the aim is to design a minimum-error system for *labelling* a signal with one of a number of likely classes of signal.

To design a classifier; a set of models are trained for the classes of signals that are of interest in the application. The simplest form that the models can assume is a bank, or code book, of waveforms, each representing the prototype for one class of signals. A more complete model for each class of signals takes the form of a probability distribution function. In the classification phase, a signal is labelled with the nearest or the most likely class. For example, in communication of a binary bit stream over a band-pass channel, the binary phase–shift keying (BPSK) scheme signals the bit "1" using the waveform $A_c \sin \omega_c t$ and the bit "0" using $-A_c \sin \omega_c t$. At the receiver, the decoder has the task of classifying and labelling the received noisy signal as a "1" or a "0". Figure 1.6 illustrates a correlation receiver for a BPSK signalling scheme. The receiver has two correlators, each programmed with one of the two symbols representing the binary

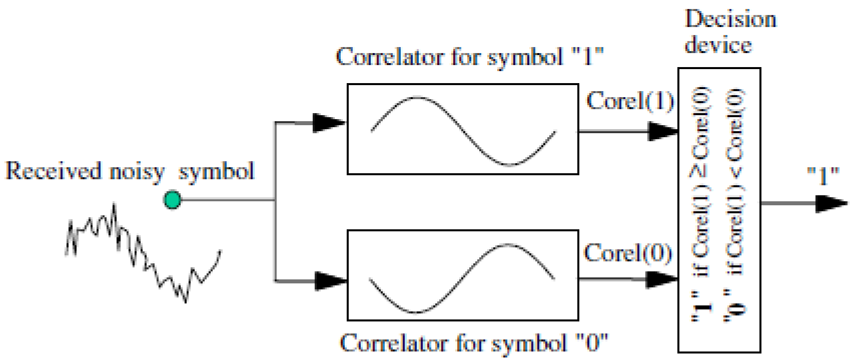


Figure 1.6 A block diagram illustration of the classifier in a binary phase-shift keying demodulation.

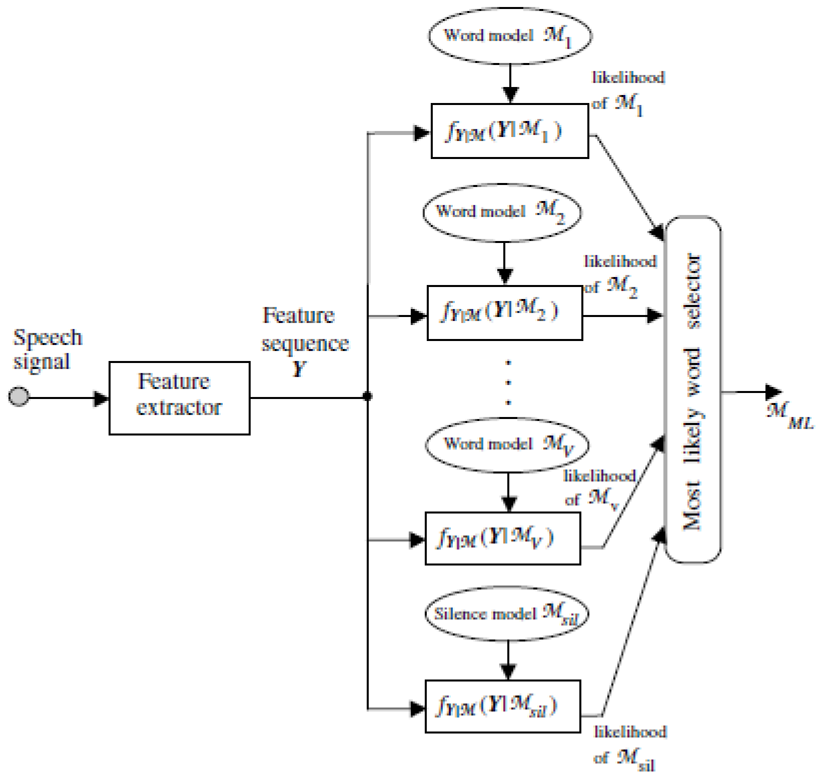


Figure 1.7 Configuration of speech recognition system, $f(Y|\mathcal{M}_i)$ is the likelihood of the model $\mathcal{M}_i$ given an observation sequence Y.

1.3.4 Linear Prediction Modelling of Speech

Linear predictive models are widely used in speech processing applications such as low–bit–rate speech coding in cellular telephony, speech enhancement and speech recognition. Speech is generated by inhaling air into the lungs, and then exhaling it through the vibrating glottis cords and the vocal tract. The random, noise-like, air flow from the lungs is spectrally shaped and amplified by the vibrations of the glottal cords and the resonance of the vocal tract. The effect of the vibrations of the glottal cords and the vocal tract is to introduce a measure of correlation and predictability on the random variations of the air from the lungs. Figure 1.8 illustrates a model for speech production. The source models the lung and emits a random excitation signal which is filtered, first by a pitch filter model of the glottal cords and then by a model of the vocal tract.

The main source of correlation in speech is the vocal tract modelled by a linear predictor. A linear predictor forecasts the amplitude of the signal at time m, $x(m)$, using a linear combination of P previous samples $[x(m-1),\cdots,x(m-P)]$ as

$$\hat{x}(m)=\sum_{k=1}^{P} a_k x(m-k) \tag{1.3}$$

where $\hat{x}(m)$ is the prediction of the signal $x(m)$, and the vector $\boldsymbol{a}^{\mathrm{T}}=[a_1,\ldots,a_P]$ is the coefficients vector of a predictor of order P. The

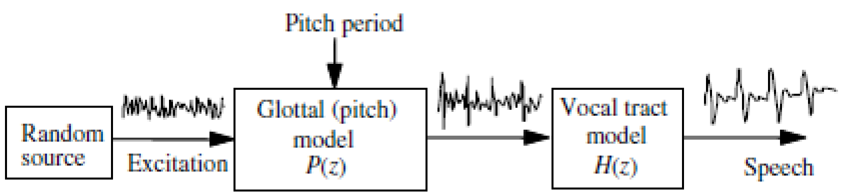


Figure 1.8 Linear predictive model of speech.

1.3.4 Linear Prediction Modelling of Speech

Linear predictive models are widely used in speech processing applications such as low–bit–rate speech coding in cellular telephony, speech enhancement and speech recognition. Speech is generated by inhaling air into the lungs, and then exhaling it through the vibrating glottis cords and the vocal tract. The random, noise-like, air flow from the lungs is spectrally shaped and amplified by the vibrations of the glottal cords and the resonance of the vocal tract. The effect of the vibrations of the glottal cords and the vocal tract is to introduce a measure of correlation and predictability on the random variations of the air from the lungs. Figure 1.8 illustrates a model for speech production. The source models the lung and emits a random excitation signal which is filtered, first by a pitch filter model of the glottal cords and then by a model of the vocal tract.

The main source of correlation in speech is the vocal tract modelled by a linear predictor. A linear predictor forecasts the amplitude of the signal at time m, $x(m)$, using a linear combination of P previous samples $[x(m-1),\cdots,x(m-P)]$ as

$$\hat{x}(m)=\sum_{k=1}^{P} a_k x(m-k) \tag{1.3}$$

where $\hat{x}(m)$ is the prediction of the signal $x(m)$, and the vector $\boldsymbol{a}^{\mathrm{T}}=[a_1,\ldots,a_P]$ is the coefficients vector of a predictor of order P. The

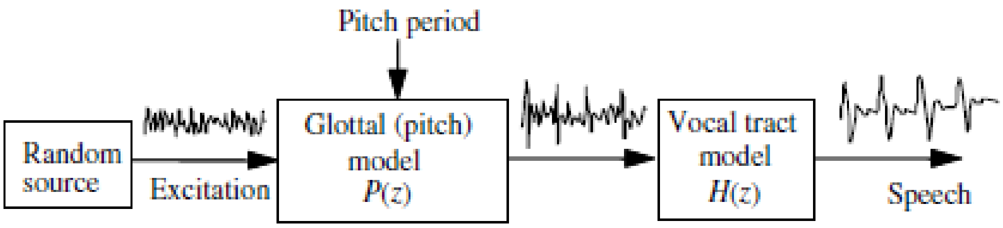


Figure 1.8 Linear predictive model of speech.

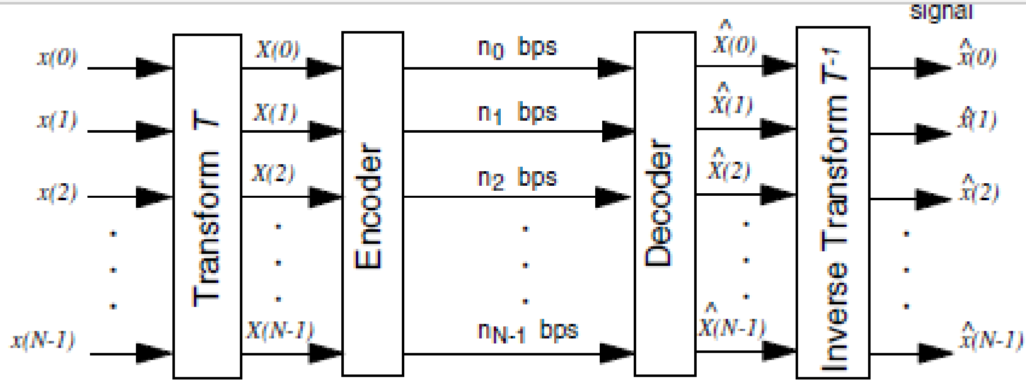


Figure 1.11 Illustration of a transform-based coder.

(b) A relatively low–amplitude frequency would be masked in the near vicinity of a large–amplitude frequency and can therefore be coarsely encoded without any audible degradation.
(c) The frequency samples are orthogonal and can be coded independently with different precisions.

The number of bits assigned to each frequency of a signal is a variable that reflects the contribution of that frequency to the reproduction of a perceptually high quality signal. In an adaptive coder, the allocation of bits to different frequencies is made to vary with the time variations of the power spectrum of the signal.

1.3.6 Detection of Signals in Noise

In the detection of signals in noise, the aim is to determine if the observation consists of noise alone, or if it contains a signal. The noisy observation $y(m)$ can be modelled as

$$y(m) = b(m)x(m) + n(m) \qquad (1.6)$$

where $x(m)$ is the signal to be detected, $n(m)$ is the noise and $b(m)$ is a binary-valued state indicator sequence such that $b(m)=1$ indicates the presence of the signal $x(m)$ and $b(m)=0$ indicates that the signal is absent. If the signal $x(m)$ has a known shape, then a correlator or a matched filter

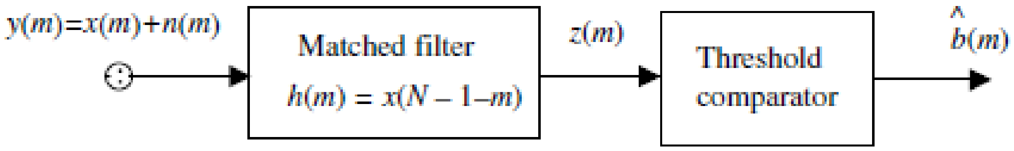


Figure 1.12 Configuration of a matched filter followed by a threshold comparator for detection of signals in noise.

can be used to detect the signal as shown in Figure 1.12. The impulse response $h(m)$ of the matched filter for detection of a signal $x(m)$ is the time-reversed version of $x(m)$ given by

$$h(m)=x(N-1-m) \qquad 0\le m\le N-1 \tag{1.7}$$

where N is the length of $x(m)$. The output of the matched filter is given by

$$z(m)=\sum_{m=0}^{N-1} h(m-k)y(m) \tag{1.8}$$

The matched filter output is compared with a threshold and a binary decision is made as

$$\hat{b}(m)=\begin{cases}1 & \text{if } z(m)\ge \text{threshold}\\ 0 & \text{otherwise}\end{cases} \tag{1.9}$$

where $\hat{b}(m)$ is an estimate of the binary state indicator sequence $b(m)$, and it may be erroneous in particular if the signal–to–noise ratio is low. Table1.1 lists four possible outcomes that together $b(m)$ and its estimate $\hat{b}(m)$ can assume. The choice of the threshold level affects the sensitivity of the

$\hat{b}(m)$	$b(m)$	Detector decision
0	0	Signal absent *Correct*
0	1	Signal absent (*Missed*)
1	0	Signal present (*False alarm*)
1	1	Signal present *Correct*

Table 1.1 Four possible outcomes in a signal detection problem.

1.4 Sampling and Analog–to–Digital Conversion

A digital signal is a sequence of real–valued or complex–valued numbers, representing the fluctuations of an information bearing quantity with time, space or some other variable. The *basic* elementary discrete-time signal is the unit-sample signal $\delta(m)$ defined as

$$\delta(m) = \begin{cases} 1 & m = 0 \\ 0 & m \neq 0 \end{cases} \tag{1.19}$$

where m is the discrete time index. A digital signal $x(m)$ can be expressed as the sum of a number of amplitude-scaled and time-shifted unit samples as

$$x(m) = \sum_{k=-\infty}^{\infty} x(k)\delta(m-k) \tag{1.20}$$

Figure 1.17 illustrates a discrete-time signal. Many random processes, such as speech, music, radar and sonar generate signals that are continuous in

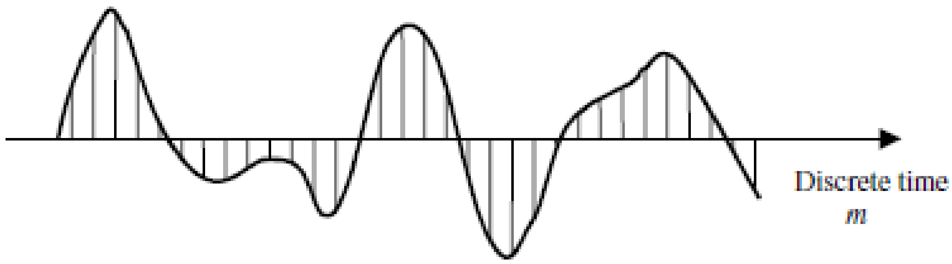


Figure 1.17 A discrete-time signal and its envelope of variation with time.

$$x(m) = \sum_{k=-\infty}^{\infty} x(k)\delta(m-k) \tag{1.20}$$

Figure 1.17 illustrates a discrete-time signal. Many random processes, such as speech, music, radar and sonar generate signals that are continuous in

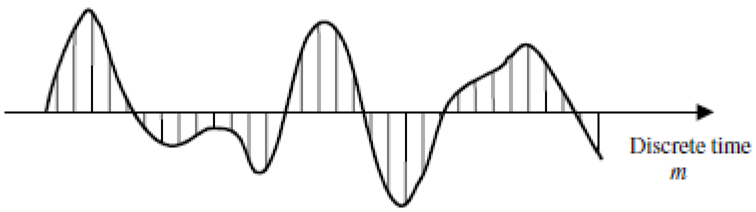


Figure 1.17 A discrete-time signal and its envelope of variation with time.

Figure 1.18 Configuration of a digital signal processing system.

1.4.1 Time-Domain Sampling and Reconstruction of Analog Signals

The conversion of an analog signal to a sequence of n-bit digits consists of two basic steps of sampling and quantisation. The sampling process, when performed with sufficiently high speed, can capture the fastest fluctuations of the signal, and can be a loss-less operation in that the analog signal can be recovered through interpolation of the sampled sequence as described in Chapter 10. The quantisation of each sample into an n-bit digit, involves some irrevocable error and possible loss of information. However, in practice the quantisation error can be made negligible by using an appropriately high number of bits as in a digital audio hi-fi. A sampled signal can be modelled as the product of a continuous-time signal $x(t)$ and a periodic impulse train $p(t)$ as

$$\begin{aligned} x_{\text{sampled}}(t) &= x(t)\,p(t) \\ &= \sum_{m=-\infty}^{\infty} x(t)\delta(t-mT_s) \end{aligned} \tag{1.21}$$

where T_s is the sampling interval and the sampling function $p(t)$ is defined as

$$p(t) = \sum_{m=-\infty}^{\infty} \delta(t-mT_s) \tag{1.22}$$

The spectrum $P(f)$ of the sampling function $p(t)$ is also a periodic impulse train given by

$$P(f) = \sum_{k=-\infty}^{\infty} \delta(f-kF_s) \tag{1.23}$$

where $F_s=1/T_s$ is the sampling frequency. Since multiplication of two time-domain signals is equivalent to the convolution of their frequency spectra we have

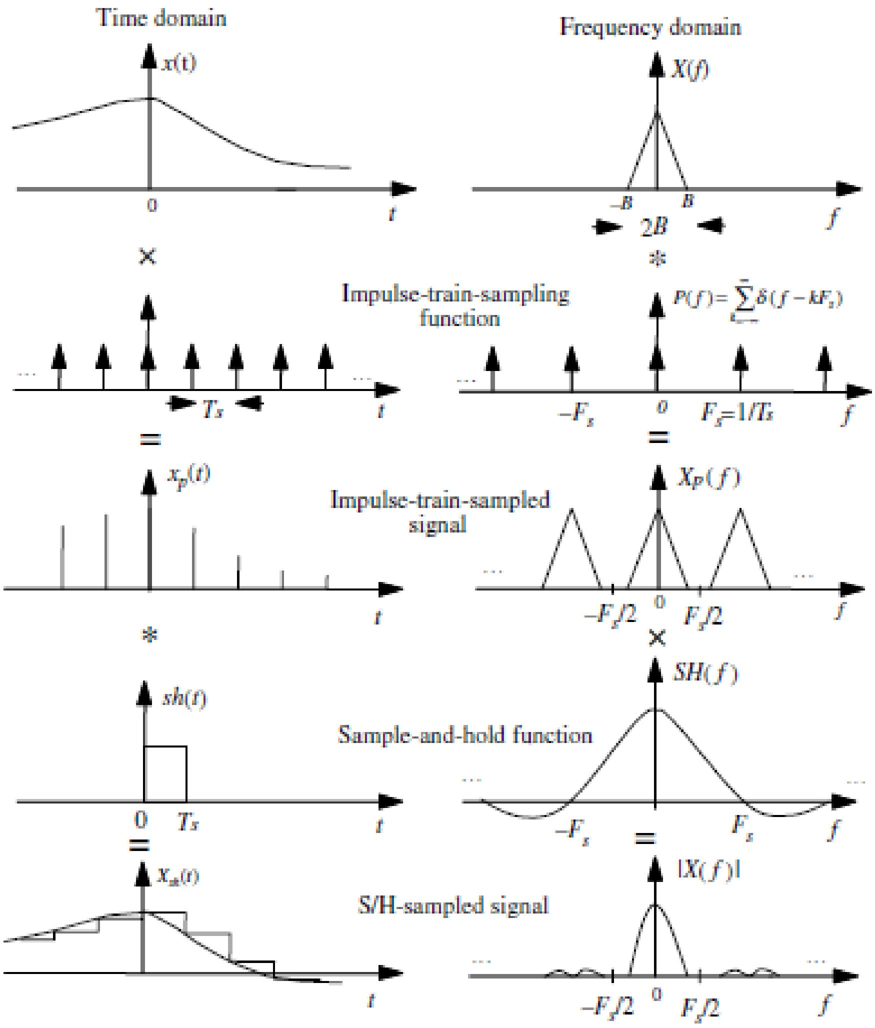


Figure 1.19 Sample-and-Hold signal modelled as impulse-train sampling followed by convolution with a rectangular pulse.

1.4.2 Quantisation

For digital signal processing, continuous-amplitude samples from the sample-and-hold are quantised and mapped into n-bit binary digits. For quantisation to n bits, the amplitude range of the signal is divided into 2^n discrete levels, and each sample is quantised to the nearest quantisation level, and then mapped to the binary code assigned to that level. Figure 1.21 illustrates the quantisation of a signal into 4 discrete levels. Quantisation is a many-to-one mapping, in that all the values that fall within the continuum of a quantisation band are mapped to the centre of the band. The mapping between an analog sample $x_a(m)$ and its quantised value $x(m)$ can be expressed as

$$x(m) = Q[x_a(m)] \tag{1.25}$$

where $Q[\cdot]$ is the quantising function.

The performance of a quantiser is measured by signal–to–quantisation noise ratio SQNR per bit. The quantisation noise is defined as

$$e(m) = x(m) - x_a(m) \tag{1.26}$$

Now consider an n-bit quantiser with an amplitude range of $\pm V$ volts. The quantisation step size is $\Delta = 2V/2^n$. Assuming that the quantisation noise is a zero-mean uniform process with an amplitude range of $\pm\Delta/2$ we can express the noise power as

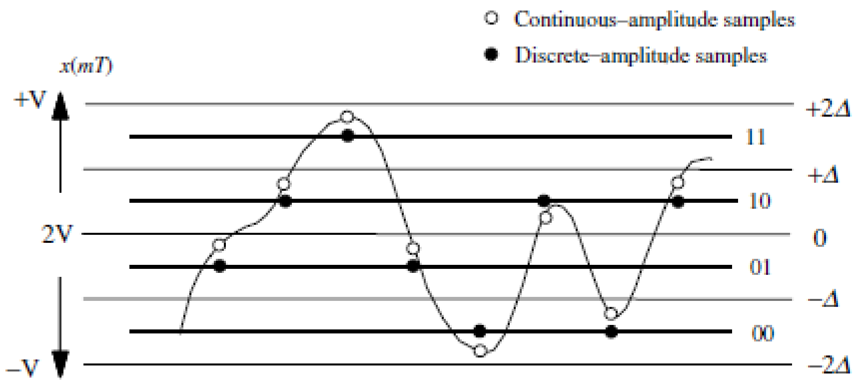
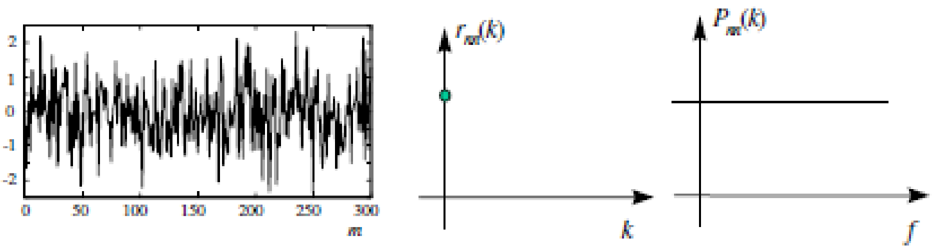


Figure 1.21 Offset-binary scalar quantisation

2.2 White Noise

White noise is defined as an uncorrelated noise process with equal power at all frequencies (Figure 2.1). A noise that has the same power at all frequencies in the range of $\pm\infty$ would necessarily need to have infinite power, and is therefore only a theoretical concept. However a band-limited noise process, with a flat spectrum covering the frequency range of a band-limited communication system, is to all intents and purposes from the point of view of the system a white noise process. For example, for an audio system with a bandwidth of 10 kHz, any flat-spectrum audio noise with a bandwidth greater than 10 kHz looks like a white noise.



(a) (b) (c)

Figure 2.1 Illustration of (a) white noise, (b) its autocorrelation, and (c) its power spectrum.

The autocorrelation function of a continuous-time zero-mean white noise process with a variance of σ^2 is a delta function given by

$$r_{NN}(\tau) = \mathcal{E}[N(t)N(t+\tau)] = \sigma^2\delta(\tau) \tag{2.1}$$

The power spectrum of a white noise, obtained by taking the Fourier transform of Equation (2.1), is given by

$$P_{NN}(f) = \int_{-\infty}^{\infty} r_{NN}(t)e^{-j2\pi ft}\,dt = \sigma^2 \tag{2.2}$$

The autocorrelation function of a continuous-time zero-mean white noise process with a variance of σ^2 is a delta function given by

$$r_{NN}(\tau) = \mathcal{E}[N(t)N(t+\tau)] = \sigma^2\delta(\tau) \tag{2.1}$$

The power spectrum of a white noise, obtained by taking the Fourier transform of Equation (2.1), is given by

$$P_{NN}(f) = \int_{-\infty}^{\infty} r_{NN}(t)e^{-j2\pi ft}\,dt = \sigma^2 \tag{2.2}$$

Equation (2.2) shows that a white noise has a constant power spectrum.

A pure white noise is a theoretical concept, since it would need to have infinite power to cover an infinite range of frequencies. Furthermore, a discrete-time signal by necessity has to be band-limited, with its highest frequency less than half the sampling rate. A more practical concept is band-limited white noise, defined as a noise with a flat spectrum in a limited bandwidth. The spectrum of band-limited white noise with a bandwidth of B Hz is given by

$$P_{NN}(f) = \begin{cases} \sigma^2, & |f| \le B \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

Thus the total power of a band-limited white noise process is $2B\sigma^2$. The autocorrelation function of a discrete-time band-limited white noise process is given by

$$r_{NN}(T_s k) = 2B\sigma^2 \frac{\sin(2\pi BT_s k)}{2\pi BT_s k} \tag{2.4}$$

where T_s is the sampling period. For convenience of notation T_s is usually assumed to be unity. For the case when T_s=1/2B, i.e. when the sampling rate is equal to the Nyquist rate, Equation (2.4) becomes

$$r_{NN}(T_s k) = 2B\sigma^2 \frac{\sin(\pi k)}{\pi k} = 2B\sigma^2\delta(k) \tag{2.5}$$

In Equation (2.5) the autocorrelation function is a delta function.

2.3 Coloured Noise

Although the concept of white noise provides a reasonably realistic and mathematically convenient and useful approximation to some predominant noise processes encountered in telecommunication systems, many other noise processes are non-white. The term coloured noise refers to any broadband noise with a non-white spectrum. For example most audio-frequency noise, such as the noise from moving cars, noise from computer fans, electric drill noise and people talking in the background, has a non-white predominantly low-frequency spectrum. Also, a white noise passing through a channel is "coloured" by the shape of the channel spectrum. Two classic varieties of coloured noise are so-called pink noise and brown noise, shown in Figures 2.2 and 2.3.

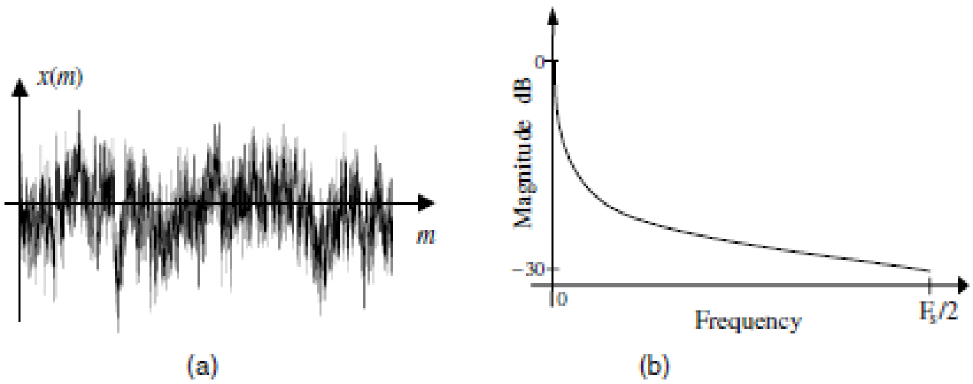


Figure 2.2 (a) A pink noise signal and (b) its magnitude spectrum.

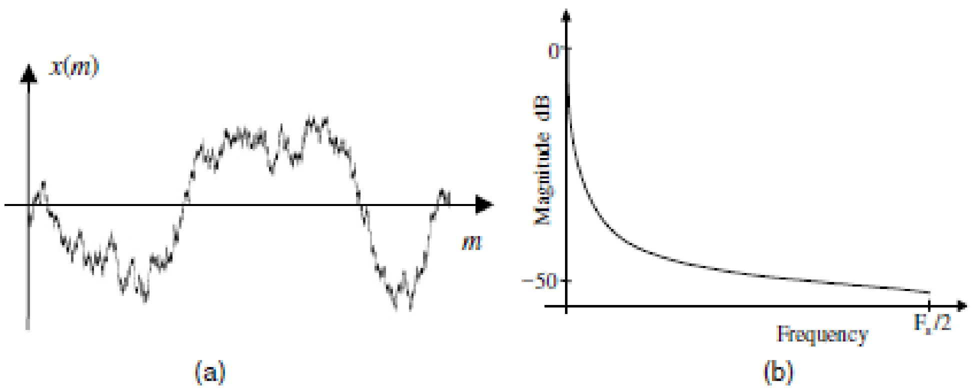


Figure 2.3 (a) A brown noise signal and (b) its magnitude spectrum.

Read about other noises Page 55 to 61

2.10.1 Additive White Gaussian Noise Model (AWGN)

In communication theory, it is often assumed that the noise is a stationary additive white Gaussian (AWGN) process. Although for some problems this is a valid assumption and leads to mathematically convenient and useful solutions, in practice the noise is often time-varying, correlated and non-Gaussian. This is particularly true for impulsive-type noise and for acoustic noise, which are non-stationary and non-Gaussian and hence cannot be modelled using the AWGN assumption. Non-stationary and non-Gaussian noise processes can be modelled by a Markovian chain of stationary sub-processes as described briefly in the next section and in detail in Chapter 5.

2.10.2 Hidden Markov Model for Noise

Most noise processes are non-stationary; that is the statistical parameters of the noise, such as its mean, variance and power spectrum, vary with time. Nonstationary processes may be modelled using the hidden Markov models (HMMs) described in detail in Chapter 5. An HMM is essentially a finite-state Markov chain of stationary subprocesses. The implicit assumption in using HMMs for noise is that the noise statistics can be modelled by a Markovian chain of stationary subprocesses. Note that a stationary noise process can be modelled by a single-state HMM. For a non-stationary noise, a multistate HMM can model the time variations of the noise process with a finite number of stationary states. For non-Gaussian noise, a mixture Gaussian density model can be used to model the space of the noise within each state. In general, the number of states per model and number of mixtures per state required to accurately model a noise process depends on

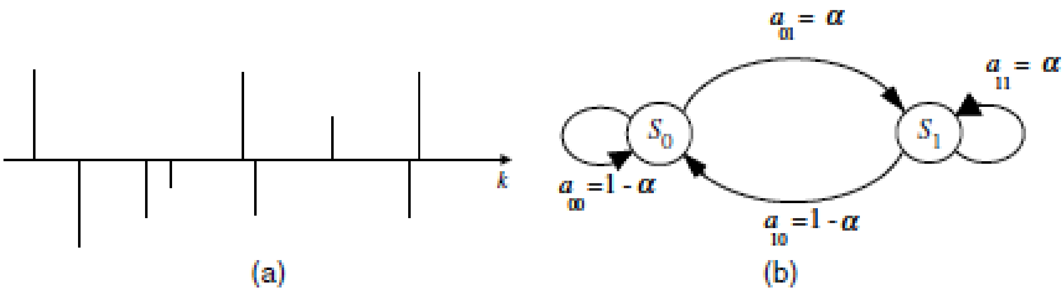


Figure 2.10 (a) An impulsive noise sequence. (b) A binary-state model of impulsive noise.

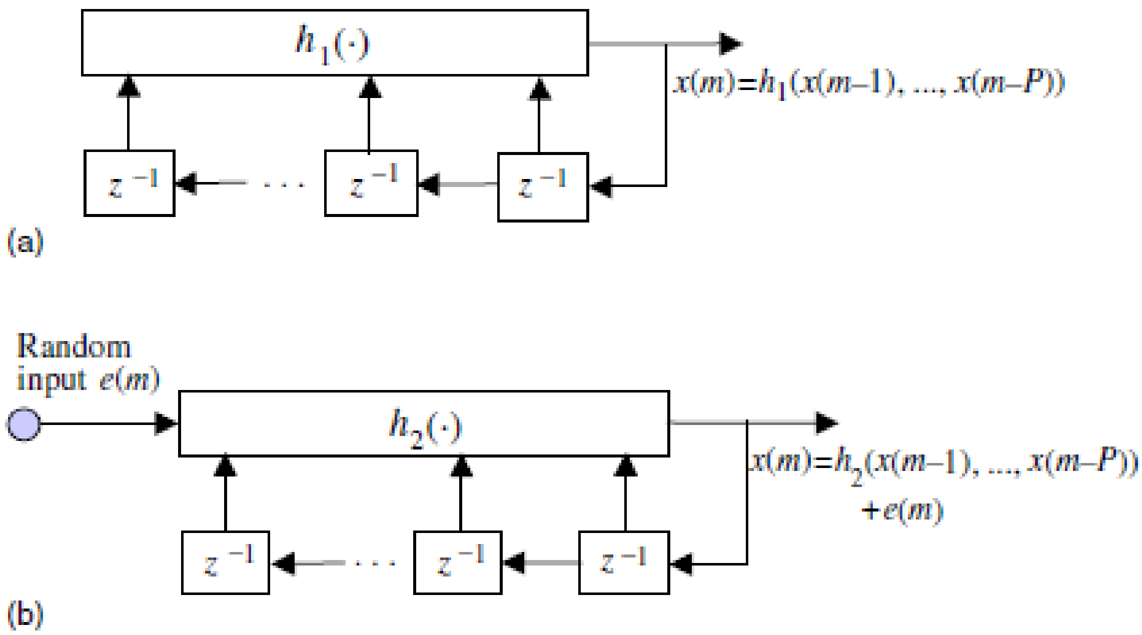


Figure 3.1 Illustration of deterministic and stochastic signal models: (a) a deterministic signal model, (b) a stochastic signal model.

$$x(m) = a\,x(m-1) - x(m-2) \tag{3.2}$$

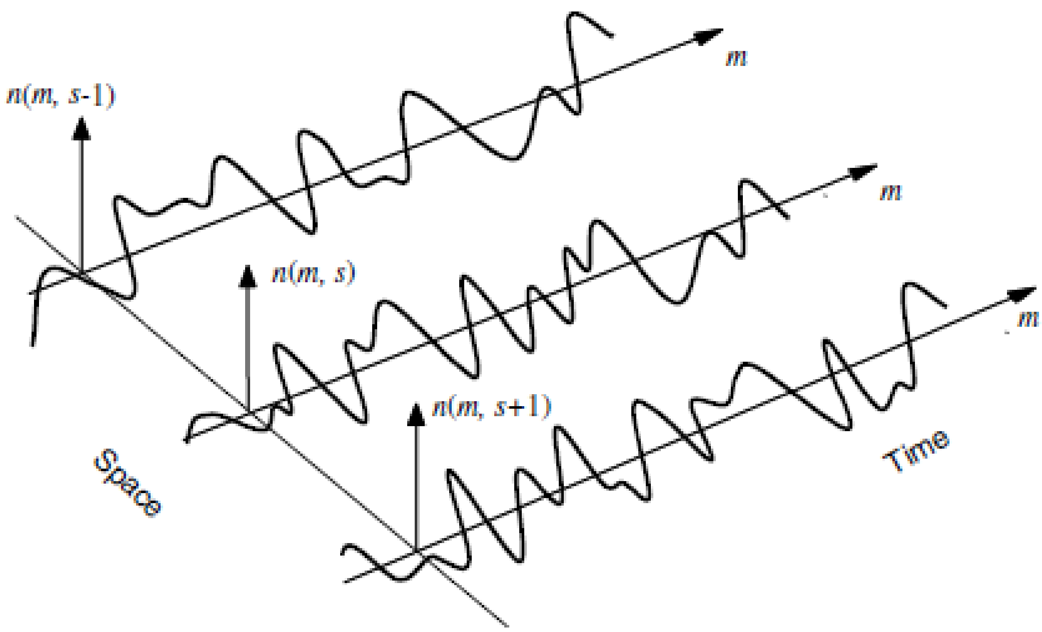


Figure 3.2 Illustration of three realisations in the space of a random noise $N(m)$.

Read Probability Functions Page 70+71

3.3.3 Non-Stationary Processes

A random process is non-stationary if its distributions or statistics vary with time. Most stochastic processes such as video signals, audio signals, financial data, meteorological data, biomedical signals, etc., are non-stationary, because they are generated by systems whose environments and parameters vary over time. For example, speech is a non-stationary process generated by a time-varying articulatory system. The loudness and the frequency composition of speech changes over time, and sometimes the change can be quite abrupt. Time-varying processes may be modelled by a combination of stationary random models as illustrated in Figure 3.5. In Figure 3.5(a) a non-stationary process is modelled as the output of a time-varying system whose parameters are controlled by a stationary process. In Figure 3.5(b) a time-varying process is modelled by a chain of time-invariant states, with each state having a different set of statistics or

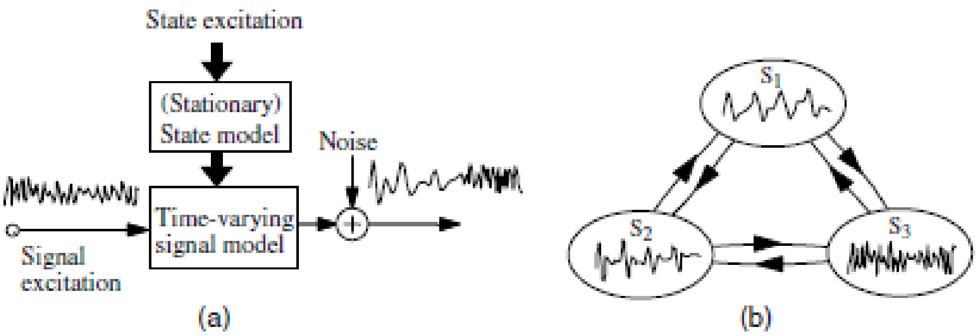


Figure 3.5 Two models for non-stationary processes: (a) a stationary process drives the parameters of a continuously time-varying model; (b) a finite-state

3.4.4 Power Spectral Density

The power spectral density (PSD) function, also called the power spectrum, of a random process gives the spectrum of the distribution of the power among the individual frequency contents of the process. The power spectrum of a wide sense stationary process $X(m)$ is defined, by the Wiener–Khinchin theorem in Chapter 9, as the Fourier transform of the autocorrelation function:

$$\begin{aligned} P_{XX}(f) &= \mathcal{E}[X(f)X^*(f)] \\ &= \sum_{m=-\infty}^{\infty} r_{xx}(k)e^{-j2\pi fm} \end{aligned} \tag{3.45}$$

where $r_{xx}(m)$ and $P_{XX}(f)$ are the autocorrelation and power spectrum of $x(m)$ respectively, and f is the frequency variable. For a real-valued stationary process, the autocorrelation is symmetric, and the power spectrum may be written as

$$P_{XX}(f) = r_{xx}(0) + \sum_{m=1}^{\infty} 2r_{xx}(m)\cos(2\pi fm) \tag{3.46}$$

The power spectral density is a real-valued non-negative function, expressed in units of watts per hertz. From Equation (3.45), the autocorrelation sequence of a random process may be obtained as the inverse Fourier transform of the power spectrum as

$$r_{xx}(m) = \int_{-1/2}^{1/2} P_{XX}(f)\, e^{j2\pi fm}\, df \tag{3.47}$$

Note that the autocorrelation and the power spectrum represent the second order statistics of a process in the time and frequency domains respectively.

3.4.9 Mean-Ergodic Processes

The time-averaged estimate of the mean of a signal $x(m)$ obtained from N samples is given by

$$\hat{\mu}_X = \frac{1}{N}\sum_{m=0}^{N-1} x(m) \tag{3.65}$$

A stationary process is said to be mean-ergodic if the time-averaged value of an infinitely long realisation of the process is the same as the ensemble-mean taken across the space of the process. Therefore, for a mean-ergodic process, we have

$$\lim_{N\to\infty} \mathcal{E}[\hat{\mu}_X] = \mu_X \tag{3.66}$$

$$\lim_{N\to\infty} \mathrm{var}[\hat{\mu}_X] = 0 \tag{3.67}$$

where μ_X is the "true" ensemble average of the process. Condition (3.67) is also referred to as mean-ergodicity in the mean square error (or minimum variance of error) sense. The time-averaged estimate of the mean of a signal, obtained from a random realisation of the process, is itself a random variable, with is own mean, variance and probability density function. If the number of observation samples N is relatively large then, from the central limit theorem the probability density function of the estimate $\hat{\mu}_X$ is Gaussian. The expectation of $\hat{\mu}_X$ is given by

$$\mathcal{E}[\hat{\mu}_x] = \mathcal{E}\left[\frac{1}{N}\sum_{m=0}^{N-1} x(m)\right] = \frac{1}{N}\sum_{m=0}^{N-1}\mathcal{E}[x(m)] = \frac{1}{N}\sum_{m=0}^{N-1}\mu_x = \mu_x \tag{3.68}$$

Page 85 to 104, Read other processes.

3.5.6 Shot Noise

Shot noise happens when there is randomness in a directional flow of particles: as in the flow of electrons from the cathode to the anode of a cathode ray tube, the flow of photons in a laser beam, the flow and recombination of electrons and holes in semiconductors, and the flow of photoelectrons emitted in photodiodes. Shot noise has the form of a random pulse sequence. The pulse sequence can be modelled as the response of a linear filter excited by a Poisson-distributed binary impulse input sequence (Figure 3.11). Consider a Poisson-distributed binary-valued impulse process $x(t)$. Divide the time axis into uniform short intervals of Δt such that only one occurrence of an impulse is possible within each time interval. Let $x(m\Delta t)$ be "1" if an impulse is present in the interval $m\Delta t$ to $(m+1)\Delta t$, and "0" otherwise. For $x(m\Delta t)$, we have

$$\mathcal{E}[x(m\Delta t)] = 1 \times P(x(m\Delta t) = 1) + 0 \times P(x(m\Delta t) = 0) = \lambda \Delta t \qquad (3.109)$$

and

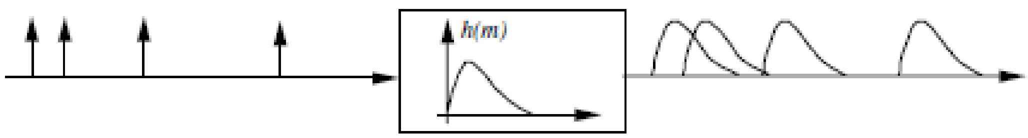


Figure 3.11 Shot noise is modelled as the output of a filter excited with a process.

$$\mathcal{E}[x(m\Delta t)x(n\Delta t)] = \begin{cases} 1 \times P(x(m\Delta t) = 1) = \lambda \Delta t, & m = n \\ 1 \times P(x(m\Delta t) = 1)) \times P(x(n\Delta t) = 1) = (\lambda \Delta t)^2, & m \neq n \end{cases} \qquad (3.110)$$

BAYESIAN ESTIMATION

Bayesian estimation is a framework for the formulation of statistical inference problems. In the prediction or estimation of a random process from a related observation signal, the Bayesian philosophy is based on combining the evidence contained in the signal with prior knowledge of the probability distribution of the process. Bayesian methodology includes the classical estimators such as maximum a posteriori (MAP), maximum-likelihood (ML), minimum mean square error (MMSE) and minimum mean absolute value of error (MAVE) as special cases. The hidden Markov model, widely used in statistical signal processing, is an example of a Bayesian model. Bayesian inference is based on minimisation of the so-called Bayes' risk function, which includes a posterior model of the unknown parameters given the observation and a cost-of-error function. This chapter begins with an introduction to the basic concepts of estimation theory, and considers the statistical measures that are used to quantify the performance of an estimator. We study Bayesian estimation methods and consider the effect of using a prior model on the mean and the variance of an estimate. The estimate–maximise (EM) method for the estimation of a set of unknown parameters from an incomplete observation is studied, and applied to the mixture Gaussian modelling of the space of a continuous random variable. This chapter concludes with an introduction to the Bayesian classification of discrete or finite-state signals, and the K-means clustering method.

4.1 Bayesian Estimation Theory: Basic Definitions

Estimation theory is concerned with the determination of the best estimate of an unknown parameter vector from an observation signal, or the recovery of a clean signal degraded by noise and distortion. For example, given a noisy sine wave, we may be interested in estimating its basic parameters (i.e. amplitude, frequency and phase), or we may wish to recover the signal itself. An estimator takes as the input a set of noisy or incomplete observations, and, using a dynamic model (e.g. a linear predictive model) and/or a probabilistic model (e.g. Gaussian model) of the process, estimates the unknown parameters. The estimation accuracy depends on the available information and on the efficiency of the estimator. In this chapter, the Bayesian estimation of continuous-valued parameters is studied. The modelling and classification of finite-state parameters is covered in the next chapter.

Bayesian theory is a general inference framework. In the estimation or prediction of the state of a process, the Bayesian method employs both the evidence contained in the observation signal and the accumulated prior probability of the process. Consider the estimation of the value of a random parameter vector $\boldsymbol{\theta}$, given a related observation vector $\boldsymbol{y}$. From Bayes' rule the posterior probability density function (pdf) of the parameter vector $\boldsymbol{\theta}$ given $\boldsymbol{y}$, $f_{\Theta|Y}(\boldsymbol{\theta}\,|\,\boldsymbol{y})$, can be expressed as

$$f_{\Theta|Y}(\boldsymbol{\theta}\,|\,\boldsymbol{y}) = \frac{f_{Y|\Theta}(\boldsymbol{y}\,|\,\boldsymbol{\theta})\, f_{\Theta}(\boldsymbol{\theta})}{f_Y(\boldsymbol{y})} \tag{4.1}$$

where for a given observation, $f_Y(\boldsymbol{y})$ is a constant and has only a normalising effect. Thus there are two variable terms in Equation (4.1): one term $f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})$ is the likelihood that the observation signal $\boldsymbol{y}$ was generated by the parameter vector $\boldsymbol{\theta}$ and the second term is the prior probability of the parameter vector having a value of $\boldsymbol{\theta}$. The relative influence of the likelihood pdf $f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})$ and the prior pdf $f_{\Theta}(\boldsymbol{\theta})$ on the posterior pdf $f_{\Theta|Y}(\boldsymbol{\theta}|\boldsymbol{y})$ depends on the shape of these function, i.e. on how relatively peaked each pdf is. In general the more peaked a probability density function, the more it will influence the outcome of the estimation process. Conversely, a uniform pdf will have no influence.
The remainder of this chapter is concerned with different forms of Bayesian estimation and its applications. First, in this section, some basic concepts of estimation theory are introduced.

4.1.1 Dynamic and Probability Models in Estimation

Optimal estimation algorithms utilise dynamic and statistical models of the observation signals. A dynamic predictive model captures the correlation structure of a signal, and models the dependence of the present and future values of the signal on its past trajectory and the input stimulus. A statistical probability model characterises the random fluctuations of a signal in terms of its statistics, such as the mean and the covariance, and most completely in terms of a probability model. Conditional probability models, in addition to modelling the random fluctuations of a signal, can also model the dependence of the signal on its past values or on some other related process.

As an illustration consider the estimation of a P-dimensional parameter vector $\boldsymbol{\theta} = [\theta_0, \theta_1, \ldots, \theta_{P-1}]$ from a noisy observation vector $\boldsymbol{y} = [y(0), y(1), \ldots, y(N-1)]$ modelled as

$$\boldsymbol{y} = h(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{e}) + \boldsymbol{n} \tag{4.2}$$

where, as illustrated in Figure 4.1, the function $h(\cdot)$ with a random input $\boldsymbol{e}$, output $\boldsymbol{x}$, and parameter vector $\boldsymbol{\theta}$, is a predictive model of the signal $\boldsymbol{x}$, and $\boldsymbol{n}$ is an additive random noise process. In Figure 4.1, the distributions of the random noise $\boldsymbol{n}$, the random input $\boldsymbol{e}$ and the parameter vector $\boldsymbol{\theta}$ are modelled by probability density functions, $f_N(\boldsymbol{n})$, $f_E(\boldsymbol{e})$, and $f_\Theta(\boldsymbol{\theta})$ respectively. The pdf model most often used is the Gaussian model. Predictive and statistical models of a process *guide* the estimator towards the set of values of the unknown parameters that are most consistent with both the prior distribution of the model parameters and the noisy observation. In general, the more modelling information used in an estimation process, the better the results, provided that the models are an accurate characterisation of the observation and the parameter process.

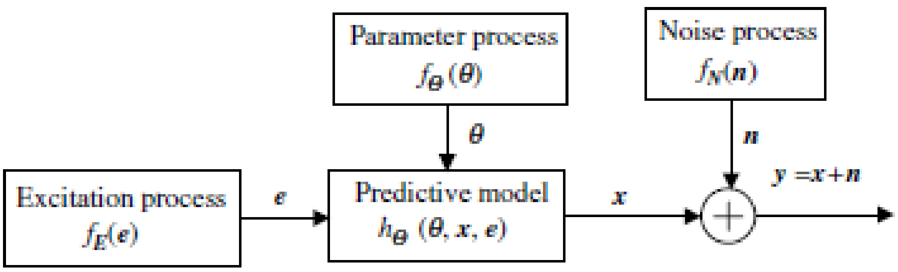


Figure 4.1 A random process $\boldsymbol{y}$ is described in terms of a predictive model $h(\cdot)$, and statistical models $f_E(\cdot)$, $f_\Theta(\cdot)$ and $f_N(\cdot)$.

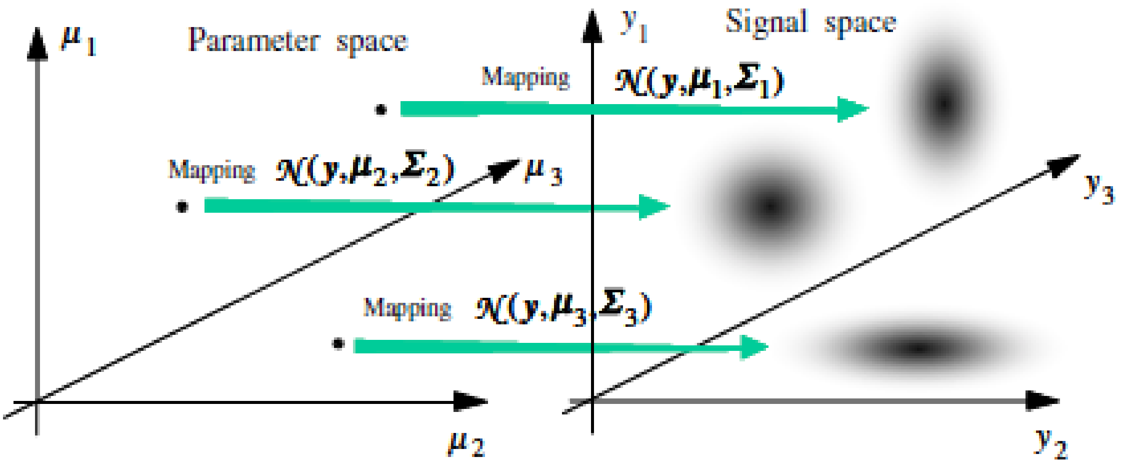


Figure 4.2 Illustration of three points in the parameter space of a Gaussian process and the associated signal spaces, for simplicity the variances are not shown in parameter space.

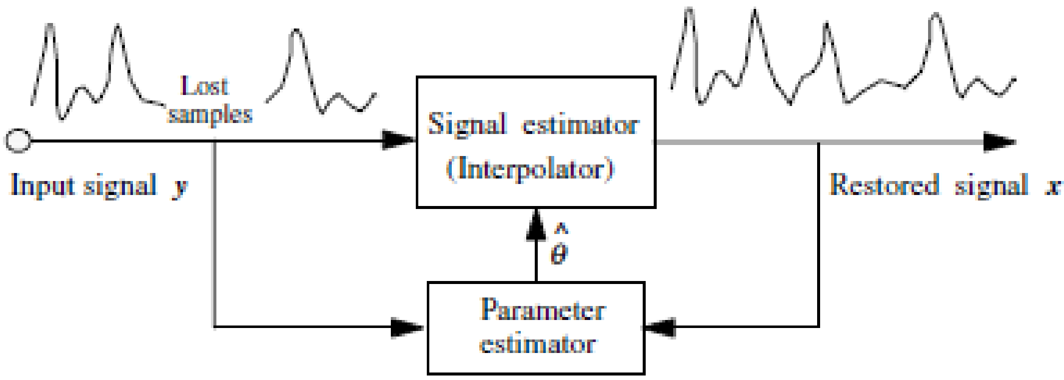


Figure 4.3 Illustration of signal restoration using a parametric model of the signal process.

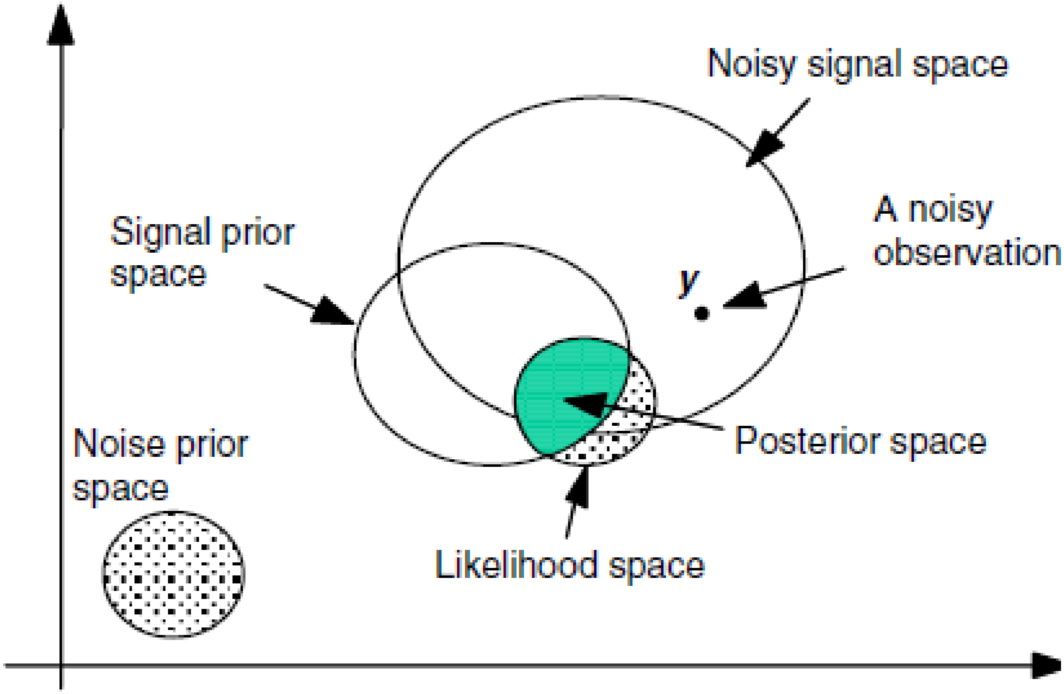


Figure 4.6 Sketch of a two-dimensional signal and noise spaces, and the likelihood and posterior spaces of a noisy observation ***y***.

4.2 Bayesian Estimation

The Bayesian estimation of a parameter vector $\boldsymbol{\theta}$ is based on the minimisation of a Bayesian risk function defined as an average cost-of-error function:

$$\begin{aligned}\mathcal{R}(\hat{\boldsymbol{\theta}}) &= \mathcal{E}[C(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})] \\ &= \int_{\theta}\int_{Y} C(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})\, f_{Y,\Theta}(\boldsymbol{y},\boldsymbol{\theta})\, d\boldsymbol{y}\, d\boldsymbol{\theta} \\ &= \int_{\theta}\int_{Y} C(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})\, f_{\Theta|Y}(\boldsymbol{\theta}\mid \boldsymbol{y})\, f_{Y}(\boldsymbol{y})\, d\boldsymbol{y}\, d\boldsymbol{\theta}\end{aligned} \tag{4.18}$$

where the cost-of-error function $C(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})$ allows the appropriate weighting of the various outcomes to achieve desirable objective or subjective properties. The cost function can be chosen to associate a high cost with outcomes that are undesirable or disastrous. For a given observation vector $\boldsymbol{y}$, $f_{\boldsymbol{Y}}(\boldsymbol{y})$ is a constant and has no effect on the risk-minimisation process. Hence Equation (4.18) may be written as a conditional risk function:

$$\mathcal{R}(\hat{\boldsymbol{\theta}}\mid \mathbf{y}) = \int_{\theta} C(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})\, f_{\Theta|Y}(\boldsymbol{\theta}\mid \mathbf{y})\, d\boldsymbol{\theta} \tag{4.19}$$

The Bayesian estimate obtained as the minimum-risk parameter vector is given by

$$\hat{\boldsymbol{\theta}}_{\text{Bayesian}} = \arg\min_{\hat{\theta}} \mathcal{R}(\hat{\boldsymbol{\theta}}\mid \mathbf{y}) = \arg\min_{\hat{\theta}} \left[\int_{\theta} C(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})\, f_{\Theta|Y}(\boldsymbol{\theta}\mid \mathbf{y})\, d\boldsymbol{\theta}\right] \tag{4.20}$$

Using Bayes' rule, Equation (4.20) can be written as

$$\hat{\boldsymbol{\theta}}_{\text{Bayesian}} = \arg\min_{\hat{\theta}} \left[\int_{\theta} C(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})\, f_{Y|\Theta}(\boldsymbol{y}\mid \boldsymbol{\theta})\, f_{\Theta}(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right] \tag{4.21}$$

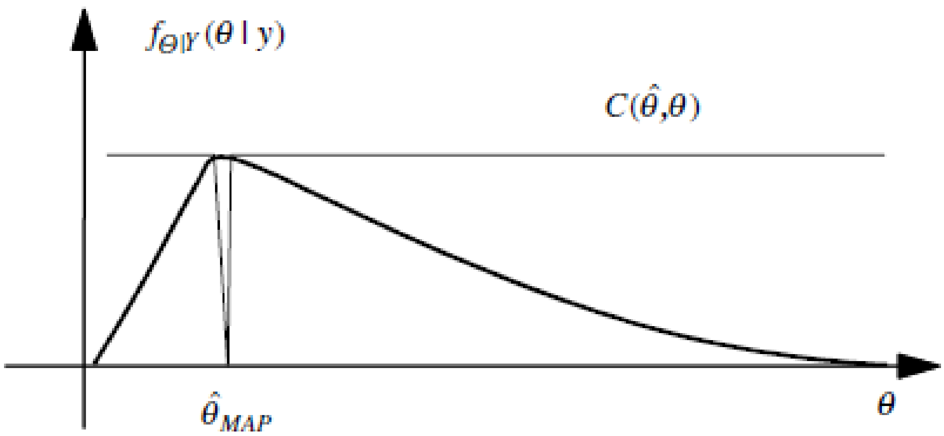


Figure 4.7 Illustration of the Bayesian cost function for the MAP estimate.

Page 121 to 128 Other estimation

4.3 The Estimate–Maximise (EM) Method

The EM algorithm is an iterative likelihood maximisation method with applications in blind deconvolution, model-based signal interpolation, spectral estimation from noisy observations, estimation of a set of model parameters from a training data set, etc. The EM is a framework for solving problems where it is difficult to obtain a direct ML estimate either because the data is incomplete or because the problem is difficult.

To define the term *incomplete data*, consider a signal $\boldsymbol{x}$ from a random process $\boldsymbol{X}$ with an unknown parameter vector $\boldsymbol{\theta}$ and a pdf $f_{X;\Theta}(\boldsymbol{x};\boldsymbol{\theta})$. The notation $f_{X;\Theta}(\boldsymbol{x};\boldsymbol{\theta})$ expresses the dependence of the pdf of $\boldsymbol{X}$ on the value of the unknown parameter $\boldsymbol{\theta}$. The signal $\boldsymbol{x}$ is the so-called *complete data* and the ML estimate of the parameter vector $\boldsymbol{\theta}$ may be obtained from $f_{X;\Theta}(\boldsymbol{x};\boldsymbol{\theta})$. Now assume that the signal $\boldsymbol{x}$ goes through a many-to-one non-invertible transformation (e.g. when a number of samples of the vector $\boldsymbol{x}$ are lost) and is observed as $\boldsymbol{y}$. The observation $\boldsymbol{y}$ is the so-called incomplete data. Maximisation of the likelihood of the incomplete data, $f_{Y;\Theta}(\boldsymbol{y};\boldsymbol{\theta})$, with respect to the parameter vector $\boldsymbol{\theta}$ is often a difficult task, whereas maximisation of the likelihood of the complete data $f_{X;\Theta}(\boldsymbol{x};\boldsymbol{\theta})$ is relatively easy. Since the complete data is unavailable, the parameter estimate is obtained through maximisation of the *conditional expectation* of the log-likelihood of the complete data defined as

$$\mathcal{E}\left[\ln f_{X;\Theta}(\boldsymbol{x};\boldsymbol{\theta}) \middle| \boldsymbol{y}\right] = \int_{X} f_{X|Y;\Theta}(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta}) \ln f_{X;\Theta}(\boldsymbol{x};\boldsymbol{\theta})\, d\boldsymbol{x} \tag{4.86}$$

In Equation (4.86), the computation of the term $f_{X|Y;\Theta}(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta})$ requires an estimate of the unknown parameter vector $\boldsymbol{\theta}$. For this reason, the expectation of the likelihood function is maximised iteratively starting with an initial estimate of $\boldsymbol{\theta}$, and updating the estimate as described in the following.

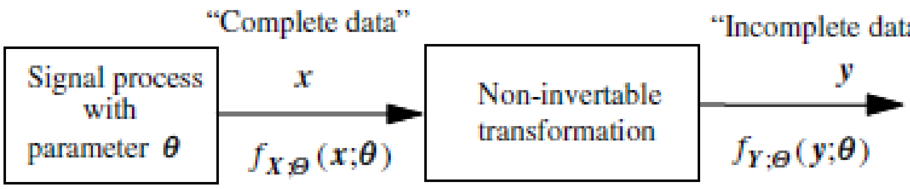


Figure 4.14 Illustration of transformation of complete data to incomplete data.

HIDDEN MARKOV MODELS

Hidden Markov models (HMMs) are used for the statistical modelling of non-stationary signal processes such as speech signals, image sequences and time-varying noise. An HMM models the time variations (and/or the space variations) of the statistics of a random process with a Markovian chain of state-dependent stationary subprocesses. An HMM is essentially a Bayesian finite state process, with a Markovian prior for modelling the transitions between the states, and a set of state probability density functions for modelling the random variations of the signal process within each state. This chapter begins with a brief introduction to continuous and finite state non-stationary models, before concentrating on the theory and applications of hidden Markov models. We study the various HMM structures, the Baum–Welch method for the maximum-likelihood training of the parameters of an HMM, and the use of HMMs and the Viterbi decoding algorithm for the classification and decoding of an unlabelled observation signal sequence. Finally, applications of the HMMs for the enhancement of noisy signals are considered.

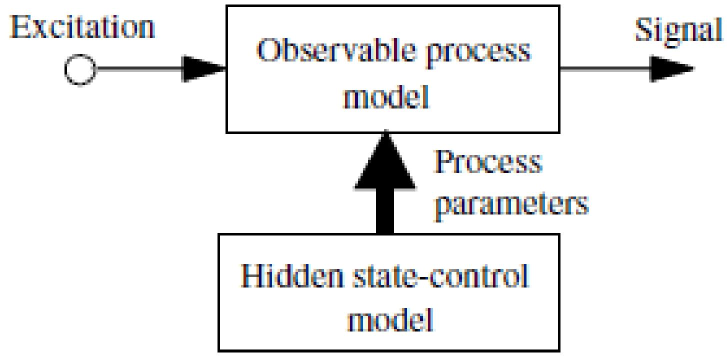


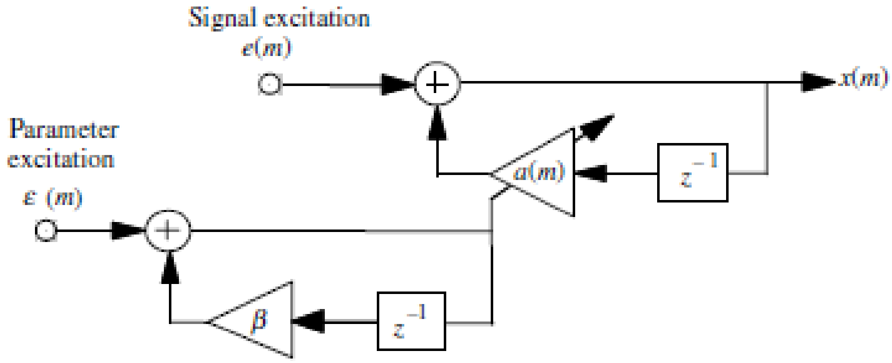
Figure 5.1 Illustration of a two-layered model of a non-stationary process.

5.1 Statistical Models for Non-Stationary Processes

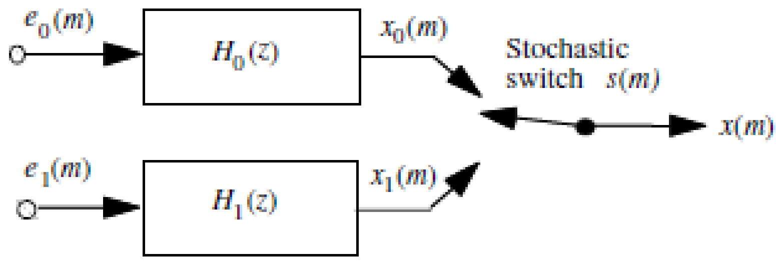
A non-stationary process can be defined as one whose statistical parameters vary over time. Most "naturally generated" signals, such as audio signals, image signals, biomedical signals and seismic signals, are non-stationary, in that the parameters of the systems that generate the signals, and the environments in which the signals propagate, change with time.

A non-stationary process can be modelled as a double-layered stochastic process, with a hidden process that controls the time variations of the statistics of an observable process, as illustrated in Figure 5.1. In general, non-stationary processes can be classified into one of two broad categories:

(a) *Continuously variable state* processes.
(b) *Finite state* processes.



(a)



(b)

Figure 5.2 (a) A continuously variable state AR process. (b) A binary-state AR process.

parameters of the non-stationary AR model. For this model, the observation signal equation and the parameter state equation can be expressed as

$$x(m) = a(m)x(m-1) + e(m) \qquad \text{Observation equation} \qquad (5.1)$$

$$a(m) = \beta a(m-1) + \varepsilon(m) \qquad \text{Hidden state equation} \qquad (5.2)$$

where $a(m)$ is the time-varying coefficient of the observable AR process and β is the coefficient of the hidden state-control process.

A simple example of a finite state non-stationary model is the binary-state autoregressive process illustrated in Figure 5.2(b), where at each time instant a random switch selects one of the two AR models for connection to the output terminal. For this model, the output signal $x(m)$ can be expressed as

$$x(m) = \bar{s}(m)x_0(m) + s(m)x_1(m) \qquad (5.3)$$

where the binary switch $s(m)$ selects the state of the process at time m, and $\bar{s}(m)$ denotes the Boolean complement of $s(m)$.

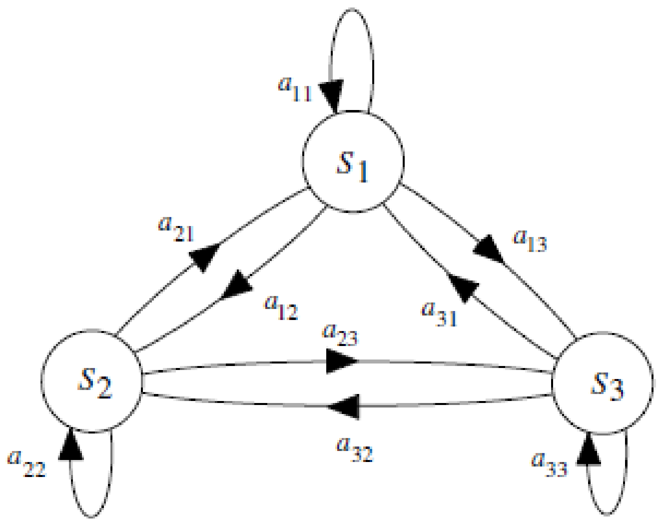
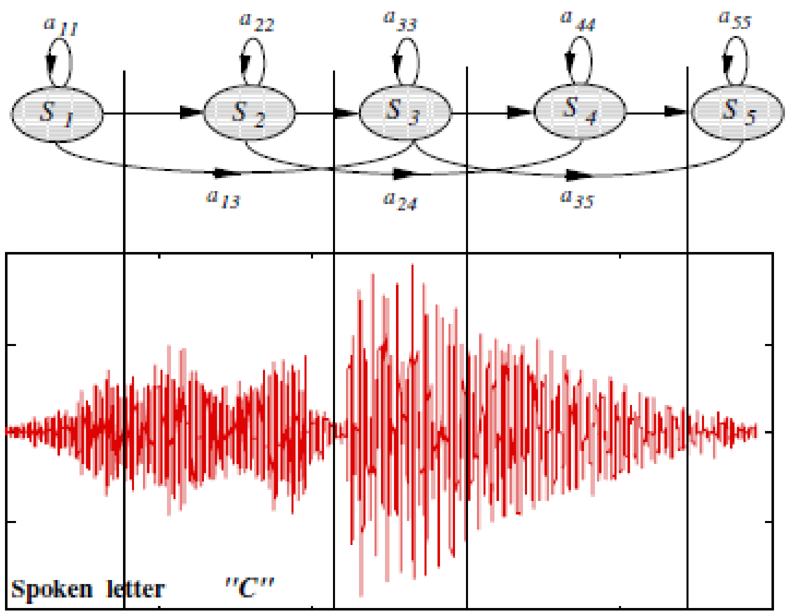


Figure 5.4 A three-state ergodic HMM structure.



5.2.3 Parameters of a Hidden Markov Model

A hidden Markov model has the following parameters:

Number of states N. This is usually set to the total number of distinct, or elementary, stochastic events in a signal process. For example, in modelling a binary-state process such as impulsive noise, N is set to 2, and in isolated-word speech modelling N is set between 5 to 10.

State transition-probability matrix $\boldsymbol{A}=\{a_{ij}, i,j=1, \dots N\}$. This provides a Markovian connection network between the states, and models the variations in the duration of the signals associated with each state. For a left–right HMM (see Figure 5.5), $a_{ij}=0$ for $i>j$, and hence the transition matrix $\boldsymbol{A}$ is upper-triangular.

State observation vectors $\{\boldsymbol{\mu}_{i1}, \boldsymbol{\mu}_{i2}, \dots, \boldsymbol{\mu}_{iM}, i=1, \dots, N\}$. For each state a set of M prototype vectors model the centroids of the signal space associated with each state.

State observation vector probability model. This can be either a discrete model composed of the M prototype vectors and their associated probability mass function (pmf) $\boldsymbol{P}=\{P_{ij}(\cdot); i=1, \dots, N, j=1, \dots M\}$, or it may be a continuous (usually Gaussian) pdf model $\boldsymbol{F}=\{f_{ij}(\cdot); i=1, \dots, N, j=1, \dots, M\}$.

Initial state probability vector $\boldsymbol{\pi}=[\pi_1, \pi_2, \dots, \pi_N]$.

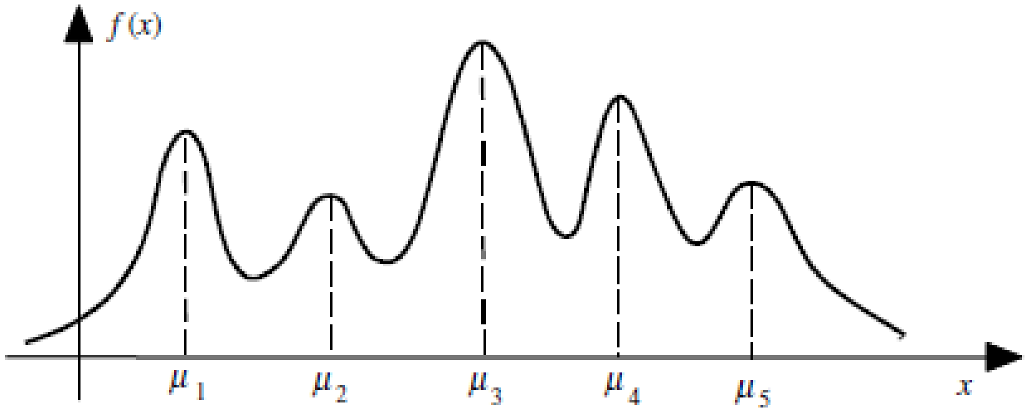


Figure 5.7 A mixture Gaussian probability density function.

5.2.5 State Transition Probabilities

The first-order Markovian property of an HMM entails that the transition probability to any state $s(t)$ at time t depends only on the state of the process at time t–1, $s(t$–1), and is independent of the previous states of the HMM. This can be expressed as

$$\begin{aligned} &Prob\big(s(t)=j \,\big|\, s(t-1)=i, s(t-2)=k, \ldots, s(t-N)=l\big) \\ &=Prob\big(s(t)=j \,\big|\, s(t-1)=i\big)=a_{ij} \end{aligned} \tag{5.7}$$

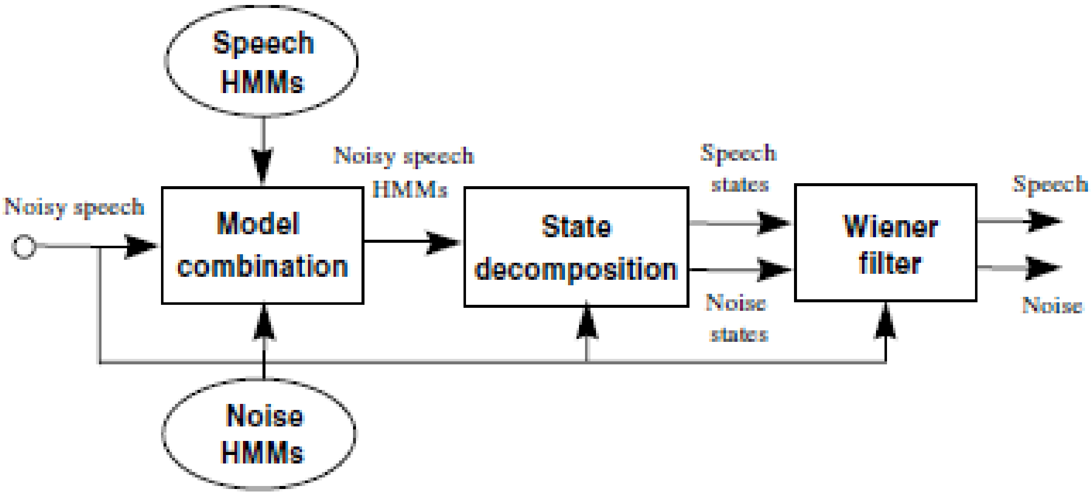


Figure 5.12 Outline configuration of HMM-based noisy speech recognition and enhancement.

5.6.2 Decomposition of State Sequences of Signal and Noise

The HMM-based state decomposition problem can be stated as follows: given a noisy signal and the HMMs of the signal and the noise processes, estimate the underlying states of the signal and the noise.

HMM state decomposition can be obtained using the following method:

(a) Given the noisy signal and a set of combined signal and noise models, estimate the maximum-likelihood (ML) combined noisy HMM for the noisy signal.
(b) Obtain the ML state sequence of from the ML combined model.
(c) Extract the signal and noise states from the ML state sequence of the ML combined noisy signal model.

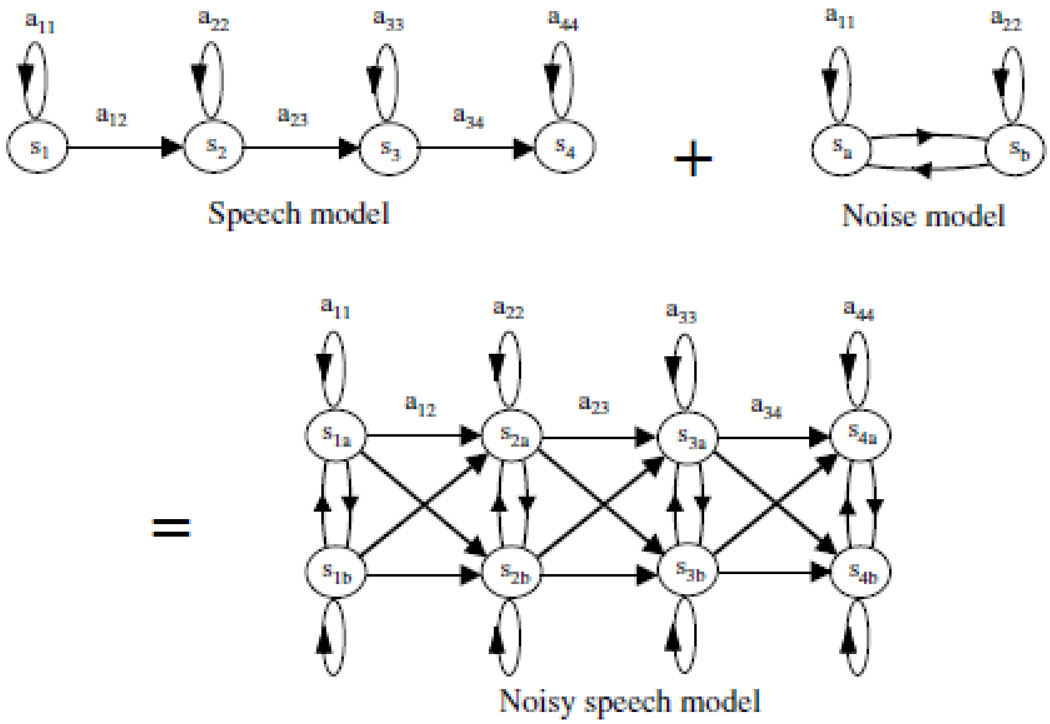


Figure 5.13 Outline configuration of HMM-based noisy speech recognition and enhancement. S_{ij} is a combination of the state i of speech with the state j of noise.

5.7 HMM-Based Wiener Filters

The least mean square error Wiener filter is derived in Chapter 6. For a stationary signal $x(m)$, observed in an additive noise $n(m)$, the Wiener filter equations in the time and the frequency domains are derived as :

$$\boldsymbol{w} = (\boldsymbol{R}_{xx} + \boldsymbol{R}_{nn})^{-1}\boldsymbol{r}_{xx} \tag{5.55}$$

and

$$W(f) = \frac{P_{XX}(f)}{P_{XX}(f) + P_{NN}(f)} \tag{5.56}$$

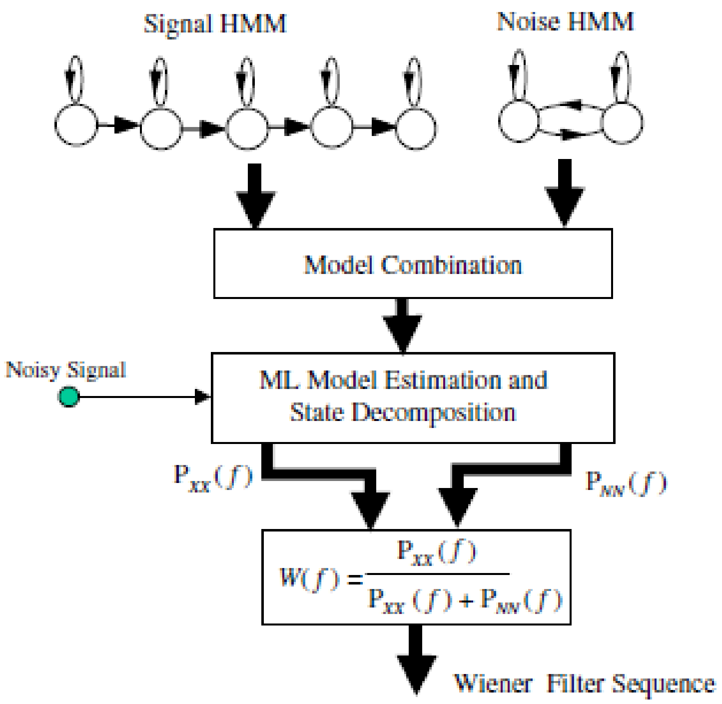


Figure 5.14 Illustrations of HMMs with state-dependent Wiener filters.

WIENER FILTERS

Wiener theory, formulated by Norbert Wiener, forms the foundation of data-dependent linear least square error filters. Wiener filters play a central role in a wide range of applications such as linear prediction, echo cancellation, signal restoration, channel equalisation and system identification. The coefficients of a Wiener filter are calculated to minimise the average squared distance between the filter output and a desired signal. In its basic form, the Wiener theory assumes that the signals are stationary processes. However, if the filter coefficients are periodically recalculated for every block of N signal samples then the filter adapts itself to the average characteristics of the signals within the blocks and becomes block-adaptive. A block-adaptive (or segment adaptive) filter can be used for signals such as speech and image that may be considered almost stationary over a relatively small block of samples. In this chapter, we study Wiener filter theory, and consider alternative methods of formulation of the Wiener filter problem. We consider the application of Wiener filters in channel equalisation, time-delay estimation and additive noise reduction. A case study of the frequency response of a Wiener filter, for additive noise reduction, provides useful insight into the operation of the filter. We also deal with some implementation issues of Wiener filters.

6.1 Wiener Filters: Least Square Error Estimation

Wiener formulated the continuous-time, least mean square error, estimation problem in his classic work on interpolation, extrapolation and smoothing of time series (Wiener 1949). The extension of the Wiener theory from continuous time to discrete time is simple, and of more practical use for implementation on digital signal processors. A Wiener filter can be an infinite-duration impulse response (IIR) filter or a finite-duration impulse response (FIR) filter. In general, the formulation of an IIR Wiener filter results in a set of non-linear equations, whereas the formulation of an FIR Wiener filter results in a set of linear equations and has a closed-form solution. In this chapter, we consider FIR Wiener filters, since they are relatively simple to compute, inherently stable and more practical. The main drawback of FIR filters compared with IIR filters is that they may need a large number of coefficients to approximate a desired response.

Figure 6.1 illustrates a Wiener filter represented by the coefficient vector $\boldsymbol{w}$. The filter takes as the input a signal $y(m)$, and produces an output signal $\hat{x}(m)$, where $\hat{x}(m)$ is the least mean square error estimate of a desired or target signal $x(m)$. The filter input–output relation is given by

$$\begin{aligned}\hat{x}(m) &= \sum_{k=0}^{P-1} w_k\, y(m-k) \\ &= \boldsymbol{w}^{\mathrm{T}} \boldsymbol{y}\end{aligned} \tag{6.1}$$

where m is the discrete-time index, $\boldsymbol{y}^{\mathrm{T}}=[y(m), y(m-1), \ldots, y(m-P-1)]$ is the filter input signal, and the parameter vector $\boldsymbol{w}^{\mathrm{T}}=[w_0, w_1, \ldots, w_{P-1}]$ is the Wiener filter coefficient vector. In Equation (6.1), the filtering operation is expressed in two alternative and equivalent forms of a convolutional sum and an inner vector product. The Wiener filter error signal, $e(m)$ is defined as the difference between the desired signal $x(m)$ and the filter output signal $\hat{x}(m)$:

$$\begin{aligned}e(m) &= x(m) - \hat{x}(m) \\ &= x(m) - \boldsymbol{w}^{\mathrm{T}} \boldsymbol{y}\end{aligned} \tag{6.2}$$

In Equation (6.2), for a given input signal $y(m)$ and a desired signal $x(m)$, the filter error $e(m)$ depends on the filter coefficient vector $\boldsymbol{w}$.

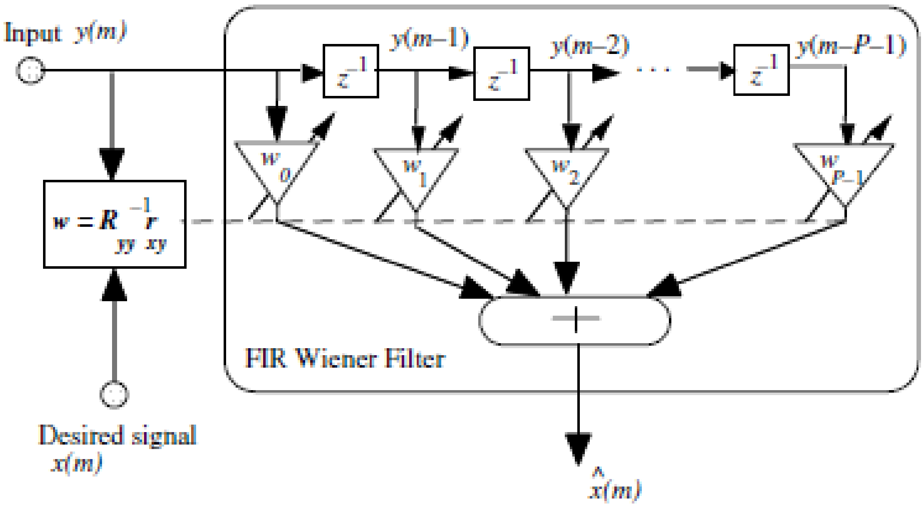


Figure 6.1 Illustration of a Wiener filter structure.

To explore the relation between the filter coefficient vector $\boldsymbol{w}$ and the error signal $e(m)$ we expand Equation (6.2) for N samples of the signals $x(m)$ and $y(m)$:

$$\begin{pmatrix} e(0) \\ e(1) \\ e(2) \\ \vdots \\ e(N-1) \end{pmatrix} = \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-1) \end{pmatrix} - \begin{pmatrix} y(0) & y(-1) & y(-2) & \dots & y(1-P) \\ y(1) & y(0) & y(-1) & \dots & y(2-P) \\ y(2) & y(1) & y(0) & \dots & y(3-P) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y(N-1) & y(N-2) & y(N-3) & \dots & y(N-P) \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{P-1} \end{pmatrix} \tag{6.3}$$

In a compact vector notation this matrix equation may be written as

$$\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{Y}\boldsymbol{w} \tag{6.4}$$

where $\boldsymbol{e}$ is the error vector, $\boldsymbol{x}$ is the desired signal vector, $\boldsymbol{Y}$ is the input signal matrix and $\boldsymbol{Y}\boldsymbol{w} = \hat{\boldsymbol{x}}$ is the Wiener filter output signal vector. It is assumed that the P initial input signal samples $[y(-1), \dots, y(-P-1)]$ are either known or set to zero.

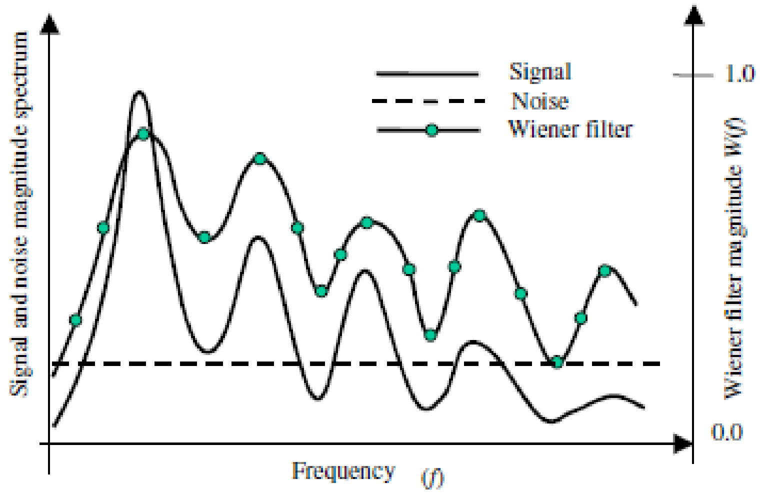


Figure 6.5 Illustration of the variation of Wiener frequency response with signal spectrum for additive white noise. The Wiener filter response broadly follows the signal spectrum.

$$Y(f) = X(f) + N(f) \tag{6.49}$$

where $X(f)$ and $N(f)$ are the signal and noise spectra. For a signal observed in additive random noise, the frequency-domain Wiener filter is obtained as

$$W(f) = \frac{P_{XX}(f)}{P_{XX}(f) + P_{NN}(f)} \tag{6.50}$$

where $P_{XX}(f)$ and $P_{NN}(f)$ are the signal and noise power spectra. Dividing the numerator and the denominator of Equation (6.50) by the noise power spectra $P_{NN}(f)$ and substituting the variable $SNR(f)=P_{XX}(f)/P_{NN}(f)$ yields

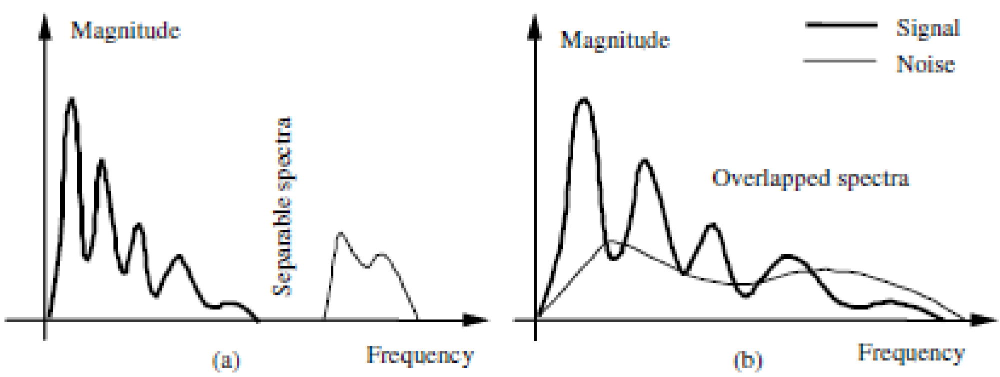


Figure 6.6 Illustration of separability: (a) The signal and noise spectra do not overlap, and the signal can be recovered by a low-pass filter; (b) the signal and noise spectra overlap, and the noise can be reduced but not completely removed.

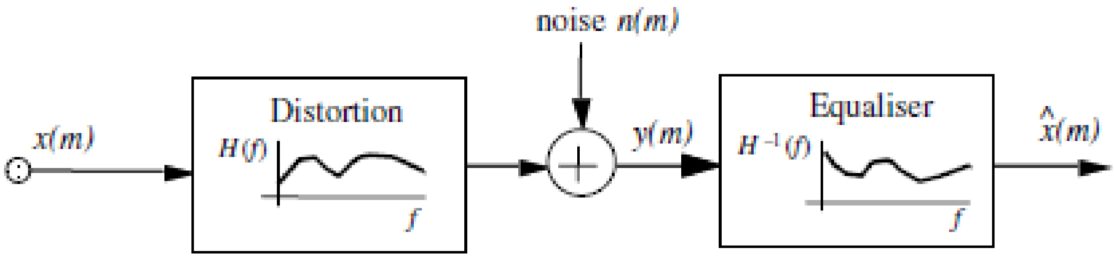


Figure 6.7 Illustration of a channel model followed by an equaliser.

$$W(f)=\frac{P_{XX}(f)H^*(f)}{P_{XX}(f)|H(f)|^2+P_{NN}(f)} \tag{6.62}$$

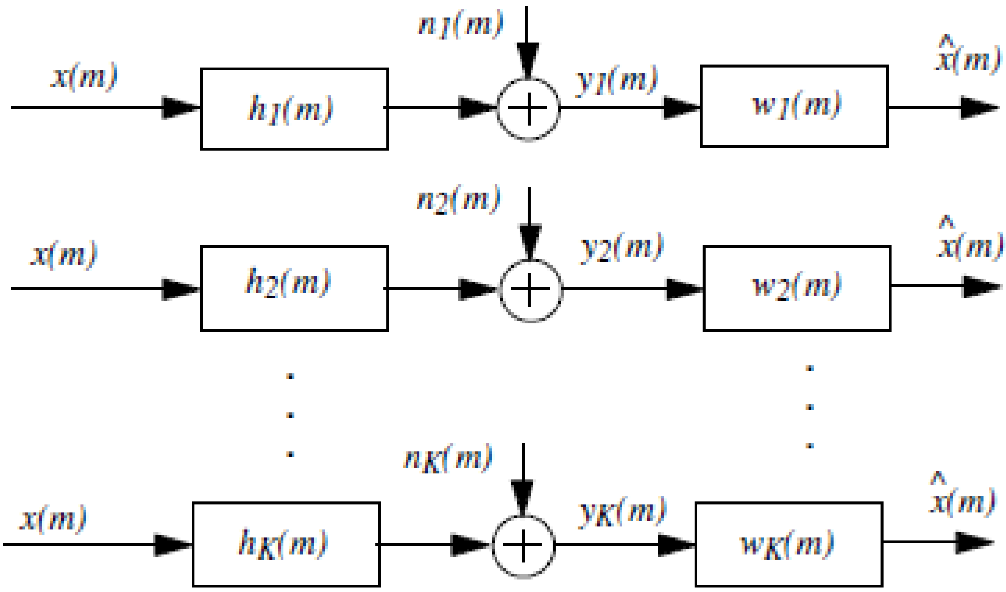


Figure 6.8 Illustration of a multichannel system where Wiener filters are used to time-align the signals from different channels.

$$\begin{aligned} e(m) &= y_2(m)-\sum_{k=0}^{P-1} w_k y_1(m) \\ &= \left(Ax(m-D)-\sum_{k=0}^{P-1} w_k x(m)\right)+\left(\sum_{k=0}^{P-1} w_k n_1(m)\right)+n_2(m) \end{aligned} \tag{6.65}$$

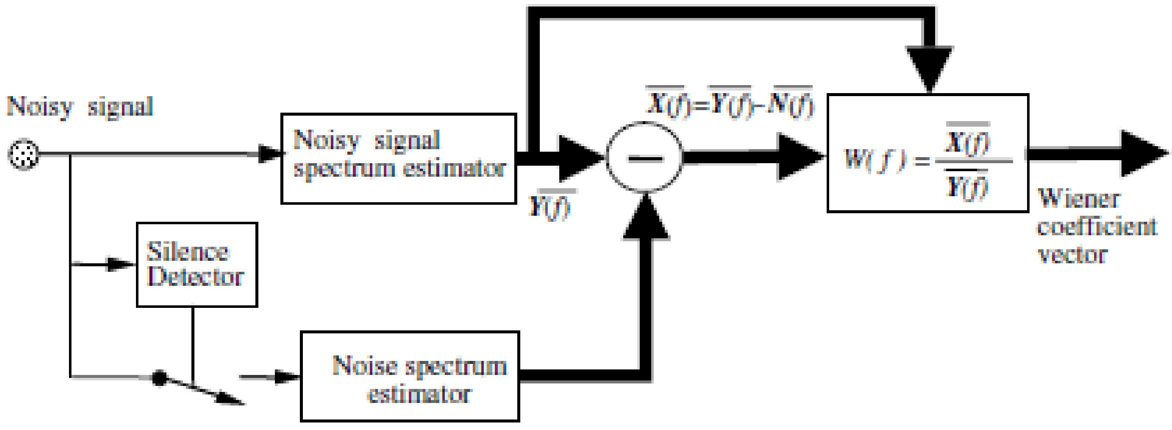


Figure 6.9 Configuration of a system for estimation of frequency Wiener filter.

The Choice of Wiener Filter Order 201

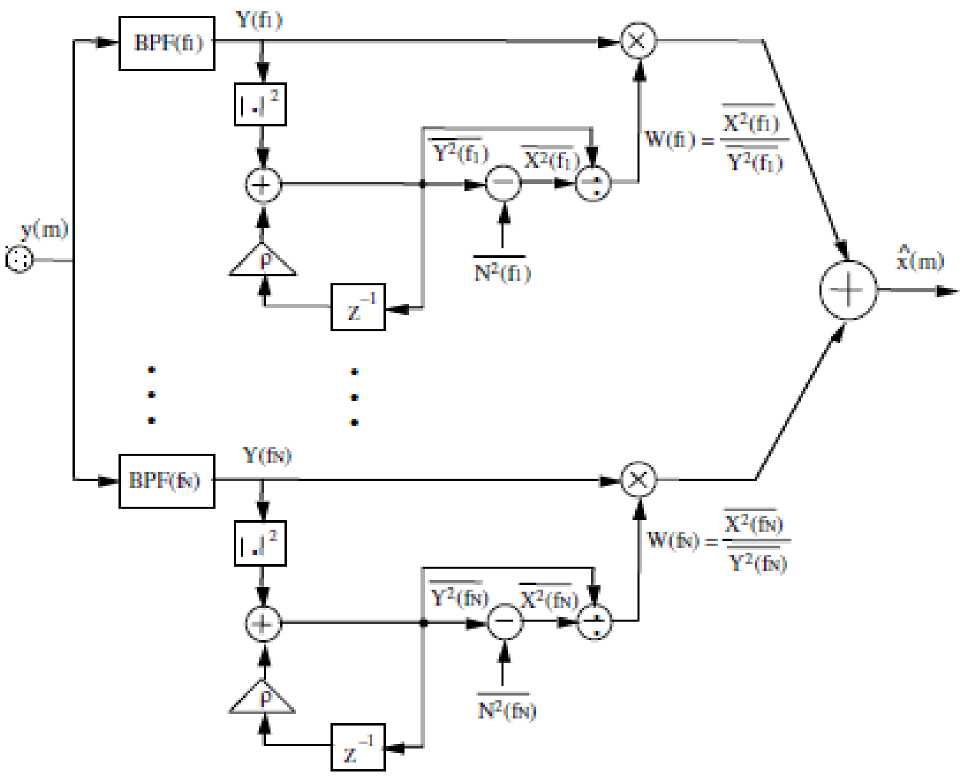
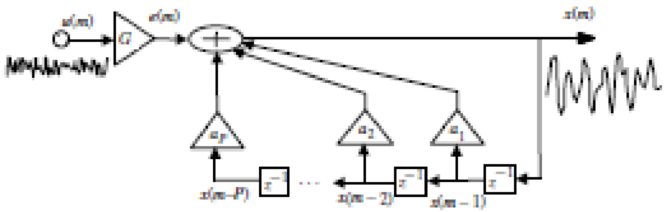


Figure 6.10 A filter-bank implementation of a Wiener filter.

Other Filters

Chapter 7 Adaptive Filter Page 225 to 246

8



LINEAR PREDICTION MODELS

Linear prediction modelling is used in a diverse area of applications, such as data forecasting, speech coding, video coding, speech recognition, model-based spectral analysis, model-based interpolation, signal restoration, and impulse/step event detection. In the statistical literature, linear prediction models are often referred to as autoregressive (AR) processes. In this chapter, we introduce the theory of linear prediction modelling and consider efficient methods for the computation of predictor coefficients. We study the forward, backward and lattice predictors, and consider various methods for the formulation and calculation of predictor coefficients, including the least square error and maximum a posteriori methods. For the modelling of signals with a quasi-periodic structure, such as voiced speech, an extended linear predictor that simultaneously utilizes the short and long-term correlation structures is introduced. We study sub-band linear predictors that are particularly useful for sub-band processing of noisy signals. Finally, the application of linear prediction in enhancement of noisy speech is considered. Further applications of linear prediction models in this book are in Chapter 11 on the interpolation of a sequence of lost samples, and in Chapters 12 and 13 on the detection and removal of impulsive noise and transient noise pulses.

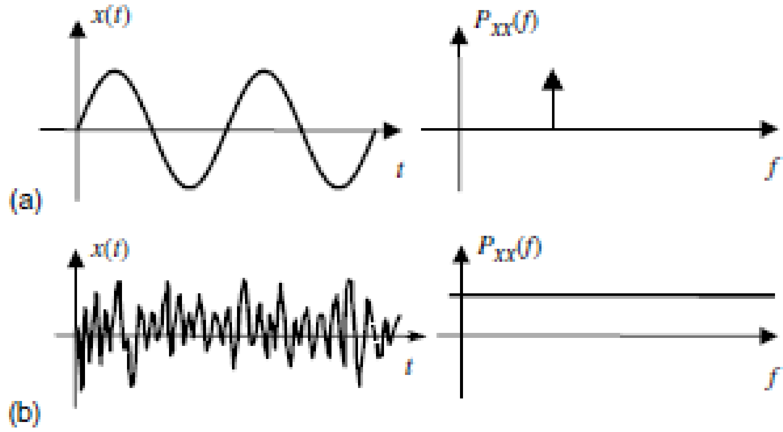


Figure 8.1 The concentration or spread of power in frequency indicates the predictable or random character of a signal: (a) a predictable signal; (b) a random signal.

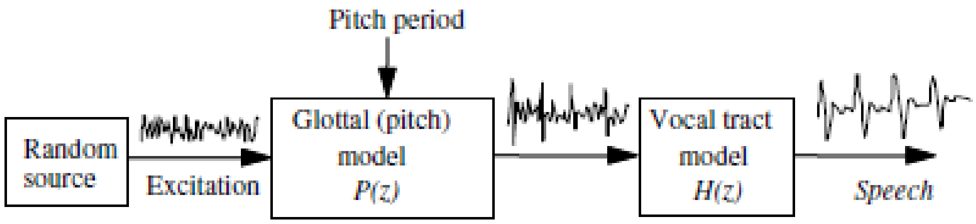


Figure 8.2 A source–filter model of speech production.

The pitch filter models the vibrations of the glottal cords, and generates a sequence of quasi-periodic excitation pulses for voiced sounds as shown in Figure 8.2. The pitch filter model is also termed the "long-term predictor" since it models the correlation of each sample with the samples a pitch period away. The main source of correlation and power in speech is the vocal tract. The vocal tract is modelled by a linear predictor model, which is also termed the "short-term predictor", because it models the correlation of each sample with the few preceding samples. In this section, we study the short-term linear prediction model. In Section 8.3, the predictor model is extended to include long-term pitch period correlations.

A linear predictor model forecasts the amplitude of a signal at time m, $x(m)$, using a linearly weighted combination of P past samples $[x(m-1), x(m-2), ..., x(m-P)]$ as

$$\hat{x}(m) = \sum_{k=1}^{P} a_k x(m-k) \tag{8.1}$$

where the integer variable m is the discrete time index, $\hat{x}(m)$ is the prediction of $x(m)$, and a_k are the predictor coefficients. A block-diagram implementation of the predictor of Equation (8.1) is illustrated in Figure 8.3.

The prediction error $e(m)$, defined as the difference between the actual sample value $x(m)$ and its predicted value $\hat{x}(m)$, is given by

$$\begin{aligned} e(m) &= x(m) - \hat{x}(m) \\ &= x(m) - \sum_{k=1}^{P} a_k x(m-k) \end{aligned} \tag{8.2}$$

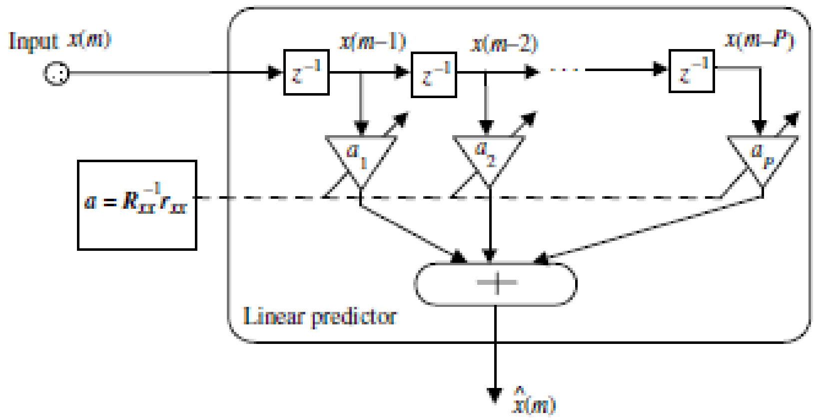


Figure 8.3 Block-diagram illustration of a linear predictor.

For information-bearing signals, the prediction error $e(m)$ may be regarded as the information, or the innovation, content of the sample $x(m)$. From Equation (8.2) a signal generated, or modelled, by a linear predictor can be described by the following feedback equation

$$x(m) = \sum_{k=1}^{P} a_k x(m-k) + e(m) \tag{8.3}$$

Figure 8.4 illustrates a linear predictor model of a signal $x(m)$. In this model, the random input excitation (i.e. the prediction error) is $e(m)=Gu(m)$, where $u(m)$ is a zero-mean, unit-variance random signal, and G, a gain term, is the square root of the variance of $e(m)$:

$$G = \left(\mathcal{E}[e^2(m)]\right)^{1/2} \tag{8.4}$$

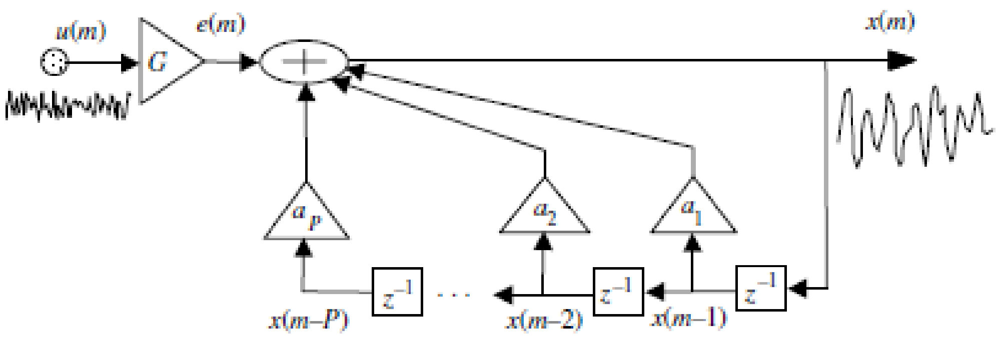


Figure 8.4 Illustration of a signal generated by a linear predictive model.

8.6.1 Frequency-Domain Signal Restoration Using Prediction Models

The following algorithm is a frequency-domain implementation of the linear prediction model-based restoration of a signal observed in additive white noise.

Initialisation: Set the initial signal estimate to noisy signal $\hat{\boldsymbol{x}}_0 = \boldsymbol{y}$,
For iterations $i = 0, 1, \ldots$

Step 1 Estimate the predictor parameter vector $\hat{\boldsymbol{a}}_i$:

$$\hat{\boldsymbol{a}}_i(\hat{\boldsymbol{x}}_i) = \left(\hat{\boldsymbol{X}}_i^{\mathrm{T}} \hat{\boldsymbol{X}}_i\right)^{-1} \left(\hat{\boldsymbol{X}}_i^{\mathrm{T}} \hat{\boldsymbol{x}}_i\right) \tag{8.92}$$

Step 2 Calculate an estimate of the model gain G using the Parseval's theorem:

$$\frac{1}{N}\sum_{f=0}^{N-1}\frac{\hat{G}^2}{\left|1-\sum_{k=1}^{P}\hat{a}_{k,i}e^{-j2\pi fk/N}\right|^2}=\sum_{m=0}^{N-1}y^2(m)-N\hat{\sigma}_n^2 \tag{8.93}$$

where $\hat{a}_{k,i}$ are the coefficient estimates at iteration i, and $N\hat{\sigma}_n^2$ is the energy of white noise over N samples.

Step 3 Calculate an estimate of the power spectrum of speech model:

$$\hat{P}_{X_iX_i}(f)=\frac{\hat{G}^2}{\left|1-\sum_{k=1}^{P}\hat{a}_{k,i}e^{-j2\pi fk/N}\right|^2} \tag{8.94}$$

Step 4 Calculate the Wiener filter frequency response:

$$\hat{W}_i(f)=\frac{\hat{P}_{X_iX_i}(f)}{\hat{P}_{X_iX_i}(f)+\hat{P}_{N_iN_i}(f)} \tag{8.95}$$

where $\hat{P}_{N_iN_i}(f)=\hat{\sigma}_n^2$ is an estimate of the noise power spectrum.

Step 5 Filter the magnitude spectrum of the noisy speech as

$$\hat{X}_{i+1}(f)=\hat{W}_i(f)Y(f) \tag{8.96}$$

Restore the time domain signal $\hat{x}_{i+1}$ by combining $\hat{X}_{i+1}(f)$ with the phase of noisy signal and the complex signal to time domain.

Step 6 Goto step 1 and repeat until convergence, or for a specified number of iterations.

Figure 8.13 illustrates a block diagram configuration of a Wiener filter using a linear prediction estimate of the signal spectrum. Figure 8.14 illustrates the result of an iterative restoration of the spectrum of a noisy speech signal.

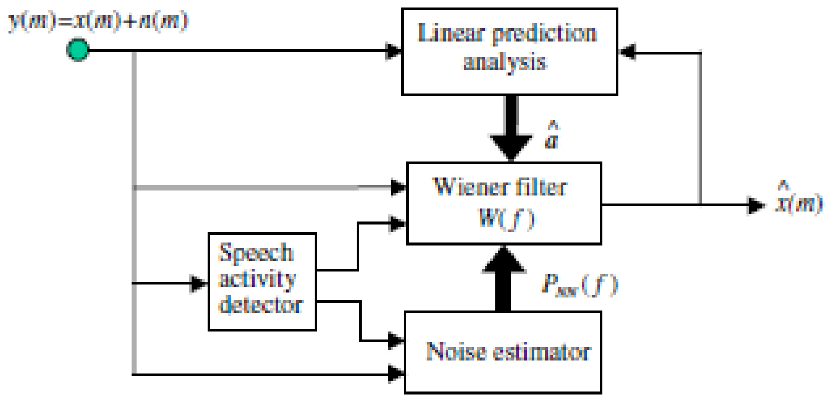


Figure 8.13 Iterative signal restoration based on linear prediction model of speech.

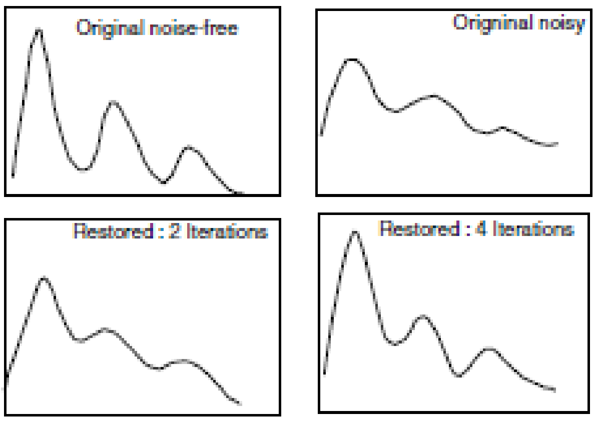


Figure 8.14 Illustration of restoration of a noisy signal with iterative linear prediction based method.

$$W_k(f) = \frac{P_{X,k}(f)}{P_{Y,k}(f)} = \frac{g_{X,k}^2}{|A_{X,k}(f)|^2} \frac{|A_{Y,k}(f)|^2}{g_{Y,k}^2} \tag{8.98}$$

where $P_{X,k}(f)$ and $P_{Y,k}(f)$ are the power spectra of the clean signal and the noisy signal for the k^{th} subband respectively. From Equation (8.98) the square-root Wiener filter is given by

$$W_k^{1/2}(f) = \frac{g_{X,k}}{|A_{X,k}(f)|} \frac{|A_{Y,k}(f)|}{g_{Y,k}} \tag{8.99}$$

The linear prediction Wiener filter of Equation (8.99) can be implemented in the time domain with a cascade of a linear predictor of the clean signal, followed by an inverse predictor filter of the noisy signal as expressed by the following relations (see Figure 8.15):

$$z_k(m) = \sum_{i=1}^{P} a_{Xk}(i) z_k(m-i) + \frac{g_X}{g_Y} y_k(m) \tag{8.100}$$

$$\hat{x}_k(m) = \sum_{i=0}^{P} a_{Yk}(i) z_k(m-i) \tag{8.101}$$

where $\hat{x}_k(m)$ is the restored estimate of $x_k(m)$ the clean speech signal and $z_k(m)$ is an intermediate signal.

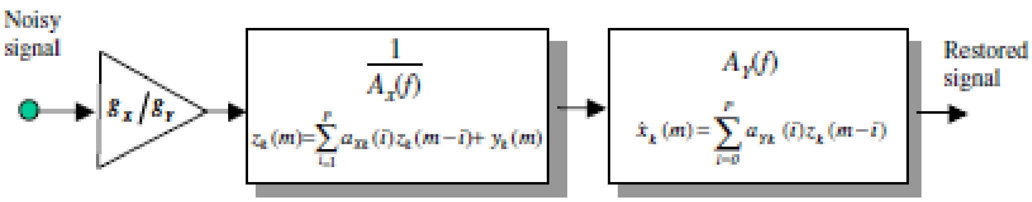


Figure 8.15 A cascade implementation of the LP squared-root Wiener filter.

9.1 Power Spectrum and Correlation

The power spectrum of a signal gives the distribution of the signal power among various frequencies. The power spectrum is the Fourier transform of the correlation function, and reveals information on the correlation structure of the signal. The strength of the Fourier transform in signal analysis and pattern recognition is its ability to reveal spectral structures that may be used to characterise a signal. This is illustrated in Figure 9.1 for the two extreme cases of a sine wave and a purely random signal. For a periodic signal, the power is concentrated in extremely narrow bands of frequencies, indicating the existence of structure and the predictable character of the signal. In the case of a pure sine wave as shown in Figure 9.1(a) the signal power is concentrated in one frequency. For a purely random signal as shown in Figure 9.1(b) the signal power is spread equally in the frequency domain, indicating the lack of structure in the signal.

In general, the more correlated or predictable a signal, the more concentrated its power spectrum, and conversely the more random or unpredictable a signal, the more spread its power spectrum. Therefore the power spectrum of a signal can be used to deduce the existence of repetitive structures or correlated patterns in the signal process. Such information is crucial in detection, decision making and estimation problems, and in systems analysis.

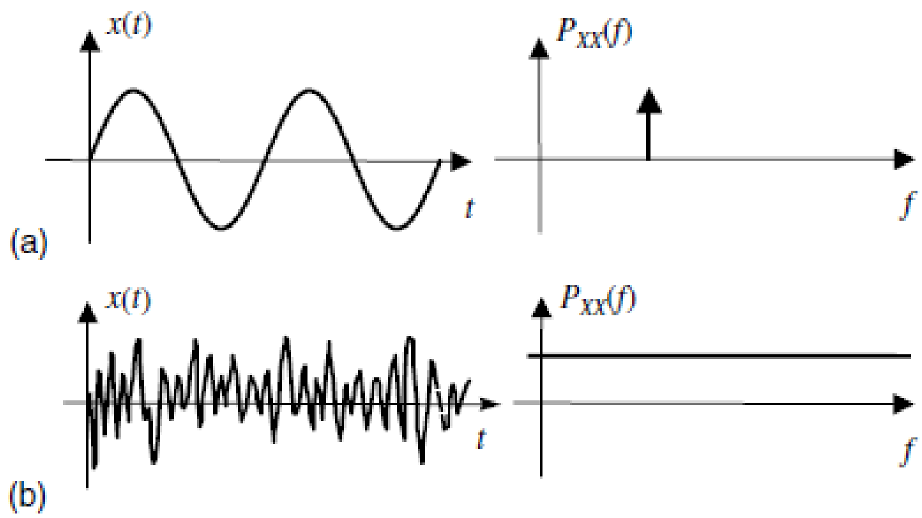


Figure 9.1 The concentration/spread of power in frequency indicates the correlated or random character of a signal: (a) a predictable signal, (b) a random signal.

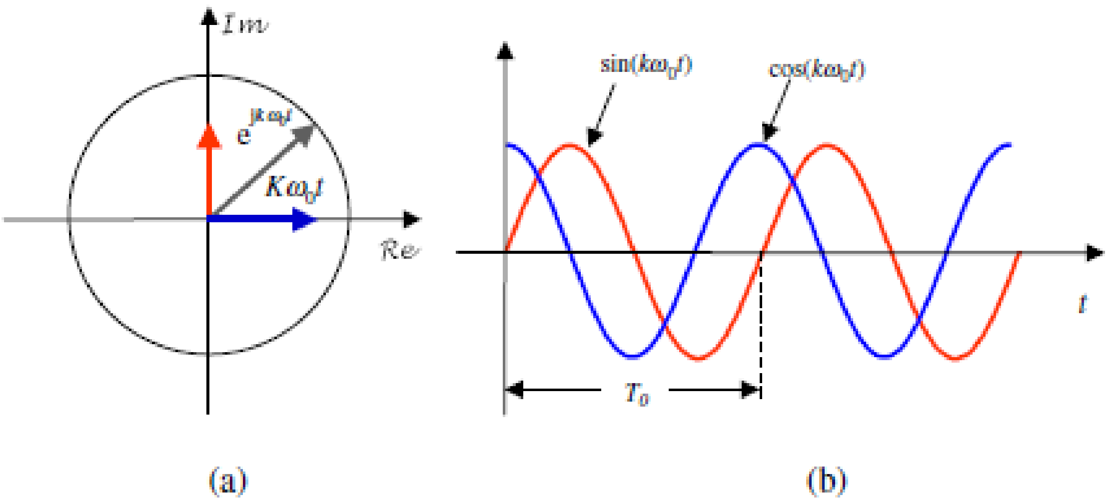


Figure 9.2 Fourier basis functions: (a) real and imaginary parts of a complex sinusoid, (b) vector representation of a complex exponential.

9.2 Fourier Series: Representation of Periodic Signals

The following three sinusoidal functions form the *basis functions* for the Fourier analysis:

$$x_1(t) = \cos\omega_0 t \tag{9.1}$$

$$x_2(t) = \sin\omega_0 t \tag{9.2}$$

$$x_3(t) = \cos\omega_0 t + j\sin\omega_0 t = e^{j\omega_0 t} \tag{9.3}$$

Associated with the complex exponential function $e^{j\omega_0 t}$ is a set of harmonically related complex exponentials of the form

$$[1, e^{\pm j\omega_0 t}, e^{\pm j2\omega_0 t}, e^{\pm j3\omega_0 t}, \ldots] \tag{9.5}$$

The set of exponential signals in Equation (9.5) are periodic with a fundamental frequency $\omega_0=2\pi/T_0=2\pi F_0$, where T_0 is the period and F_0 is the fundamental frequency. These signals form the set of *basis functions* for the Fourier analysis. Any linear combination of these signals of the form

$$\sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t} \tag{9.6}$$

is also periodic with a period T_0. Conversely any periodic signal $x(t)$ can be synthesised from a linear combination of harmonically related exponentials. The Fourier series representation of a periodic signal is given by the following synthesis and analysis equations:

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t} \quad k = \cdots \quad -1,0,1,\ldots \quad \text{(synthesis equation)} \tag{9.7}$$

$$c_k = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} x(t) e^{-jk\omega_0 t}\, dt \quad k = \ldots -1,0,1,\ldots \quad \text{(analysis equation)} \tag{9.8}$$

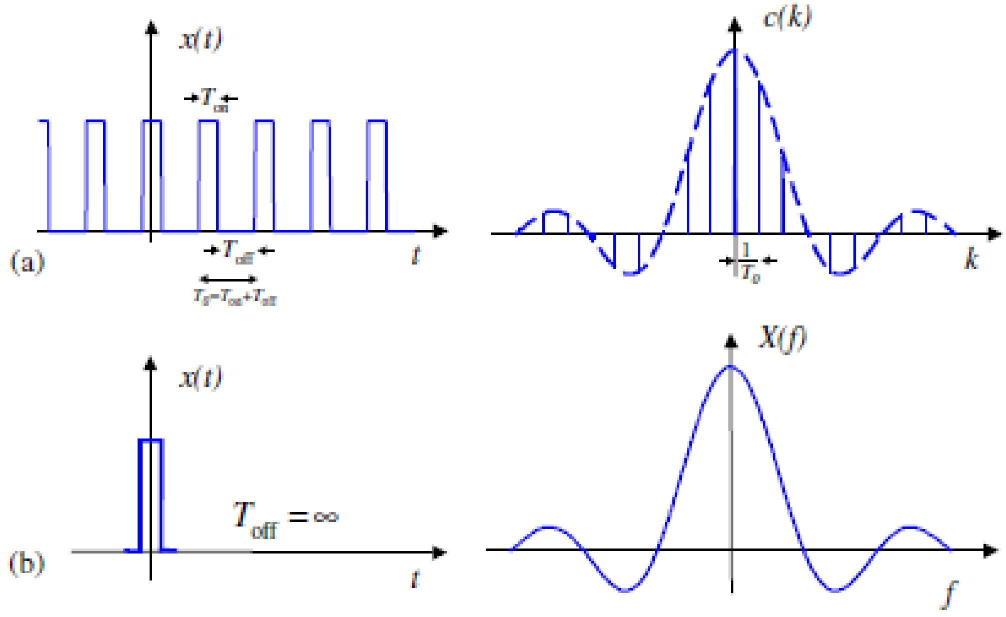


Figure 9.3 (a) A periodic pulse train and its line spectrum. (b) A single pulse from the periodic train in (a) with an imagined "off" duration of infinity; its spectrum is the envelope of the spectrum of the periodic signal in (a).

The Fourier synthesis and analysis equations for aperiodic signals, the so-called *Fourier transform pair*, are given by

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \tag{9.9}$$

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \tag{9.10}$$

Note from Equation (9.10), that $X(f)$ *may be interpreted as a measure of the correlation of the signal x(t) and the complex sinusoid* $e^{-j2\pi ft}$.

The condition for existence and computability of the Fourier transform integral of a signal $x(t)$ is that the signal must have finite energy:

$$\int_{-\infty}^{\infty} |x(t)|^2 dt < \infty \tag{9.11}$$

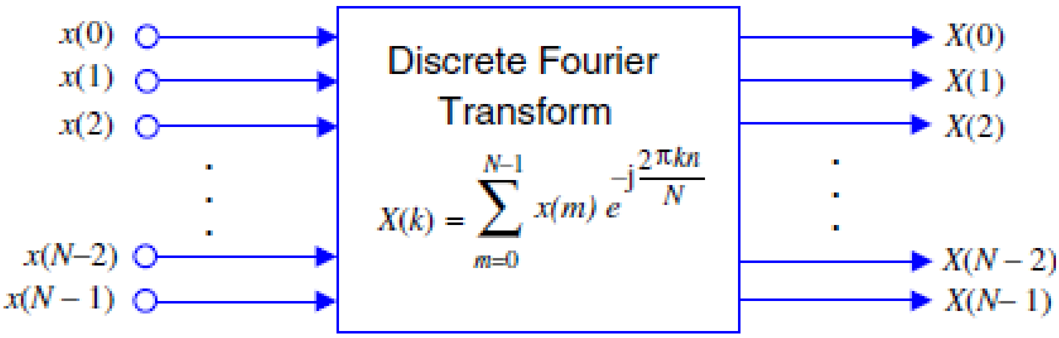


Figure 9.4 Illustration of the DFT as a parallel-input, parallel-output processor.

9.3.1 Discrete Fourier Transform (DFT)

For a finite-duration, discrete-time signal $x(m)$ of length N samples, the discrete Fourier transform (DFT) is defined as N uniformly spaced spectral samples

$$X(k) = \sum_{m=0}^{N-1} x(m) e^{-j(2\pi/N)mk}, \qquad k = 0, \ldots, N-1 \qquad (9.12)$$

(see Figure9.4). The inverse discrete Fourier transform (IDFT) is given by

$$x(m) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j(2\pi/N)mk}, \qquad m = 0, \ldots, N-1 \qquad (9.13)$$

9.3.3 Energy-Spectral Density and Power-Spectral Density

Energy, or power, spectrum analysis is concerned with the distribution of the signal energy or power in the frequency domain. For a deterministic discrete-time signal, the energy-spectral density is defined as

$$|X(f)|^2 = \left| \sum_{m=-\infty}^{\infty} x(m) e^{-j2\pi fm} \right|^2 \qquad (9.15)$$

The energy spectrum of $x(m)$ may be expressed as the Fourier transform of the autocorrelation function of $x(m)$:

$$\begin{aligned} |X(f)|^2 &= X(f) X^*(f) \\ &= \sum_{m=-\infty}^{\infty} r_{xx}(m) e^{-j2\pi fm} \end{aligned} \qquad (9.16)$$

Page 293 to 310 Several methods to perform spectral estimation

INTERPOLATION

Interpolation is the estimation of the unknown, or the lost, samples of a signal using a weighted average of a number of known samples at the neighbourhood points. Interpolators are used in various forms in most signal processing and decision making systems. Applications of interpolators include conversion of a discrete-time signal to a continuous-time signal, sampling rate conversion in multirate communication systems, low-bit-rate speech coding, up-sampling of a signal for improved graphical representation, and restoration of a sequence of samples irrevocably distorted by transmission errors, impulsive noise, dropouts, etc. This chapter begins with a study of the basic concept of ideal interpolation of a band-limited signal, a simple model for the effects of a number of missing samples, and the factors that affect the interpolation process. The classical approach to interpolation is to construct a polynomial that passes through the known samples. In Section 10.2, a general form of polynomial interpolation and its special forms, Lagrange, Newton, Hermite and cubic spline interpolators, are considered. Optimal interpolators utilise predictive and statistical models of the signal process. In Section 10.3, a number of model-based interpolation methods are considered. These methods include maximum a posteriori interpolation, and least square error interpolation based on an autoregressive model. Finally, we consider time–frequency interpolation, and interpolation through searching an adaptive signal codebook for the best-matching signal.

10.1.1 Interpolation of a Sampled Signal

A common application of interpolation is the reconstruction of a continuous-time signal $x(t)$ from a discrete-time signal $x(m)$. The condition for the recovery of a continuous-time signal from its samples is given by the Nyquist sampling theorem. The Nyquist theorem states that a band-limited signal, with a highest frequency content of F_c (Hz), can be reconstructed from its samples *if* the sampling speed is greater than $2F_c$ samples per second. Consider a band-limited continuous-time signal $x(t)$, sampled at a rate of F_s samples per second. The discrete-time signal $x(m)$ may be expressed as the following product:

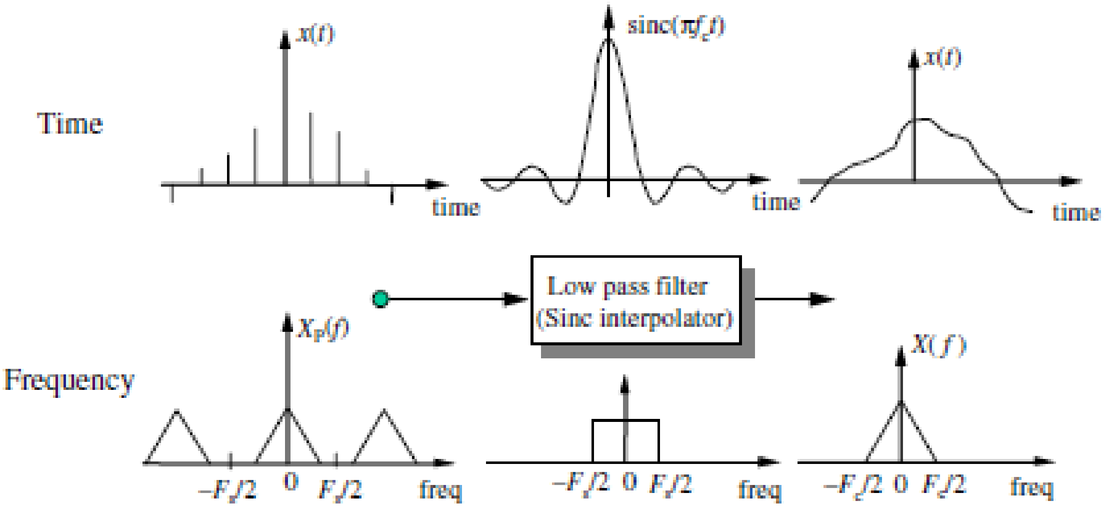


Figure 10.1 Reconstruction of a continuous-time signal from its samples. In frequency domain interpolation is equivalent to low-pass filtering.

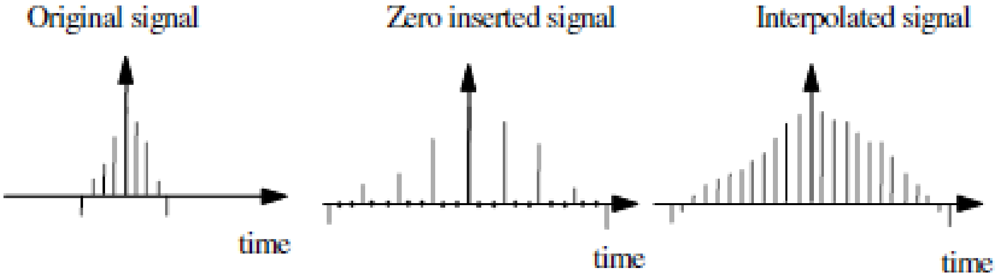


Figure 10.2 Illustration of up-sampling by a factor of 3 using a two-stage process of zero-insertion and digital low-pass filtering.

$$x(m)=x(t)\,p(t)=\sum_{m=-\infty}^{\infty} x(t)\delta(t-mT_s) \tag{10.1}$$

where $p(t)=\Sigma\delta(t-mT_s)$ is the sampling function and $T_s=1/F_s$ is the sampling interval. Taking the Fourier transform of Equation (10.1), it can be shown that the spectrum of the sampled signal is given by

$$X_s(f)=X(f)*P(f)=\sum_{k=-\infty}^{\infty} X(f+kf_s) \tag{10.2}$$

where $X(f)$ and $P(f)$ are the spectra of the signal $x(t)$ and the sampling function $p(t)$ respectively, and * denotes the convolution operation. Equation (10.2), illustrated in Figure 10.1, states that the spectrum of a sampled signal is composed of the original base-band spectrum $X(f)$ and the repetitions or images of $X(f)$ spaced uniformly at frequency intervals of $F_s=1/T_s$. When the sampling frequency is above the Nyquist rate, the base-band spectrum $X(f)$ is not overlapped by its images $X(f\pm kF_s)$, and the original signal can be recovered by a low-pass filter as shown in Figure 10.1. Hence the ideal interpolator of a band-limited discrete-time signal is an ideal low-pass filter with a sinc impulse response. The recovery of a continuous-time signal through sinc interpolation can be expressed as

$$x(t)=\sum_{m=-\infty}^{\infty} x(m)T_s f_c \,\mathrm{sinc}\left[\pi f_c (t-mT_s)\right] \tag{10.3}$$

In practice, the sampling rate F_s should be sufficiently greater than $2F_c$, say $2.5F_c$, in order to accommodate the transition bandwidth of the interpolating low-pass filter.

10.1.2 Digital Interpolation by a Factor of *I*

Applications of digital interpolators include sampling rate conversion in multirate communication systems and up-sampling for improved graphical representation. To change a sampling rate by a factor of $V=I/D$ (where I and D are integers), the signal is first interpolated by a factor of I, and then the interpolated signal is decimated by a factor of D.

Consider a band-limited discrete-time signal $x(m)$ with a base-band spectrum $X(f)$ as shown in Figure 10.2. The sampling rate can be increased by a factor of I through interpolation of I–1 samples between every two samples of $x(m)$. In the following it is shown that digital interpolation by a factor of I can be achieved through a two-stage process of (a) insertion of I–1 zeros in between every two samples and (b) low-pass filtering of the zero-inserted signal by a filter with a cutoff frequency of $F_s/2I$, where F_s is the sampling rate. Consider the zero-inserted signal $x_z(m)$ obtained by inserting I–1 zeros between every two samples of $x(m)$ and expressed as

$$x_z(m) = \begin{cases} x\left(\dfrac{m}{I}\right), & m = 0, \pm I, \pm 2I, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{10.4}$$

The spectrum of the zero-inserted signal is related to the spectrum of the original discrete-time signal by

$$\begin{aligned} X_z(f) &= \sum_{m=-\infty}^{\infty} x_z(m) e^{-j2\pi f m} \\ &= \sum_{m=-\infty}^{\infty} x(m) e^{-j2\pi f m I} \\ &= X(I.f) \end{aligned} \tag{10.5}$$

Equation (10.5) states that the spectrum of the zero-inserted signal $X_z(f)$ is a frequency-scaled version of the spectrum of the original signal $X(f)$. Figure 10.2 shows that the base-band spectrum of the zero-inserted signal is composed of I repetitions of the based band spectrum of the original signal. The interpolation of the zero-inserted signal is therefore equivalent to filtering out the repetitions of $X(f)$ in the base band of $X_z(f)$, as illustrated in Figure 10.2. Note that to maintain the real-time duration of the signal the

10.1.4 The Factors That Affect Interpolation Accuracy

The interpolation accuracy is affected by a number of factors, the most important of which are as follows:

(a) The predictability, or correlation structure of the signal: as the correlation of successive samples increases, the predictability of a sample from the neighbouring samples increases. In general, interpolation improves with the increasing correlation structure, or equivalently the decreasing bandwidth, of a signal.
(b) The sampling rate: as the sampling rate increases, adjacent samples become more correlated, the redundant information increases, and interpolation improves.
(c) Non-stationary characteristics of the signal: for time-varying signals the available samples some distance in time away from the missing samples may not be relevant because the signal characteristics may have completely changed. This is particularly important in interpolation of a large sequence of samples.
(d) The length of the missing samples: in general, interpolation quality decreases with increasing length of the missing samples.
(e) Finally, interpolation depends on the optimal use of the data and the efficiency of the interpolator.

10.2 Polynomial Interpolation

The classical approach to interpolation is to construct a polynomial interpolator that passes through the known samples. Polynomial interpolators may be formulated in various forms, such as power series, Lagrange interpolation and Newton interpolation. These various forms are mathematically equivalent and can be transformed from one into another. Suppose the data consists of N+1 samples $\{x(t_0), x(t_1), ..., x(t_N)\}$, where $x(t_n)$ denotes the amplitude of the signal $x(t)$ at time t_n. The polynomial of order N that passes through the N+1 known samples is unique (Figure 10.4) and may be written in power series form as

$$\hat{x}(t) = p_N(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + \cdots + a_N t^N \qquad (10.13)$$

where $P_N(t)$ is a polynomial of order N, and the a_k are the polynomial coefficients. From Equation (10.13), and a set of N+1 known samples, a

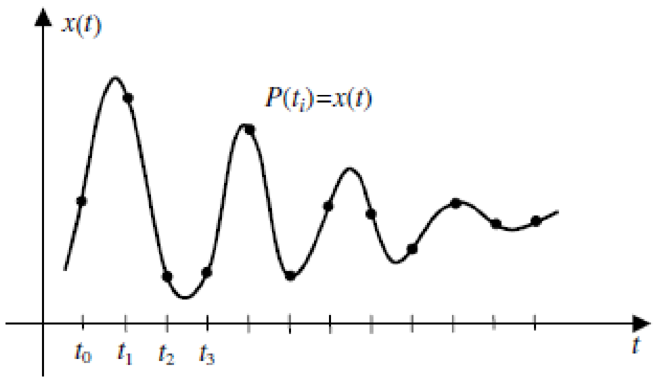


Figure 10.4 Illustration of an Interpolation curve through a number of samples.

$$
\begin{aligned}
x(t_0) &= a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 + \cdots + a_N t_0^N \\
x(t_1) &= a_0 + a_1 t_1 + a_2 t_1^2 + a_3 t_1^3 + \cdots + a_N t_1^N \\
\vdots \quad & \quad \vdots \quad \quad \ddots \\
x(t_N) &= a_0 + a_1 t_N + a_2 t_N^2 + a_3 t_N^3 + \cdots + a_N t_N^N
\end{aligned} \tag{10.14}
$$

From Equation (10.14). the polynomial coefficients are given by

$$
\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix} = \begin{pmatrix} 1 & t_0 & t_0^2 & t_0^3 & \cdots & t_0^N \\ 1 & t_1 & t_1^2 & t_1^3 & \cdots & t_1^N \\ 1 & t_2 & t_2^2 & t_2^3 & \cdots & t_2^N \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_N & t_N^2 & t_N^3 & \cdots & t_N^N \end{pmatrix}^{-1} \begin{pmatrix} x(t_0) \\ x(t_1) \\ x(t_2) \\ \vdots \\ x(t_N) \end{pmatrix} \tag{10.15}
$$

Page 327 to 350 Other interpolation methods

SPECTRAL SUBTRACTION

Spectral subtraction is a method for restoration of the power spectrum or the magnitude spectrum of a signal observed in additive noise, through subtraction of an estimate of the average noise spectrum from the noisy signal spectrum. The noise spectrum is usually estimated, and updated, from the periods when the signal is absent and only the noise is present. The assumption is that the noise is a stationary or a slowly varying process, and that the noise spectrum does not change significantly in-between the update periods. For restoration of time-domain signals, an estimate of the instantaneous magnitude spectrum is combined with the phase of the noisy signal, and then transformed via an inverse discrete Fourier transform to the time domain. In terms of computational complexity, spectral subtraction is relatively inexpensive. However, owing to random variations of noise, spectral subtraction can result in negative estimates of the short-time magnitude or power spectrum. The magnitude and power spectrum are non-negative variables, and any negative estimates of these variables should be mapped into non-negative values. This non-linear rectification process distorts the distribution of the restored signal. The processing distortion becomes more noticeable as the signal-to-noise ratio decreases. In this chapter, we study spectral subtraction, and the different methods of reducing and removing the processing distortions.

11.1 Spectral Subtraction

In applications where, in addition to the noisy signal, the noise is accessible on a separate channel, it may be possible to retrieve the signal by subtracting an estimate of the noise from the noisy signal. For example, the adaptive noise canceller of Section 1.3.1 takes as the inputs the noise and the noisy signal, and outputs an estimate of the clean signal. However, in many applications, such as at the receiver of a noisy communication channel, the only signal that is available is the noisy signal. In these situations, it is not possible to cancel out the random noise, but it may be possible to reduce the *average effects* of the noise on the signal spectrum. The effect of additive noise on the magnitude spectrum of a signal is to increase the mean and the variance of the spectrum as illustrated in Figure 11.1. The increase in the variance of the signal spectrum results from the random fluctuations of the noise, and cannot be cancelled out. The increase in the mean of the signal spectrum can be removed by subtraction of an estimate of the mean of the noise spectrum from the noisy signal spectrum. The noisy signal model in the time domain is given by

$$y(m) = x(m) + n(m) \tag{11.1}$$

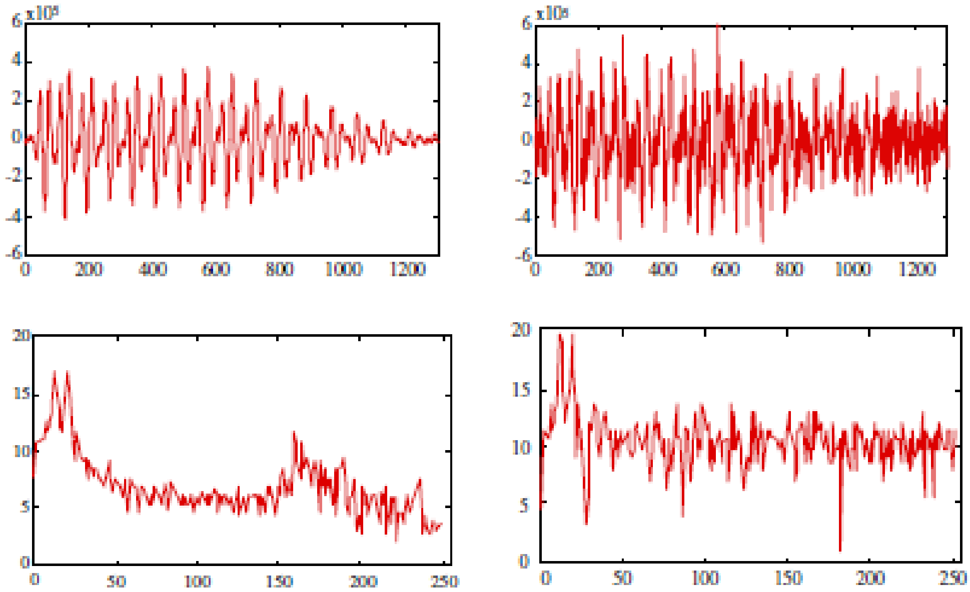


Figure 11.1 Illustrations of the effect of noise on a signal in the time and the frequency domains.

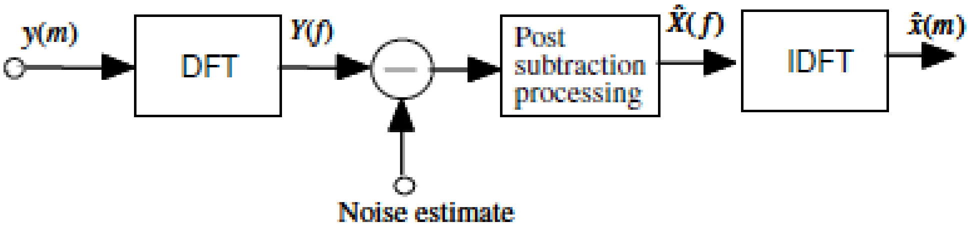


Figure 11.2 A block diagram illustration of spectral subtraction.

in Equation (11.5) controls the amount of noise subtracted from the noisy signal. For full noise subtraction, α=1 and for over-subtraction α>1. The time-averaged noise spectrum is obtained from the periods when the signal is absent and only the noise is present as

$$\overline{|N(f)|^b} = \frac{1}{K}\sum_{i=0}^{K-1} |N_i(f)|^b \tag{11.6}$$

In Equation (11.6), $|N_i(f)|$ is the spectrum of the i^{th} noise frame, and it is assumed that there are K frames in a noise-only period, where K is a variable. Alternatively, the averaged noise spectrum can be obtained as the output of a first order digital low-pass filter as

$$\overline{|N_i(f)|^b} = \rho\,\overline{|N_{i-1}(f)|^b} + (1-\rho)\,|N_i(f)|^b \tag{11.7}$$

11.1.1 Power Spectrum Subtraction

The power spectrum subtraction, or squared-magnitude spectrum subtraction, is defined by the following equation:

$$|\hat{X}(f)|^2 = |Y(f)|^2 - \overline{|N(f)|^2} \tag{11.11}$$

where it is assumed that α, the subtraction factor in Equation (11.5), is unity. We denote the power spectrum by $\mathcal{E}[|X(f)|^2]$, the time-averaged power spectrum by $\overline{|X(f)|^2}$ and the *instantaneous* power spectrum by $|X(f)|^2$. By expanding the instantaneous power spectrum of the noisy

signal $|Y(f)|^2$, and grouping the appropriate terms, Equation (11.11) may be rewritten as

$$|\hat{X}(f)|^2 = |X(f)|^2 + \underbrace{\left(|N(f)|^2 - \overline{|N(f)|^2}\right)}_{\text{Noise variations}} + \underbrace{X^*(f)N(f) + X(f)N^*(f)}_{\text{Cross products}} \tag{11.12}$$

Taking the expectations of both sides of Equation (11.12), and assuming that the signal and the noise are uncorrelated ergodic processes, we have

$$\mathcal{E}[|\hat{X}(f)|^2] = \mathcal{E}[|X(f)|^2] \tag{11.13}$$

11.1.3 Spectral Subtraction Filter: Relation to Wiener Filters

The spectral subtraction equation can be expressed as the product of the noisy signal spectrum and the frequency response of a spectral subtraction filter as

$$\begin{aligned} |\hat{X}(f)|^2 &= |Y(f)|^2 - \overline{|N(f)|^2} \\ &= H(f)|Y(f)|^2 \end{aligned} \tag{11.16}$$

where $H(f)$, the frequency response of the spectral subtraction filter, is defined as

$$\begin{aligned} H(f) &= 1 - \frac{\overline{|N(f)|^2}}{|Y(f)|^2} \\ &= \frac{|Y(f)|^2 - \overline{|N(f)|^2}}{|Y(f)|^2} \end{aligned} \tag{11.17}$$

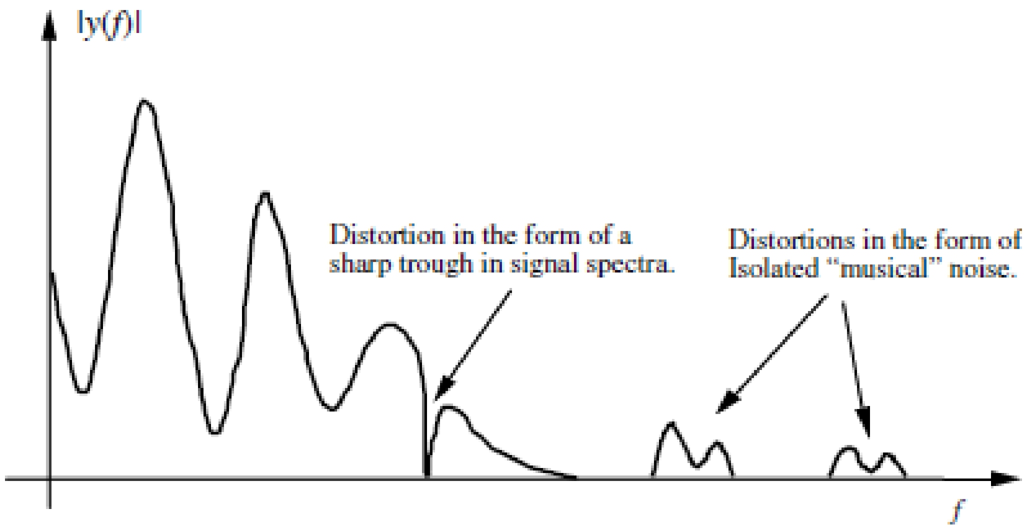


Figure 11.3 Illustration of distortions that may result from spectral subtraction.

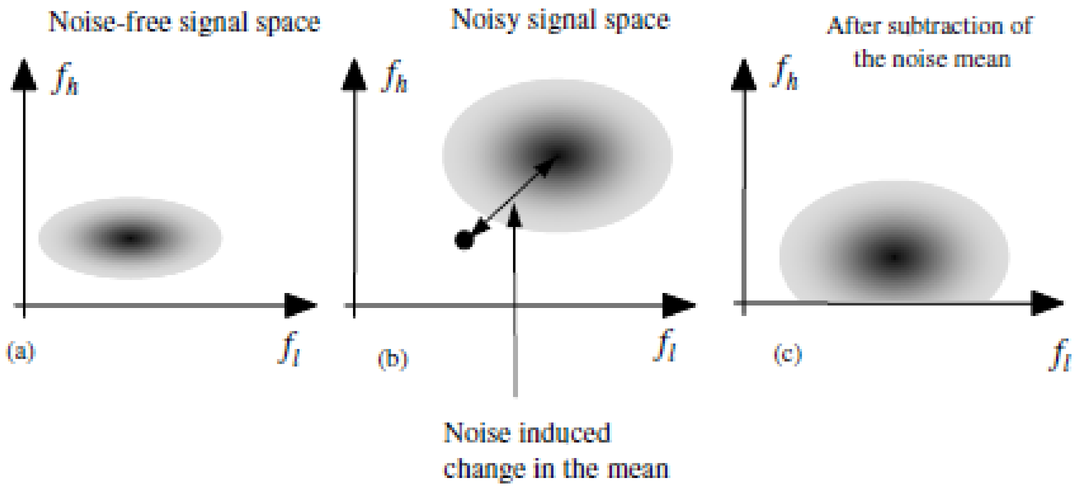


Figure 11.4 Illustration of the distorting effect of spectral subtraction on the space of the magnitude spectrum of a signal.

11.2.2 Reducing the Noise Variance

The distortions that result from spectral subtraction are due to the variations of the noise spectrum. In Section 9.2 we considered the methods of reducing the variance of the estimate of a power spectrum. For a white noise process with variance σ_n^2, it can be shown that the variance of the DFT spectrum of the noise $N(f)$ is given by

$$\mathrm{Var}[|N(f)|^2] \approx P_{NN}^2(f) = \sigma_n^4 \tag{11.20}$$

and the variance of the running average of K independent spectral components is

$$\mathrm{Var}\left[\frac{1}{K}\sum_{i=0}^{K-1}|N_i(f)|^2\right] \approx \frac{1}{K}P_{NN}^2(f) \approx \frac{1}{K}\sigma_n^4 \tag{11.21}$$

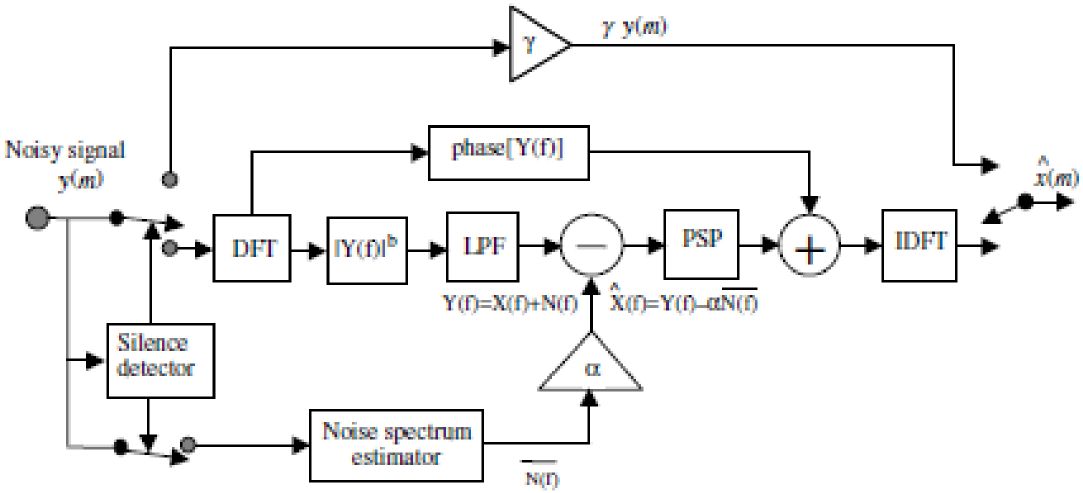


Figure 11.7 Block diagram configuration of a spectral subtraction system. PSP = post spectral subtraction processing.

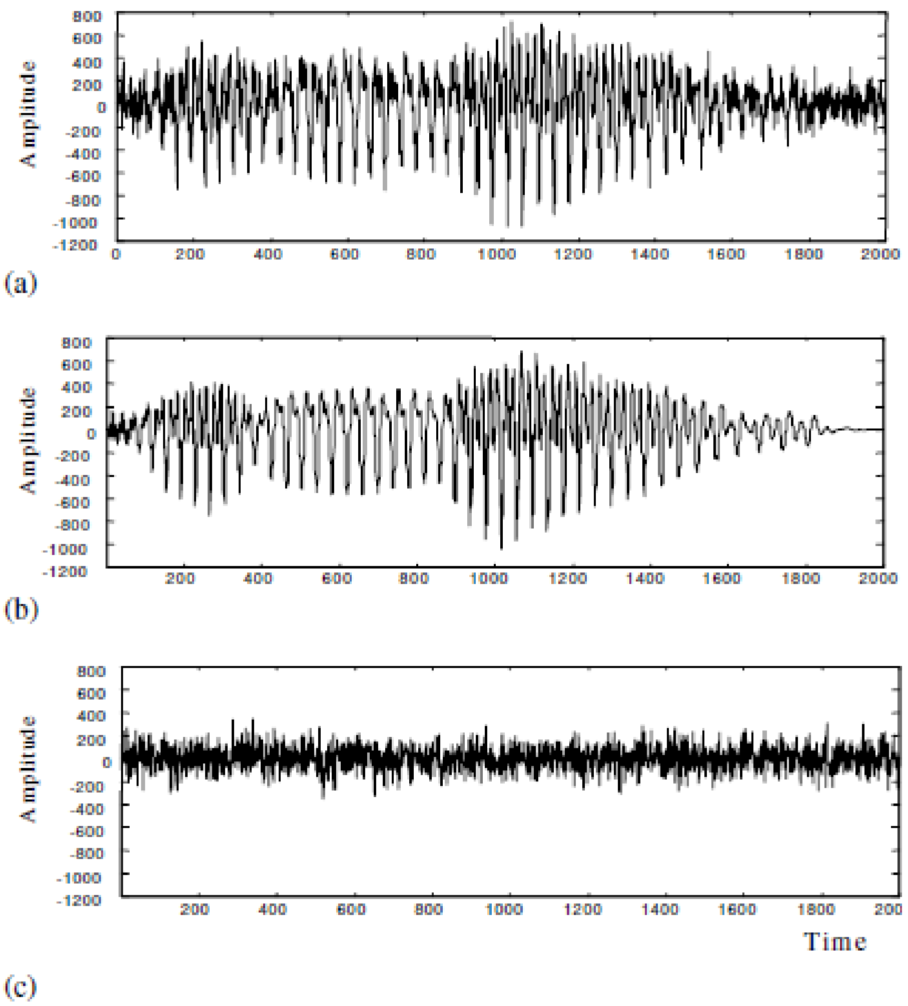


Figure 11.9 (a) A noisy signal. (b) Restored signal after spectral subtraction. (c) Noise estimate obtained by subtracting (b) from (a).

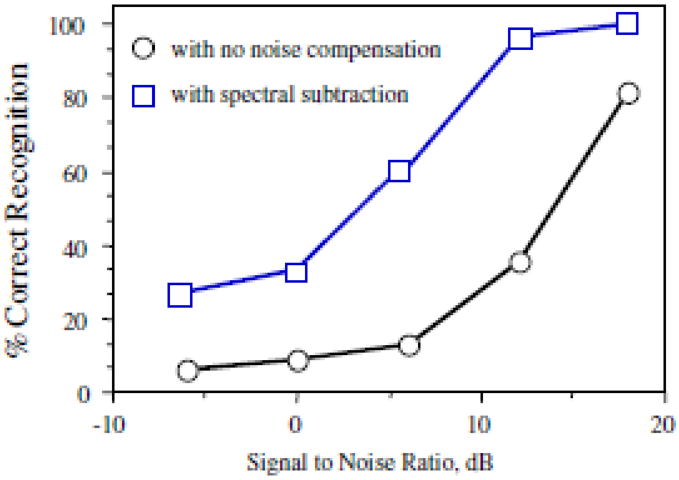


Figure 11.10 The effect of spectral subtraction in improving speech recognition (for a spoken digit data base) in the presence of helicopter noise.

IMPULSIVE NOISE

Impulsive noise consists of relatively short duration "on/off" noise pulses, caused by a variety of sources, such as switching noise, adverse channel environments in a communication system, dropouts or surface degradation of audio recordings, clicks from computer keyboards, etc. An impulsive noise filter can be used for enhancing the quality and intelligibility of noisy signals, and for achieving robustness in pattern recognition and adaptive control systems. This chapter begins with a study of the frequency/time characteristics of impulsive noise, and then proceeds to consider several methods for statistical modelling of an impulsive noise process. The classical method for removal of impulsive noise is the median filter. However, the median filter often results in some signal degradation. For optimal performance, an impulsive noise removal system should utilise (a) the distinct features of the noise and the signal in the time and/or frequency domains, (b) the statistics of the signal and the noise processes, and (c) a model of the physiology of the signal and noise generation. We describe a model-based system that detects each impulsive noise, and then proceeds to replace the samples obliterated by an impulse. We also consider some methods for introducing robustness to impulsive noise in parameter estimation.

12.1 Impulsive Noise

In this section, first the mathematical concepts of an analog and a digital impulse are introduced, and then the various forms of real impulsive noise in communication systems are considered.

The mathematical concept of an analog impulse is illustrated in Figure 12.1. Consider the unit-area pulse $p(t)$ shown in Figure 12.1(a). As the pulse width Δ tends to zero, the pulse tends to an impulse. The impulse function shown in Figure 12.1(b) is defined as a pulse with an infinitesimal time width as

$$\delta(t) = \underset{\Delta \to 0}{\text{limit}}\ p(t) = \begin{cases} 1/\Delta, & |t| \le \Delta/2 \\ 0, & |t| > \Delta/2 \end{cases} \tag{12.1}$$

The integral of the impulse function is given by

$$\int_{-\infty}^{\infty} \delta(t)\, dt = \Delta \times \frac{1}{\Delta} = 1 \tag{12.2}$$

The Fourier transform of the impulse function is obtained as

$$\Delta(f) = \int_{-\infty}^{\infty} \delta(t) e^{-j2\pi ft} dt = e^{0} = 1 \tag{12.3}$$

where f is the frequency variable. The impulse function is used as a *test function* to obtain the impulse response of a system. This is because as

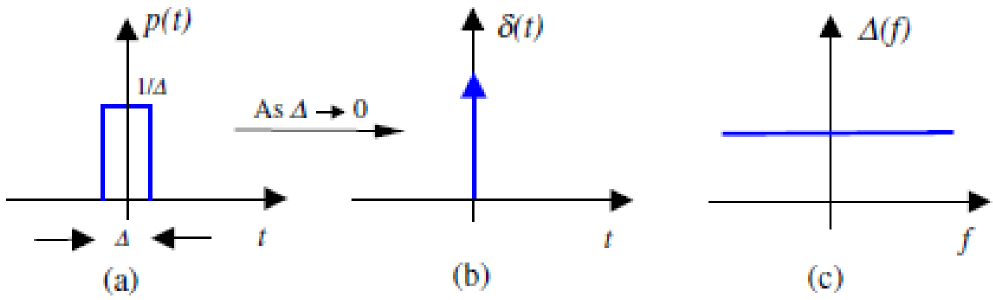


Figure 12.1 (a) A unit-area pulse, (b) The pulse becomes an impulse as $\Delta \to 0$, (c) The spectrum of the impulse function.

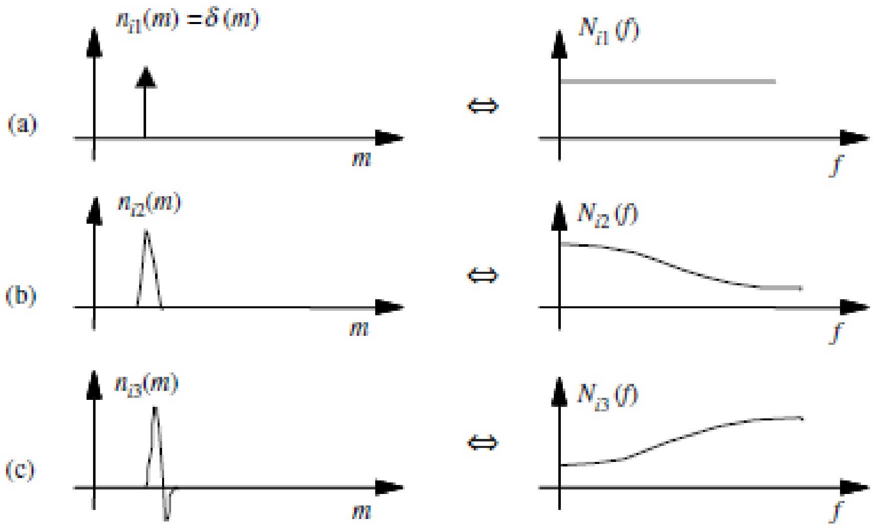


Figure 12.2 Time and frequency sketches of (a) an ideal impulse, and (b) and (c) short-duration pulses.

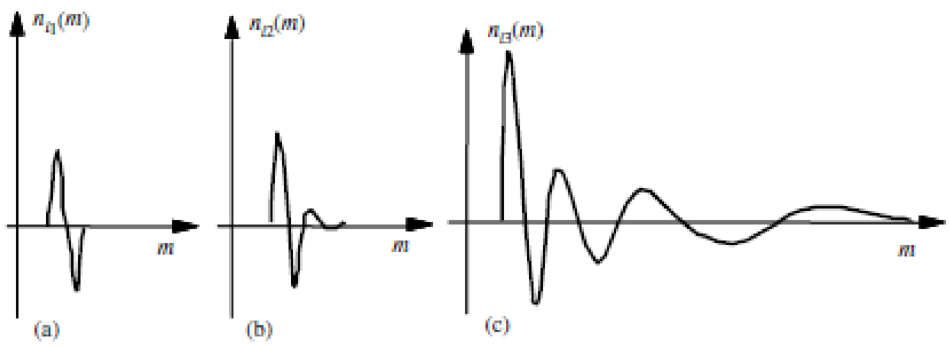


Figure 12.3 Illustration of variations of the impulse response of a non-linear system with increasing amplitude of the impulse.

12.2.1 Bernoulli–Gaussian Model of Impulsive Noise

In a Bernoulli-Gaussian model of an impulsive noise process, the random time of occurrence of the impulses is modelled by a binary Bernoulli process $b(m)$ and the amplitude of the impulses is modelled by a Gaussian

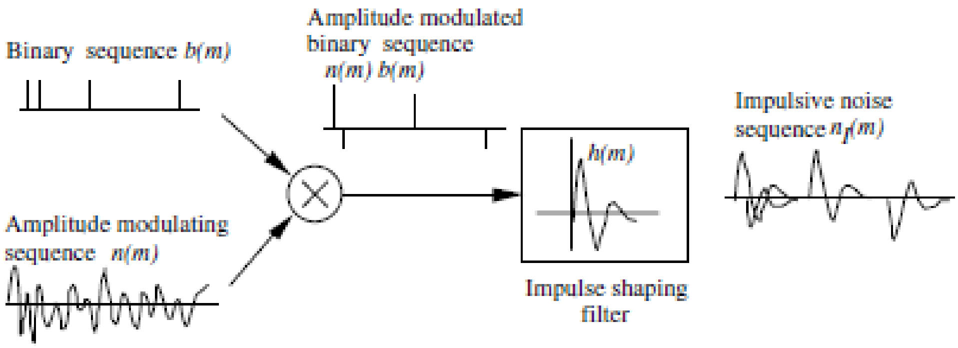


Figure 12.4 Illustration of an impulsive noise model as the output of a filter excited by an amplitude-modulated binary sequence.

12.2.4 Signal to Impulsive Noise Ratio

For impulsive noise the average signal to impulsive noise ratio, averaged over an entire noise sequence including the time instances when the impulses are absent, depends on two parameters: (a) the average power of each impulsive noise, and (b) the rate of occurrence of impulsive noise. Let P_{impulse} denote the average power of each impulse, and P_{signal} the signal power. We may define a "local" time-varying signal to impulsive noise ratio as

$$SINR(m) = \frac{P_{\text{signal}}(m)}{P_{\text{impulse}}\, b(m)} \qquad (12.21)$$

The average signal to impulsive noise ratio, assuming that the parameter α is the fraction of signal samples contaminated by impulsive noise, can be defined as

$$SINR = \frac{P_{\text{signal}}}{\alpha\, P_{\text{impulse}}} \qquad (12.22)$$

Note that from Equation (12.22), for a given signal power, there are many pair of values of α and P_{Impulse} that can yield the same average SINR.

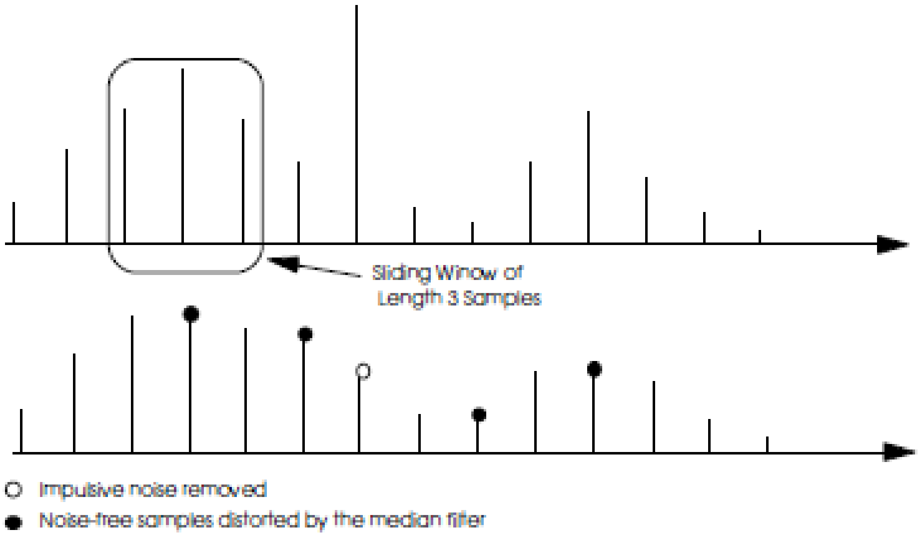


Figure 12.7 Input and output of a median filter. Note that in addition to suppressing the impulsive outlier, the filter also distorts some genuine signal components.

12.3 Median Filters

The classical approach to removal of impulsive noise is the median filter. The median of a set of samples $\{x(m)\}$ is a member of the set $x_{med}(m)$ such that; half the population of the set are larger than $x_{med}(m)$ and half are smaller than $x_{med}(m)$. Hence the median of a set of samples is obtained by sorting the samples in the ascending or descending order, and then selecting the mid-value. In median filtering, a window of predetermined length slides sequentially over the signal, and the mid-sample within the window is replaced by the median of all the samples that are inside the window, as illustrated in Figure 12.7.

The output $\hat{x}(m)$ of a median filter with input $y(m)$ and a median window of length $2K+1$ samples is given by

$$\begin{aligned}\hat{x}(m) &= y_{med}(m) \\ &= \text{median}\left[y(m-K),\ldots,y(m),\ldots,y(m+K)\right]\end{aligned} \tag{12.23}$$

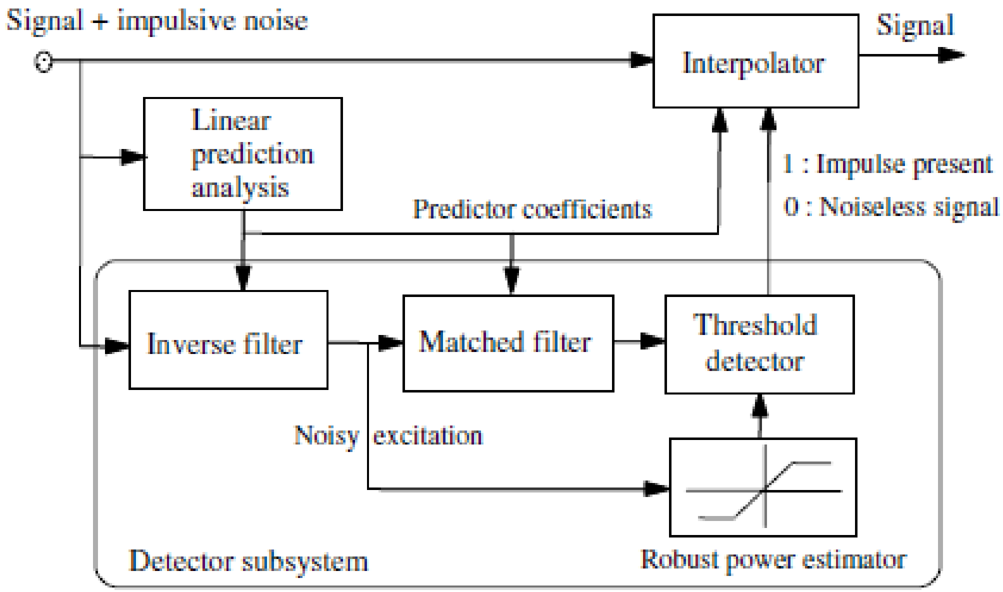


Figure 12.8 Configuration of an impulsive noise removal system incorporating a detector and interpolator subsystems.

(a) The scale of the signal amplitude is reduced to almost that of the original excitation signal, whereas the scale of the noise amplitude remains unchanged or increases.
(b) The signal is decorrelated, whereas the impulsive noise is smeared and transformed to a scaled version of the impulse response of the inverse filter.

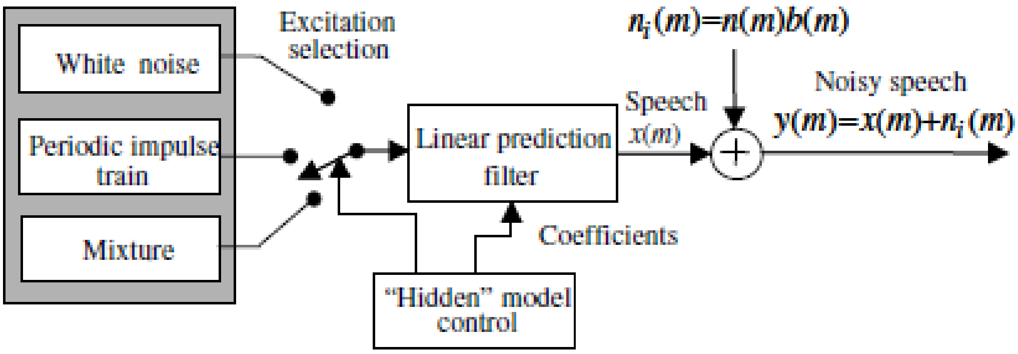


Figure 12.9 Noisy speech model. The signal is modelled by a linear predictor. Impulsive noise is modelled as an amplitude-modulated binary-state process.

12.4.2 Analysis of Improvement in Noise Detectability

In the following, the improvement in noise detectability that results from inverse filtering is analysed. Using Equation (12.25), we can rewrite a noisy signal model as

$$\begin{aligned} y(m) &= x(m)+n_i(m) \\ &= \sum_{k=1}^{P} a_k\, x(m-k)+e(m)+n_i(m) \end{aligned} \tag{12.26}$$

where $y(m)$, $x(m)$ and $n_i(m)$ are the noisy signal, the signal and the noise respectively. Using an estimate $\hat{\boldsymbol{a}}$ of the predictor coefficient vector $\boldsymbol{a}$, the noisy signal $y(m)$ can be inverse-filtered and transformed to the noisy excitation signal $v(m)$ as

$$\begin{aligned} v(m) &= y(m)-\sum_{k=1}^{P} \hat{a}_k\, y(m-k) \\ &= x(m)+n_i(m)-\sum_{k=1}^{P} (a_k-\tilde{a}_k)[x(m-k)+n_i(m-k)] \end{aligned} \tag{12.27}$$

where $\tilde{a}_k$ is the error in the estimate of the predictor coefficient. Using Equation (12.25) Equation (12.27) can be rewritten in the following form:

$$v(m) = e(m)+n_i(m)+\sum_{k=1}^{P} \tilde{a}_k x(m-k)-\sum_{k=1}^{P} \hat{a}_k n_i(m-k) \tag{12.28}$$

The improvement resulting from the inverse filter can be formulated as follows. The impulsive noise to signal ratio for the noisy signal is given by

$$\frac{\text{impulsive noise power}}{\text{signal power}} = \frac{\mathcal{E}[n_i^2(m)]}{\mathcal{E}[x^2(m)]} \tag{12.29}$$

where $\mathcal{E}[\cdot]$ is the expectation operator. Note that in impulsive noise detection, the signal of interest is the impulsive noise to be detected from the accompanying signal. Assuming that the dominant noise term in the noisy excitation signal $v(m)$ is the impulse $n_i(m)$, the impulsive noise to excitation signal ratio is given by

$$\frac{\text{impulsive noise power}}{\text{excitation power}} = \frac{\mathcal{E}[n_i^2(m)]}{\mathcal{E}[e^2(m)]} \tag{12.30}$$

The overall gain in impulsive noise to signal ratio is obtained, by dividing Equations (12.29) and (12.30), as

$$\frac{\mathcal{E}[x^2(m)]}{\mathcal{E}[e^2(m)]} = \text{gain} \tag{12.31}$$

12.5 Robust Parameter Estimation

In Figure 12.8, the threshold used for detection of impulsive noise from the excitation signal is derived from a nonlinear robust estimate of the excitation power. In this section, we consider robust estimation of a parameter, such as the signal power, in the presence of impulsive noise.

A *robust* estimator is one that is not over-sensitive to deviations of the input signal from the assumed distribution. In a robust estimator, an input sample with unusually large amplitude has only a limited effect on the estimation results. Most signal processing algorithms developed for adaptive filtering, speech recognition, speech coding, etc. are based on the assumption that the signal and the noise are Gaussian-distributed, and employ a mean square distance measure as the optimality criterion. The mean square error criterion is sensitive to non-Gaussian events such as impulsive noise. A large impulsive noise in a signal can substantially

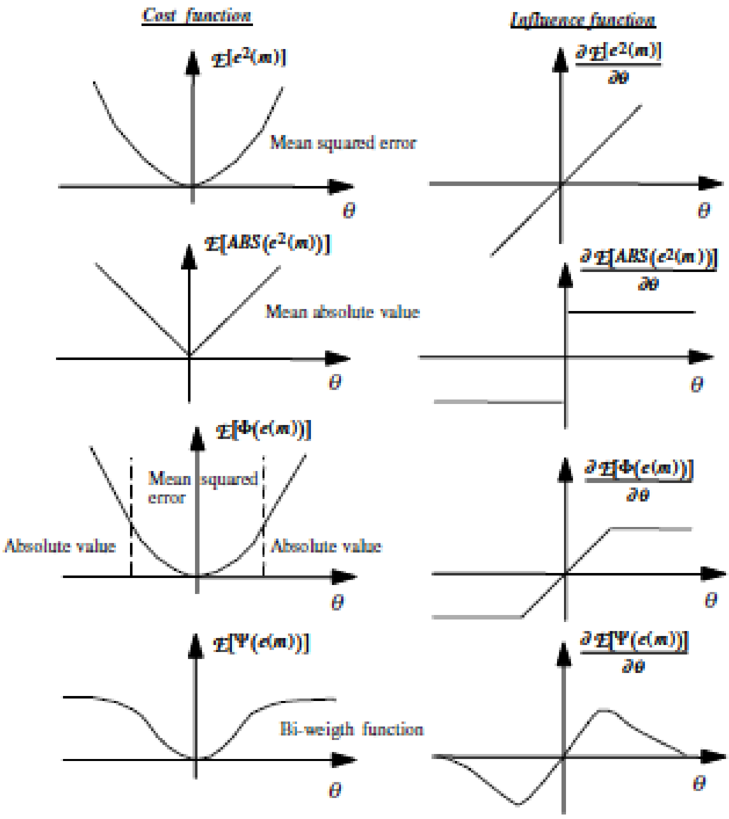


Figure 12.11 Illustration of a number of cost of error functions and the corresponding influence functions.

TRANSIENT NOISE PULSES

Transient noise pulses differ from the short-duration impulsive noise studied in the previous chapter, in that they have a longer duration and a relatively higher proportion of low-frequency energy content, and usually occur less frequently than impulsive noise. The sources of transient noise pulses are varied, and may be electromagnetic, acoustic or due to physical defects in the recording medium. Examples of transient noise pulses include switching noise in telephony, noise pulses due to adverse radio transmission environments, noise pulses due to on/off switching of nearby electric devices, scratches and defects on damaged records, click sounds from a computer keyboard, etc. The noise pulse removal methods considered in this chapter are based on the observation that transient noise pulses can be regarded as the response of the communication channel, or the playback system, to an impulse. In this chapter, we study the characteristics of transient noise pulses and consider a template-based method, a linear predictive model and a hidden Markov model for the modelling and removal of transient noise pulses. The subject of this chapter closely follows that of Chapter 12 on impulsive noise.

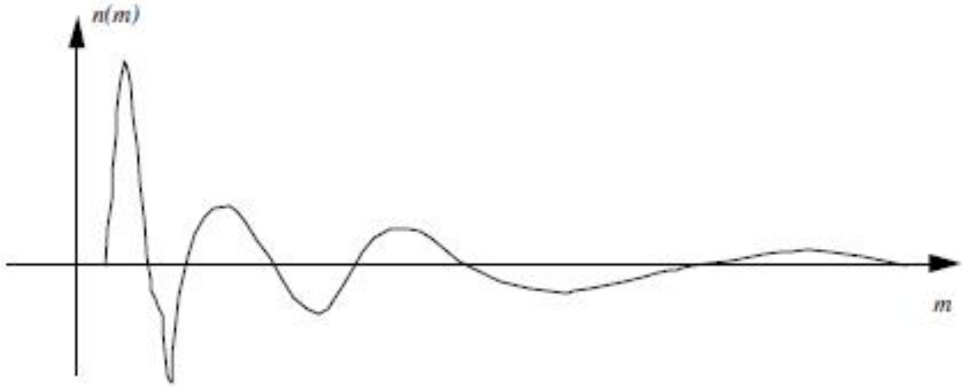
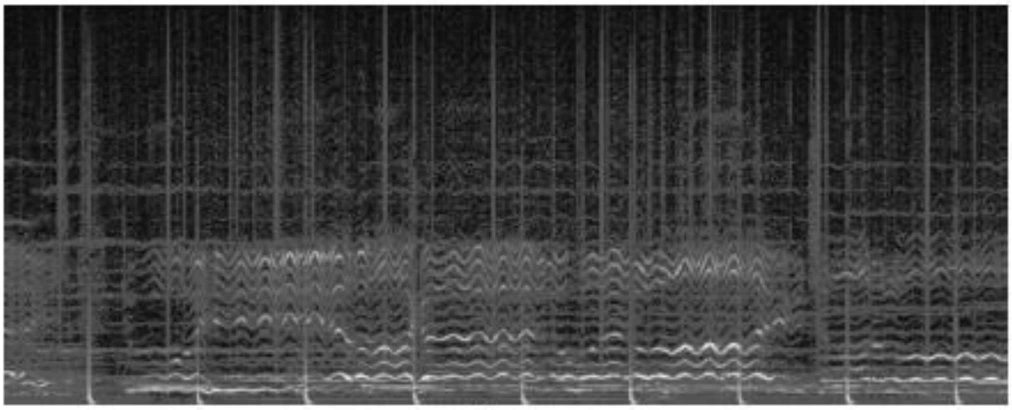


Figure 13.1 The profile of a transient noise pulse from a scratched gramophone record.

(a)



(b)

Figure 13.2 An example of (a) the time-domain waveform and (b) the spectrogram of transient noise scratch pulses in a damaged gramophone record.

13.2 Transient Noise Pulse Models

To a first approximation, a transient noise pulse $n(m)$ can be modelled as the impulse response of a linear time-invariant filter model of the channel as

$$n(m)=\sum_{k} h_k A\delta(m-k)=Ah_m \tag{13.1}$$

where A is the amplitude of the driving impulse and h_k is the channel impulse response. A burst of overlapping, or closely spaced, noise pulses can be modelled as the response of a channel to a sequence of impulses as

$$n(m)=\sum_{k} h_k \sum_{j} A_j\delta\big((m-T_j)-k\big)=\sum_{j} A_j h_{m-T_j} \tag{13.2}$$

where it is assumed that the j^{th} transient pulse is due to an impulse of amplitude A_j at time T_j. In practice, a noise model should be able to deal with the statistical variations of a variety of noise and channel types. In this section, we consider three methods for modelling the temporal, spectral and durational characteristics of a transient noise pulse process:

(a) a template-based model;
(b) a linear-predictive model;
(c) a hidden Markov model.

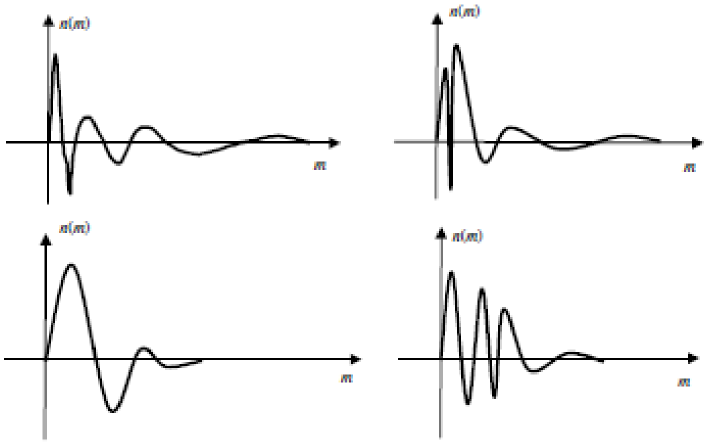


Figure 13.3 A number of prototype transient pulses.

13.2.2 Autoregressive Model of Transient Noise Pulses

Model-based methods have the advantage over template-based methods that overlapped noise pulses can be modelled as the response of the model to a number of closely spaced impulsive inputs. In this section, we consider an autoregressive (AR) model of transient noise pulses. The AR model for a single noise pulse $n(m)$ can be described as

$$n(m)=\sum_{k=1}^{P} c_k n(m-k)+A\delta(m) \tag{13.3}$$

where c_k are the AR model coefficients, and the excitation is an impulse function $\delta(m)$ of amplitude A. A number of closely spaced and overlapping transient noise pulses can be modelled as the response of the AR model to a sequence of impulses:

$$n(m)=\sum_{k=1}^{P} c_k n(m-k)+\sum_{j}^{M} A_j\delta(m-T_j) \tag{13.4}$$

where it is assumed that T_j is the start of the j^{th} pulse in a burst of M excitation pulses.

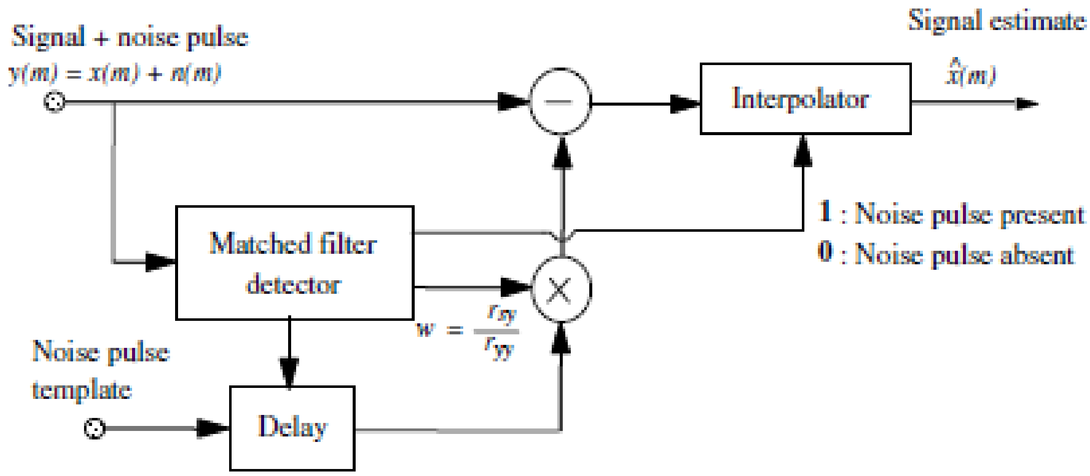
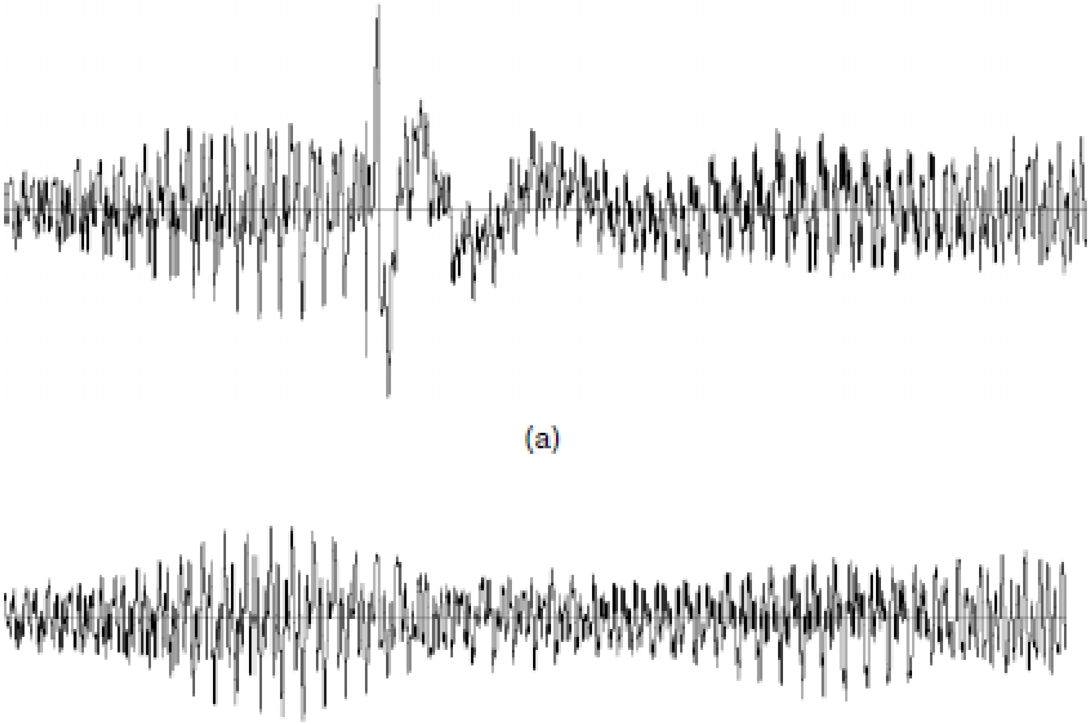


Figure 13.6 Transient noise pulse removal system.

$$n(m)=w\overline{n}(m-D) \tag{13.18}$$



(a)

(b)

Figure 13.7 (a) A signal from an old gramophone record with a scratch noise pulse. (b) The restored signal.

ECHO CANCELLATION

Echo is the repetition of a waveform due to reflection from points where the characteristics of the medium through which the wave propagates changes. Echo is usefully employed in sonar and radar for detection and exploration purposes. In telecommunication, echo can degrade the quality of service, and echo cancellation is an important part of communication systems. The development of echo reduction began in the late 1950s, and continues today as new integrated landline and wireless cellular networks put additional requirement on the performance of echo cancellers. There are two types of echo in communication systems: acoustic echo and telephone line hybrid echo. Acoustic echo results from a feedback path set up between the speaker and the microphone in a mobile phone, hands-free phone, teleconference or hearing aid system. Acoustic echo may be reflected from a multitude of different surfaces, such as walls, ceilings and floors, and travels through different paths. Telephone line echoes result from an impedance mismatch at telephone exchange hybrids where the subscriber's 2-wire line is connected to a 4-wire line. The perceptual effects of an echo depend on the time delay between the incident and reflected waves, the strength of the reflected waves, and the number of paths through which the waves are reflected. Telephone line echoes, and acoustic feedback echoes in teleconference and hearing aid systems, are undesirable and annoying and can be disruptive. In this chapter we study some methods for removing line echo from telephone and data telecommunication systems, and acoustic feedback echoes from microphone–loudspeaker systems.

(a) acoustic echo due to acoustic coupling between the speaker and the microphone in hands-free phones, mobile phones and teleconference systems;
(b) electrical line echo due to mismatch at the hybrid circuit connecting a 2-wire subscriber line to a 4-wire truck line in the public switched telephone network.

In the early days of expansion of telephone networks, the cost of running a 4-wire line from the local exchange to subscribers' premises was considered uneconomical. Hence, at the exchange the 4-wire truck lines are converted to 2-wire subscribers local lines using a 2/4-wire hybrid bridge circuit. At the receiver due to any imbalance between the 4/2-wire bridge circuit, some of the signal energy of the 4-wire circuit is bounced back

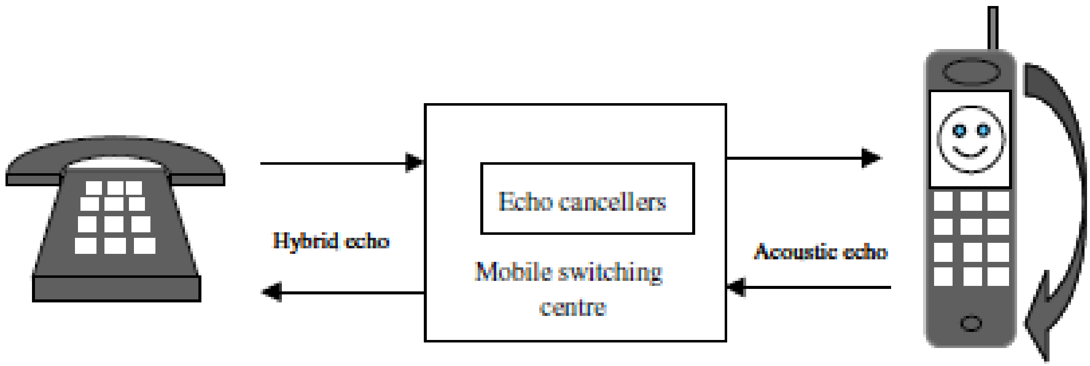


Figure 14.1 Illustration of echo in a mobile to land line system.

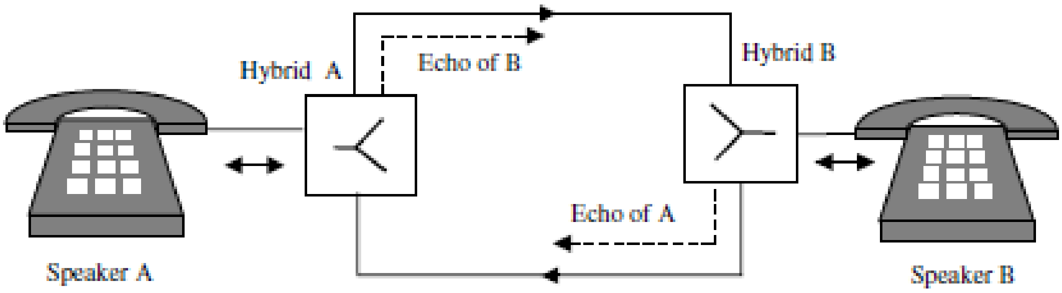


Figure 14.2 Illustration of a telephone call set up by connection of 2-wire subscriber's via hybrids to 4-wire lines at the exchange.

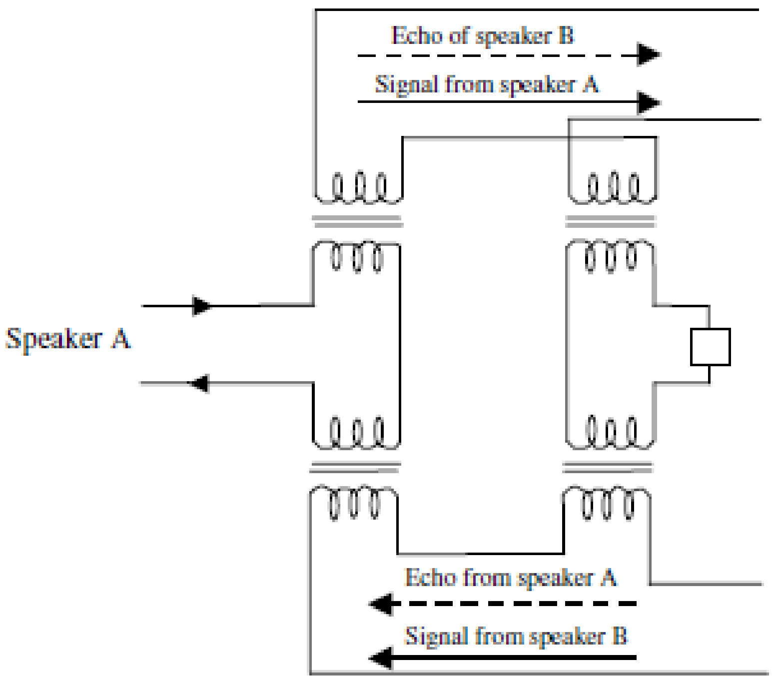


Figure 14.3 A 2-wire to 4-wire hybrid circuit.

14.3 Hybrid Echo Suppression

The development of echo reduction began in the late 1950s with the advent of echo suppression systems. Echo suppressors were first employed to manage the echo generated primarily in satellite circuits. An echo suppresser (Figure 14.4) is primarily a switch that lets the speech signal through during the speech-active periods and attenuates the line echo during the speech-inactive periods. A line echo suppresser is controlled by a speech/echo detection device. The echo detector monitors the signal levels on the incoming and outgoing lines, and decides if the signal on a line from, say, speaker B to speaker A is the speech from the speaker B to the speaker A, or the echo of speaker A. If the echo detector decides that the signal is an echo then the signal is heavily attenuated. There is a similar echo suppression unit from speaker A to speaker B. The performance of an echo suppresser depends on the accuracy of the echo/speech classification subsystem. Echo of speech often has a smaller amplitude level than the speech signal, but

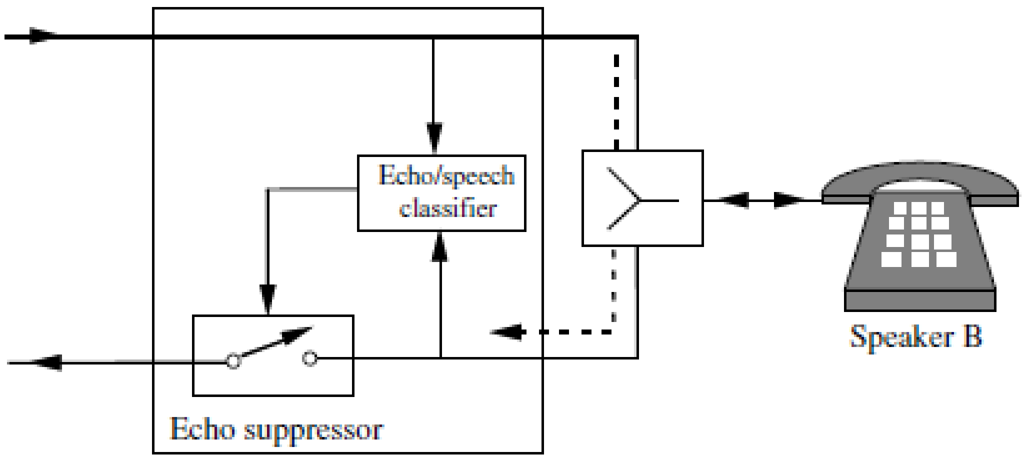


Figure 14.4 Block diagram illustration of an echo suppression system.

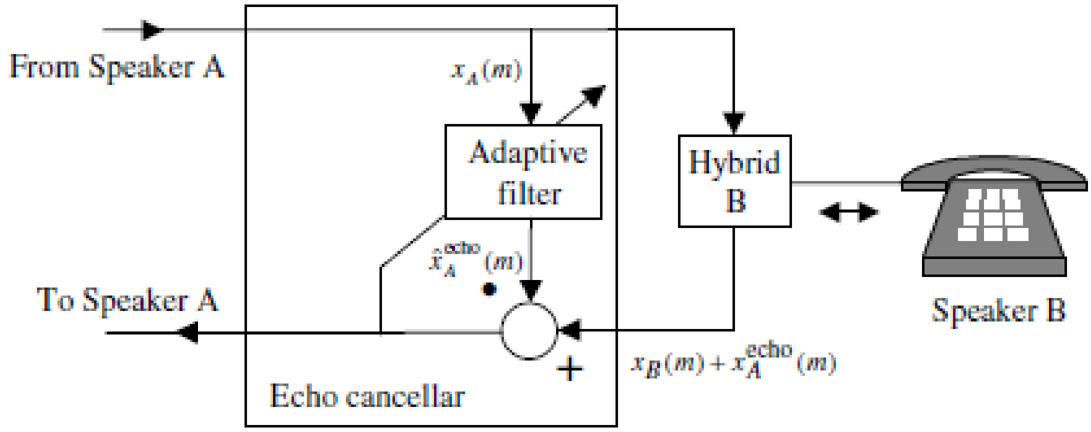


Figure 14.5 Block diagram illustration of an adaptive echo cancellation system.

Assuming that the signal on the line from speaker B to speaker A, $y_B(m)$, is composed of the speech of speaker B, $x_B(m)$, plus the echo of speaker A, $x_A^{echo}(m)$, we have

$$y_B(m) = x_B(m) + x_A^{echo}(m) \tag{14.1}$$

In practice, speech and echo signals are not simultaneously present on a phone line. This, as pointed out shortly, can be used to simplify the adaptation process. Assuming that the echo synthesiser is an FIR filter, the filter output estimate of the echo signal can be expressed as

$$\hat{x}_A^{echo}(m) = \sum_{k=0}^{P-1} w_k(m)\, x_A(m-k) \tag{14.2}$$

CHANNEL EQUALIZATION AND BLIND DECONVOLUTION

Blind deconvolution is the process of unravelling two unknown signals that have been convolved. An important application of blind deconvolution is in blind equalization for restoration of a signal distorted in transmission through a communication channel. Blind equalization has a wide range of applications, for example in digital telecommunications for removal of intersymbol interference, in speech recognition for removal of the effects of microphones and channels, in deblurring of distorted images, in dereverberation of acoustic recordings, in seismic data analysis, etc.

In practice, blind equalization is only feasible if some useful statistics of the channel input, and perhaps also of the channel itself, are available. The success of a blind equalization method depends on how much is known about the statistics of the channel input, and how useful this knowledge is in the channel identification and equalization process. This chapter begins with an introduction to the basic ideas of deconvolution and channel equalization. We study blind equalization based on the channel input power spectrum, equalization through separation of the input signal and channel response models, Bayesian equalization, nonlinear adaptive equalization for digital communication channels, and equalization of maximum-phase channels using higher-order statistics.

$$y(m)=\sum_{k=0}^{P-1} h_k(m)x(m-k)+n(m) \tag{15.2}$$

where $h_k(m)$ are the coefficients of a P^{th} order linear FIR filter model of the channel. For a time-invariant channel model, $h_k(m)=h_k$.

In the frequency domain, Equation (15.2) becomes

$$Y(f)=X(f)H(f)+N(f) \tag{15.3}$$

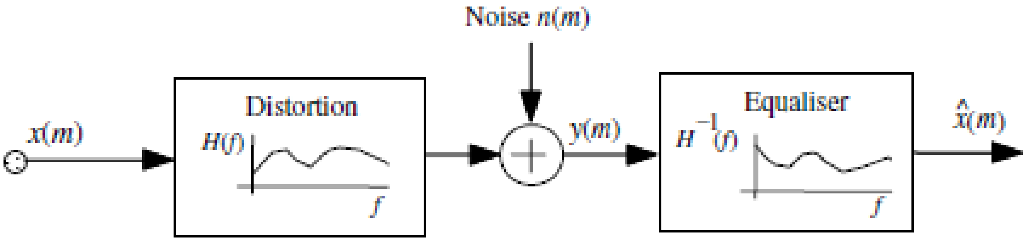


Figure 15.1 Illustration of a channel distortion model followed by an equalizer.

where $Y(f)$, $X(f)$, $H(f)$ and $N(f)$ are the frequency spectra of the channel output, the channel input, the channel response and the additive noise respectively. Ignoring the noise term and taking the logarithm of Equation (15.3) yields

$$\ln|Y(f)|=\ln|X(f)|+\ln|H(f)| \tag{15.4}$$

From Equation (15.4), in the log-frequency domain the effect of channel distortion is the addition of a "tilt" term $\ln|H(f)|$ to the signal spectrum.

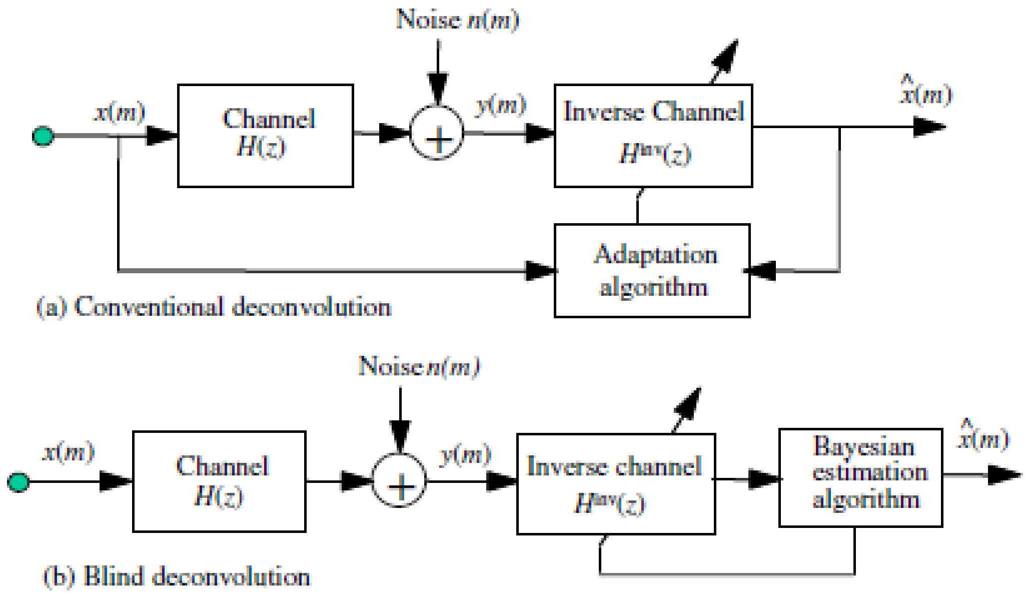


Figure 15.2 A comparative illustration of (a) a conventional equalizer with access to channel input and output, and (b) a blind equalizer.

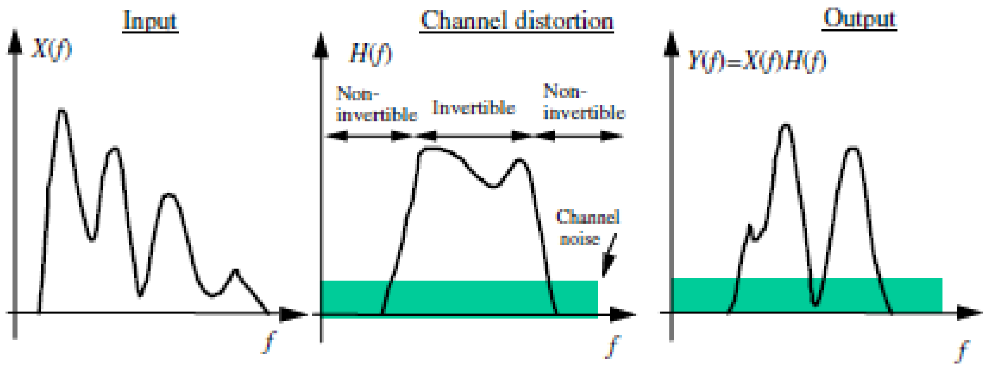


Figure 15.3 Illustration of the invertible and noninvertible regions of a channel.

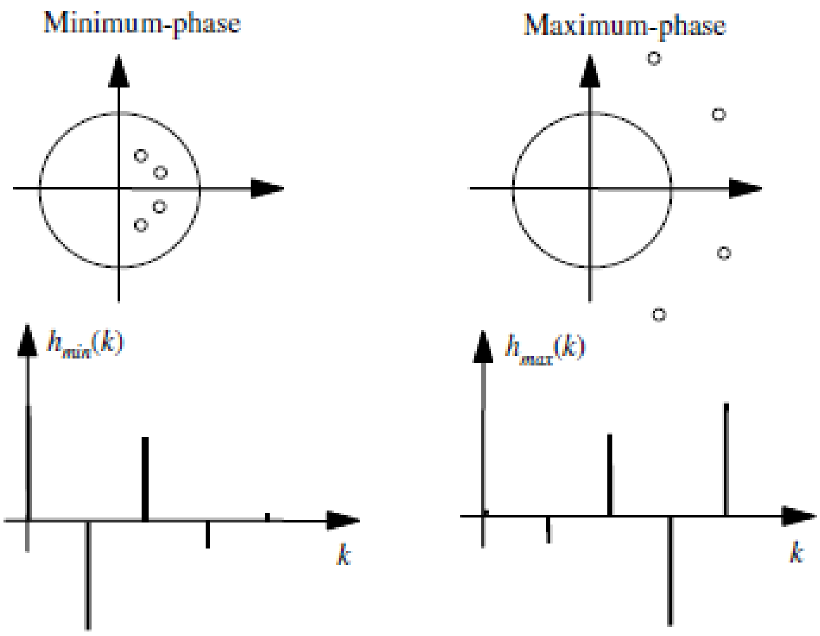


Figure 15.4 Illustration of the zero diagram and impulse response of fourth order maximum-phase and minimum-phase FIR filters.

15.1.5 Wiener Equalizer

In this section, we consider the least squared error Wiener equalization. Note that, in its conventional form, Wiener equalization is not a form of blind equalization, because the implementation of a Wiener equalizer requires the cross-correlation of the channel input and output signals, which are not available in a blind equalization application. The Wiener filter estimate of the channel input signal is given by

$$\hat{x}(m) = \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}}\, y(m-k) \tag{15.16}$$

where $\hat{h}_k^{\text{inv}}$ is an FIR Wiener filter estimate of the inverse channel impulse response. The equalization error signal $v(m)$ is defined as

$$v(m) = x(m) - \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}}\, y(m-k) \tag{15.17}$$

The Wiener equalizer with input $y(m)$ and desired output $x(m)$ is obtained from Equation (6.10) in Chapter 6 as

$$\hat{\boldsymbol{h}}^{\text{inv}} = \boldsymbol{R}_{yy}^{-1}\boldsymbol{r}_{xy} \tag{15.18}$$

where $\boldsymbol{R}_{yy}$ is the $P \times P$ autocorrelation matrix of the channel output, and $\boldsymbol{r}_{xy}$ is the P-dimensional cross-correlation vector of the channel input and output signals. A more expressive form of Equation (15.18) can be obtained by writing the noisy channel output signal in vector equation form as

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} \tag{4.19}$$

where $\boldsymbol{y}$ is an N-sample channel output vector, $\boldsymbol{x}$ is an $N+P$-sample channel input vector including the P initial samples, $\boldsymbol{H}$ is an $N\times(N+P)$ channel distortion matrix whose elements are composed of the coefficients of the channel filter, and $\boldsymbol{n}$ is a noise vector. The autocorrelation matrix of the channel output can be obtained from Equation (15.19) as

$$\boldsymbol{R}_{yy} = \mathcal{E}[\boldsymbol{y}\boldsymbol{y}^{\text{T}}] = \boldsymbol{H}\boldsymbol{R}_{xx}\boldsymbol{H}^{\text{T}} + \boldsymbol{R}_{nn} \tag{15.20}$$

Other equalization- Page 450 to 484