

# Signal Decomposition for Nonstationary Processes

by

Min Xie

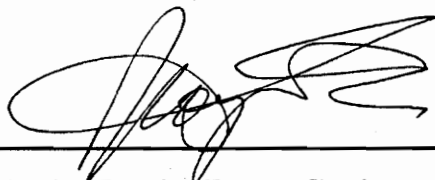
Dissertation submitted to the Faculty of the  
Virginia Polytechnic Institute and State University  
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

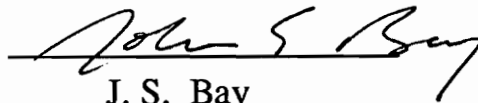
APPROVED:




A. A. (Louis) Beex, Chairman



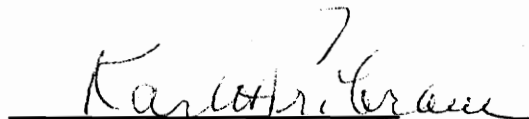
I. Jacobs



J. S. Bay



W. T. Baumann



K. H. Pribram

April 24, 1995  
Blacksburg, Virginia

c.2

LD

5655

V856

1995

X54

c.2



# Signal Decomposition for Nonstationary Processes

by

Min Xie

A. A. (Louis) Beex, Chairman

The Bradley Department of Electrical Engineering

VIRGINIA TECH

Blacksburg, VA 24061-0111

## (ABSTRACT)

The main purpose of this dissertation is to explore and develop better signal modeling (decomposition) methods for nonstationary and/or nonlinear dynamic processes. Localization is the main focus. The characteristics of a nonstationary or nonlinear signal are decomposed onto a set of basis functions, either in the *phase space* spanned by time-frequency coordinates as Gabor proposed, or in the *phase space* spanned by a set of derivatives of different degree as defined in physics.

To deal with time-varying signals, a Multiresolution Parametric Spectral Estimator (MPSE) is proposed together with its theory, techniques and applications. The resolution study provides the characteristics of windowed Fourier transforms, wavelet transforms, fixed resolution parametric spectral estimators, and the newly developed MPSE. Both the theoretical and the experimental results show that, of the above techniques, MPSE is the best in resolution. Furthermore, with proper *a priori* knowledge, MPSE can yield better resolution than the lower bound defined by the Heisenberg uncertainty principle. The application examples demonstrate the great potential of the MPSE method for tracking and analyzing time-varying processes.

To deal with the time-varying characteristics caused by linearization of nonlinear processes, the Radial Basis Function Network (RBFN) is proposed for modeling nonlinear processes from a 'local' to a 'global' level. An equal distance sample rule is proposed for constructing the RBFN. Experiments indicate that the RBFN is a promising method for modeling deterministic chaos as well as stochastic processes, be it linear or nonlinear. The 'local' to 'global' approach of the RBFN also provides great potential for structure adaptation and knowledge accumulation.

## Acknowledgments

I wish to express my deepest gratitude to my advisor, Dr. A. A. (Louis) Beex, for the advice, perspective insight, encouragement, immeasurable patience and support throughout the development of this dissertation. He provided me the intellectual stimulation and the freedom to pursue my research interests. Without these, the research for this dissertation would have been much harder.

I give my sincere thanks to Dr. Karl H. Pribram, at the Center for Brain Research and Information Sciences at Radford University, for his unlimited understanding and great support. During all the years of working with him, his experience, strategies and intuition for finding, analyzing, and solving research problems have greatly influenced me, which is a lifelong gift from him.

I would like to extend my gratitude to Dr. I. Jacobs, Dr. W. T. Baumann and Dr. J. S. Bay for their wonderful lectures and seminars during my program, helpful advice and comments in developing this dissertation, and for serving on my committee. I would also like to thank Loretta Estes for her administrative assistance.

It was the kind spirit shared by very special people, my wife and daughter, which provided me the encouragement and compassion for this research, and which made it become alive and meaningful.

# Table of Contents

<b>Glossary of Key Symbols</b>	vii
<b>1. SIGNAL DECOMPOSITION BACKGROUND</b>	<b>1</b>
1.1 General signal decomposition forms	1
1.2 Characterization of decomposition models	4
1.3 Time-frequency representations for nonstationary processes	6
1.4 Radial basis function representations for nonstationary processes	10
1.5 Network representations for signal modeling and approximation	16
1.6 Survey of previous research	20
1.6.1 Research in linear time-frequency representation	20
1.6.2 Research in parametric spectral estimation	22
1.6.3 Research in ANNs and RBFNs for function approximation	23
1.7 Objectives of this effort	25
<b>2. MULTIREOLUTION PARAMETRIC SPECTRAL ESTIMATION (MPSE)</b>	<b>29</b>
2.1 Overview of MPSE	29
2.2 Octave band-splitting filter design for MPSE	31
2.2.1 Symmetric band-splitting filter design	32
2.2.2 Asymmetric band-splitting filter design	34
2.2.3 Auxiliary band-splitting filter design	42
2.3 Realization of AR-PSE in MPSE	47
2.3.1 Overview	47
2.3.2 Derivation of the WRLS algorithm	49
2.3.3 Initialization of the WRLS algorithm	58
2.3.4 Performance of WRLS in a nonstationary environment	61
2.4 Magnitude estimation with auxiliary minimum variance spectral estimator	62

2.5	Some MPSE realization issues	70
<b>3.</b>	<b>MPSE RESOLUTION ANALYSIS</b>	<b>71</b>
3.1	Computation of the resolutions in time-frequency space	71
3.2	Frequency resolution	74
3.3	Time resolution	75
3.4	Resolution cell size in time-frequency space	82
3.5	Interpretation of MPSE in terms of the wavelet transform	83
3.6	Computer simulations for MPSE	86
3.7	Heisenberg uncertainty principle lower bound and resolution issues	99
3.8	Summary	101
<b>4.</b>	<b>DESIGN OF TIME-VARYING OPTIMAL FILTERS VIA MPSE</b>	<b>102</b>
4.1	Optimal Wiener filtering background	103
4.2	Optimal Wiener filter realization	106
4.2.1	Noncausal IIR approximation of the optimal Wiener filter	106
4.2.2	Causal IIR approximation of the optimal Wiener filter	107
4.2.3	FIR approximation of the optimal Wiener filter	110
4.3	Time-varying MPSE Wiener filtering	112
4.4	Examples of optimal filtering of noise contaminated nonstationary signals	116
4.5	Tracking noise contaminated nonstationary signals in phase space	122
4.6	Summary	125
<b>5.</b>	<b>TIME-FREQUENCY ANALYSIS OF EEG SIGNALS</b>	<b>126</b>
5.1	Introduction	126
5.2	Time-frequency analysis of EEGs	128
5.3	Experimental results	131
5.3.1	Examples of time-frequency analysis of running EEG	132

5.3.2	Examples of time-frequency analysis of evoked potentials	141
5.4	Summary	145
<b>6.</b>	<b>OPTIMAL WAVELET DESIGN VIA MPSE</b>	<b>146</b>
6.1	Wavelet transform background	146
6.2	Wavelet transform issues	149
6.3	Adaptive wavelet transforms	152
6.4	Optimal wavelet design	154
6.5	Experiments	159
6.6	Summary	162
<b>7.</b>	<b>RADIAL BASIS FUNCTION DECOMPOSITION AND ADAPTATION</b>	<b>163</b>
7.1	Introduction	163
7.2	Radial basis function network construction	165
7.2.1	Solving for the model parameters of the radial basis function network	165
7.2.2	Selection of center locations for the radial basis functions	167
7.3	RBF decomposition for time series prediction	172
7.4	Implementation of the RBFN for biological data analysis	182
7.5	Experiments in structure adaptation for the radial basis function network	189
7.5.1	Overview of structure adaptation	189
7.5.2	RBFN adaptation	191
7.5.3	Experiments for ARBFN	195
7.6	Summary	202
<b>8.</b>	<b>SUMMARY AND FUTURE RESEARCH DIRECTIONS</b>	<b>203</b>
	<b>BIBLIOGRAPHY</b>	<b>206</b>
	<b>VITA</b>	<b>218</b>

## Glossary of Key Symbols

ACF	AutoCorrelation Function
ANN	Artificial Neural Network
AR	AutoRegressive
ARMA	AutoRegressive Moving Average
AR-PSE	AutoRegressive Parametric Spectral Estimator
C-IIR	Causal Infinite Impulse Response
CQF	Conjugate Quadrature Filter
DTWS	Discrete Time Wavelet Series
DWT	Discrete Wavelet Transform
ECG	Electrocardiograph
EDSR	Equal-Distance Sampling Rule
EEG	Electroencephalograph
EMG	Electromyogram
FIR	Finite Impulse Response
FM	Frequency Modulation
FT	Fourier Transform
GMDH	Group Method of Data Handling
GT	Gabor Transform
IIR	Infinite Impulse Response
LMS	Least Mean Squares
LMVU	Linear Minimum Variance Unbiased
MA	Moving Average
MLE	Maximum Likelihood Estimate
MPSE	Multiresolution Parametric Spectral Estimator
MVSE	Minimum Variance Spectral Estimator
NC-IIR	Non-Causal Infinite Impulse Response
ODWT	Orthonormal Discrete Wavelet Transform
OLS	Orthogonal Least Squares

PSE	Parametric Spectral Estimator
QMF	Quadrature Mirror Filter
RBF	Radial Basis Function
RBFN	Radial Basis Function Network
RLS	Recursive Least Squares
r.m.s.	Root Mean Square
SNR	Signal-to-Noise Ratio
STFT	Short Term Fourier Transform
TFR	Time-Frequency Representation
WFT	Windowed Fourier Transform
WRLS	Windowed Recursive Least Squares
WT	Wavelet Transform
WVD	Wigner-Ville Distribution

# 1. SIGNAL DECOMPOSITION BACKGROUND

The main, combined purpose of this dissertation is to explore and develop better signal modeling (decomposition) methods for nonstationary and/or nonlinear dynamic processes. To serve this goal, this chapter first reviews the research in related fields. Different methods, such as parametric and nonparametric decomposition methods, and globally and locally supported basis functions for signal modeling in the phase space, are introduced in the first 5 sections serving as “Background.” These are then compared in Section 1.6, the “Survey.” Based on the research survey the objectives of this research are provided in Section 1.7, where the organization of the dissertation can be found as well.

## 1.1 General signal decomposition forms

A *signal* is generally defined as “the result of measurements on a physical system under observation [1].” The measurement results can be in different forms, such as numerical quantities, signs, or symbols, which reflect information about the state of the physical system. Given a set of observations, one often wants to condense and summarize that data by fitting a proper *model* to it. That is, the observed data are assumed to express a process which can be represented by a set of known mathematical functions, and the data and known functions are related by a set of adjustable parameters. Signal modeling is thus equivalent to signal decomposition. The model functions are known as *basis functions* in the decomposition, and are the fundamental components for constructing the observations. The adjustable parameters are often referred to as *weighting coefficients*, which reflect a measure of the dependence between the data and the basis functions.

The selection of the model basis is not unique to the observed process. Sometimes, the model basis is selected based on an underlying theory or physical law that the data are supposed to satisfy, but most often the model basis is selected from convenient mathematical forms such as the polynomials (*Taylor series expansion*), complex exponentials (*the Fourier transform*), Gaussians (*the radial basis functions*), or the derivatives of the independent variable  $x$  (*the differential or state space models*). The key to successfully modeling a signal lies in selecting the basis functions.



A good model not only fits the observations well, it also simplifies the underlying signal processing problem.

According to the relation between the signal and the model parameter set, the decomposition can be either *linear* or *nonlinear*. Quite often, the linear decomposition operation is much simpler than any nonlinear one. The former allows the application of systematic procedures, while the latter is case dependent. In many cases, a sophisticated nonlinear process can have a linear representation in the decomposition model by selecting proper nonlinear functions as its basis functions. This dissertation will focus mainly on the linear decomposition models.

The continuous form of a linear decomposition is defined by

$$y(x) = \int_{\Omega} a(\Omega) \Phi(x, \Omega) d\Omega \quad (1.1a)$$

where  $\Phi(x, \Omega)$  is the basis function, and  $a(\Omega)$  is the weighting function; both of which are continuous functions of  $\Omega$ .

The discrete form of a linear decomposition is defined by

$$y(x) = \sum_{k=1}^M a_k \Phi_k(x) \quad (1.1b)$$

where  $\Phi_1(x), \Phi_2(x), \dots, \Phi_M(x)$  are the basis functions, and  $\{a_1, a_2, \dots, a_M\}$  is the model parameter set or the weighting coefficients. The relation between the function  $y(x)$  and the basis functions is linear in (1.1a) and (1.1b), although both the function  $y(x)$  and the basis  $\Phi(x, \Omega)$  or  $\Phi_k(x)$  can be nonlinear functions of  $x$ . In this discussion *linear* refers to the model's dependence on its parameter set  $a(\Omega)$  or  $A = \{a_1, a_2, \dots, a_M\}$ , i.e. the models are linear-in-the-parameters.

The mathematical principles used for signal decomposition are found in functional analysis [1, 2, 3]. The signal and its components, the basis functions, are the elements in the *metric space* in which a distance between any two elements is defined. This distance, called a *metric*, is a functional that projects two functions onto the positive real number line,  $d: \{x, y\} \rightarrow R^+$ . The basic approach in signal decomposition follows a two step approach. The first step is to choose a metric

which measures the agreement between the data and the model for a particular choice of parameter set. The second step is to then adjust the model parameters in order to achieve the minimum distance between the observation  $y(x)$  and its model representation  $\hat{y}(x)$ , yielding the *best fit* parameters. Finding the best fit parameters is a problem of optimization, in which the metric is called the *merit function* in optimization, or the *error function* in estimation. Different merit functions can be used, such as [1]

$$d_1(y(x) - \hat{y}(x)) = \int_x |y(x) - \hat{y}(x)| dx \quad (1.2a)$$

$$\begin{aligned} d_2(y(x) - \hat{y}(x)) &= \|y(x) - \hat{y}(x)\|_2 \\ &= \left[ \int_x |y(x) - \hat{y}(x)|^2 dx \right]^{\frac{1}{2}} \end{aligned} \quad (1.2b)$$

$$d_3(y(x) - \hat{y}(x)) = \sup\{|y(x) - \hat{y}(x)|; x \in X\} \quad (1.2c)$$

where  $X = \{x, a \leq x \leq b\}$  is the support region over which  $x$  is defined. Among these definitions, the metric  $d_1$  is a measurement of the absolute difference, the metric  $d_2$  is referred to as the norm of  $y(x) - \hat{y}(x)$ , and the metric  $d_3$  is the maximum absolute difference. In optimization theory, these merit functions define different optimization criteria, such as,  $d_1$  for the *absolute error criterion*,  $d_2$  for the *least squares error criterion*, and  $d_3$  for the *minimax* or *Chebyshev error criterion*.

The model representation provides an approximation to the original signal. An important reason for modeling, in many cases, is that there is no known function which precisely describes the observations. The choice of basis functions is based on the assumption that the data can be approximated by a set of known functions. One example would be to match an arbitrary curve by using a polynomial, in which it is assumed that the unknown curve function can be extended by the Taylor series. An imperfect fit of the model basis functions is not the only cause of error; there are

also errors caused by contaminated observations, by limitations in the number of basis functions that can be chosen practically, and by numerical computation itself.

## 1.2 Characterization of decomposition models

There is a variety of ways to characterize the decomposition models. Depending on the arguments of the basis functions, the decomposition operators can be classified as *parametric* or *nonparametric*. For nonparametric decompositions, the basis functions are usually independent of the observation data, for example when the arguments of the basis functions are time (space) and/or frequency instead of the observation data. Examples of model functions for nonparametric decomposition are the complex exponentials in the Fourier transform, or a class of wavelet functions in the wavelet transform, for which the arguments are either frequency or time (space).

Unlike for nonparametric methods, the parametric decompositions usually use the observation data as arguments of their basis functions. The incoming signal is assumed to be generated by some linear or nonlinear dynamic system. The model is thus considered to be a representation of the original dynamic system by which the signal is generated. The model equations are directly linked to the differential or difference equation, or to the state space equations, of the observed system.

The differences between the parametric and nonparametric decomposition methods are not only reflected in the arguments of the basis functions, but also in the characteristics of these basis functions. In nonparametric methods, the basis functions are a class of known time- (space-) frequency functions, and each is an element or a point in  $L^2(\mathbb{R}^n)$  space, where  $n$  is the number of arguments (dimension) of the basis. There is an infinite number of basis functions, which uniformly fill the entire  $L^2(\mathbb{R}^n)$  space. The decomposition coefficients are the inner products of the signal and the basis functions, so that the result is a measure of the linear dependence between the signal and the basis functions into which the signal is decomposed. If the signal is defined in a  $L^2(\mathbb{R}^m)$  space, where  $m$  is the dimension of the signal, the nonparametric decomposition maps the signal from  $L^2(\mathbb{R}^m)$  into  $L^2(\mathbb{R}^n)$  space and, therefore, this kind of decomposition operation is also called a transformation operation by which the signal is transformed from one domain into another [4]. The reason for calling this a nonparametric decomposition is that the basis functions are independent of

the signal, and there is no dynamic model assumed to be involved in the operation. This is an important feature of the nonparametric methods since they require the minimum *a priori* knowledge of only the observation.

The basis functions in parametric decomposition methods, however, are the coordinates which span the entire signal space. The number of basis functions is often related to the order of the assumed underlying dynamic system, and finite. The coefficients of the decomposition are the projections of the signal onto these coordinates. Therefore, the basis functions are always signal dependent, and they are the representation of a specific dynamic model for the signal [5]. This kind of modeling requires *a priori* knowledge of the signal process, such as the model structure, the orders of the dynamics, etc.

The typical nonparametric methods are the Fourier Transform (FT), Windowed Fourier Transform (WFT), Short Term Fourier Transform (STFT), Gabor Transform (GT), Wavelet Transform (WT), and Wigner-Ville Distribution (WVD). These operators are referred to as time-frequency analyzers. The popularly used linear parametric decomposition model for discrete time series are the AutoRegressive models (AR), the Moving Average models (MA), and the AutoRegressive Moving Average models (ARMA).

The decomposition operation can also be characterized, according to the support region of the basis function, as locally or globally supported. The globally supported basis functions extend over the entire  $\mathbb{R}^n$  space, such as  $e^{j\omega t}$  and  $\dot{x}(t)$ . The locally defined basis functions are those locally effective functions with effectively finite duration or bandwidth (to be defined later) such as the Gaussian functions, and time window functions. Such basis functions are also called compactly supported basis functions. The Fourier transform and the AR, MA, and ARMA models are globally supported basis decompositions, while the WFT and the WT are examples of compactly supported nonparametric decompositions. An important form of a compactly supported basis function is known as the Radial Basis Function (RBF), which is of the form [6]

$$\Phi_i(\|\mathbf{X} - \mathbf{C}_i\|), \mathbf{X} \in \mathbb{R}^n, i = 1, 2, \dots, m, \quad (1.3)$$

where  $\mathbf{C}_i$  is the center location of the  $i$ -th basis function.  $\Phi_i$  maps from  $\mathbb{R}^n$  to  $\mathbb{R}$ .

There are three major reasons for compactly supported basis decomposition. The first reason is that signals, in many applications, are generally constrained to a finite dynamic region. Any modeling is then meaningful only within this feasible dynamic region. A locally fit model quite often can simplify the problem. The second reason is that when the signal is time-varying or nonstationary, the basis is no longer time-invariant, and in that case a time window can restrict the effective region of the basis. The final reason, for a compactly supported basis, is that when the relation between the basis and the signal is nonlinear, a locally defined basis function can be used for the piecewise construction of an approximation to any sophisticated nonlinear manifold, which can not be simply built by the linear combination of any known functions.

### 1.3 Time-frequency representations for nonstationary processes

One of the oldest and most effective techniques in signal processing is frequency analysis by using the Fourier transform. The process underlying the Fourier transform is to decompose the signal onto a set of known complex exponentials, that is, the signal is modeled as consisting of a class of complex oscillations. The Fourier coefficients, the linear dependence measure between the signal and the basis functions, are computed by taking the inner product of the signal and the basis functions. The ideal form of Fourier decomposition is defined by:

$$Y(f) = \left\langle y(t), e^{j2\pi ft} \right\rangle = \int_{-\infty}^{\infty} y(t) e^{-j2\pi ft} dt \quad (1.4a)$$

where  $y(t)$  is an arbitrary finite-energy signal,  $e^{j2\pi ft}$  is the basis function at frequency  $f$ , and  $Y(f)$  is the Fourier transform coefficient or the frequency representation of  $y(t)$ . The Fourier transform maps the signal from the time (space) domain into the frequency domain. The inverse Fourier transform maps the frequency representation back into the time representation, and has the following general form

$$y(t) = \left\langle Y(f), e^{-j2\pi ft} \right\rangle = \int_{-\infty}^{\infty} Y(f) e^{j2\pi ft} df \quad (1.4b)$$

The Fourier transform and its inverse establish a one to one relation between the time domain (the signal  $y(t)$ ) and the frequency domain (the spectrum  $Y(f)$ ). These two domains constitute two alternative ways of looking at the same signal.

The AutoCorrelation Function (ACF) is another form of time representation of the signal. It is a self-similarity measure of the signal over time, which tells us how the information in the signal is distributed over time, such as how long the same information lasts, whether there are any repeating patterns (periodicity) or decaying repeating patterns. The Fourier transform of the ACF, by definition, is the power spectrum of the signal, i.e.

$$P_{xx}(f) = \sum_{k=-\infty}^{\infty} r_{xx}(k) \exp(-j2\pi fk) \quad -\frac{f_s}{2} \leq f \leq \frac{f_s}{2} \quad (1.5)$$

where  $r_{xx}(k)$  is the ACF of signal  $y(t)$ , defined as

$$r_{xx}(k) = E[x^*(n)x(n+k)] \quad (1.6)$$

where  $x^*(n)$  denotes the complex conjugate of the signal  $x(n)$  and  $E[\cdot]$  is the expectation operator. From the definition, if the underlying statistical properties of the signal are known exactly, its power spectrum can be computed from (1.5). The practical reality however, is that the signal is an unknown stochastic process and its observations may have been contaminated by measurement noise, so that only an estimate of the statistical description of the signal can be made based on observations of-usually-short duration. As a result, in practice, spectral analysis is a matter of statistical estimation from observations.

The Fourier transform pair of the signal  $y(t)$  is a linear time and frequency representation, while the ACF and PSD are forms of quadratic time and frequency representations. These Fourier transform pairs are basic to the study of the ever-expanding field of signal processing. In radar, sonar, geophysics, astronomy, communications, and biomedical engineering, signal processing plays a dominant role; hence the importance of spectral analysis. However, in many applications, such as those listed above, the signal analyst is confronted with the task of processing a signal whose statistical characteristics vary with time. These signals are referred to as *time-varying* or

*nonstationary*. An accurate spectral analysis of such signals cannot be accomplished by the simple use of classical time domain representations, such as the ACF methods, or frequency domain representations, methods based on the Fourier transform. Rather, it requires the use of analysis methods that do not assume the condition of stationarity, as is done in classical spectral analysis based on the Fourier transform. For such time-varying signals, a Time-Frequency Representation (TFR) has to be used. The time and frequency dimensions form a new  $\mathbb{R}^2$  space referred to as *time-frequency space*, and the TFR maps the signal (assumed for now to be one dimensional) from  $\mathbb{R}^1$  into  $\mathbb{R}^2$ .

A variety of TFRs show up in the literature. There is the Windowed Fourier Transform (WFT) (also called Short Term Fourier Transform (STFT)), the class of Wavelet Transforms (WT), and the Wigner-Ville Distribution (WVD). The TFRs can be characterized as linear TFRs, such as WFT and WT, or quadratic TFRs such as the WVD, according to whether the time domain form of the signal or its instantaneous power form is involved in the decomposition. The general form of a linear TFR can be represented as [7-10]:

$$c(\tau, \nu) = \langle y(t), h_{\tau, \nu}^*(t) \rangle = \int_{t=-\infty}^{\infty} y(t) h_{\tau, \nu}(t) dt \quad (1.7)$$

where  $c(\tau, \nu)$  is the transform coefficient,  $h_{\tau, \nu}(t)$  is the basis function, and  $y(t)$  is the time domain representation of the signal. For WFT,  $h_{\tau, \nu}(t) = w(t - \tau) e^{-j2\pi\nu t}$ , which is a time-limited complex oscillation at frequency  $\nu$ , where  $w(t - \tau)$  is the sliding window function centered at  $t = \tau$ . For WT,  $h_{\tau, \nu}(t) = \frac{1}{\sqrt{\nu}} h\left(\frac{t - \tau}{\nu}\right)$ , where  $h(t)$  is a basic wavelet prototype called the *mother wavelet*, and  $h_{\tau, \nu}(t)$  is a set of basis functions called the *daughter wavelets*, resulting from translation and scaling of the mother wavelet  $h(t)$ . The argument  $\tau$  is the center of  $h_{\tau, \nu}(t)$ , or the time localization of the wavelet basis function, and  $\nu = \frac{f_0}{f}$  is called the *time scale*, where  $f = \frac{1}{t}$  is the frequency and  $f_0$  is the center frequency of the Fourier transform of the daughter wavelet.  $\nu^{-1}$  is equivalent to frequency in WFT. The *mother wavelet*  $h(t)$  is selected such that it is the

impulse response of a bandpass filter centered at  $f_0 = \upsilon f$ . The main feature distinguishing WFT from WT in the time domain is that the time window width in WFT is constant, so that the number of oscillations of the basis increases with frequency, while the number of oscillations of the WT basis is constant over frequency with a time-varying window width. Therefore, the scale  $\upsilon$  keeps the Q factor of the daughter wavelets constant, i.e.  $Q = \frac{\Delta f}{f_0} = \text{constant}$ , where  $\Delta f$  is the effective

bandwidth of the daughter wavelets. Unlike the conventional Fourier transform, which suffers from fixed frequency and time resolutions, the wavelet transform is capable of trading off resolution between frequency and time domains within different frequency bands. That is, it can zoom in in a high frequency band to catch fast "local" events and zoom out in a low frequency band to observe slow "global" changes. The associated resolution issue will be further discussed later.

The quadratic time-frequency representations give the signal power distribution in the time-frequency domain. There are many different forms of quadratic TFRs; one of the most popular is the Wigner-Ville Distribution (WVD) [8, 11-14], defined as:

$$W(t, f) = \int_{-\infty}^{\infty} z(t + \tau / 2) z^*(t - \tau / 2) e^{-j2\pi f \tau} d\tau \quad (1.8)$$

where  $z(t)$  is the analytic signal, i.e.  $z(t) = y(t) + jH[y(t)]$ , in which  $H$  denotes the Hilbert transform. If  $z(t) = y(t)$  in (1.8) then the equation defines the Wigner Distribution.

The TFRs discussed above are all unrelated to dynamic models, and therefore nonparametric methods. There are also parametric spectrum estimators (PSE), for example the AR model, the MA model, and the ARMA model. These parametric methods can take advantage of *a priori* knowledge about the signal, such as its dynamic model structure, which can then extract knowledge from the observations and apply it beyond the observation time window. The latter feature can yield higher frequency resolution for the parametric PSEs than is possible with nonparametric ones. The higher frequency resolution is especially apparent in a high signal-to-noise ratio (SNR) environment when using short observation windows. However, if the assumed model is invalid, the results can be misleading.



To be able to estimate the statistical characteristics of the observations, the parametric methods require stationarity of the signal. If the signal is nonstationary, a time window is applied to the data to be processed. Pseudo stationarity is assumed inside the time window. This windowing produces a trade-off between the time resolution and the variance of the estimator. With a short time window the estimator can track fast changes, with large estimation error, while a long time window smoothes out the changes of the signal in the window period, resulting in poor time resolution for changing features and lower estimation error for stationary ones.

#### 1.4 Radial basis function representations for nonstationary processes

The time varying characteristics of a signal can also be caused by the linearization of a nonlinear dynamic system. The linearization is performed by expanding the nonlinear function into a power series. If all the derivatives of a single variable nonlinear function  $f(x)$  exist, the function  $f(x)$  can be written as the following infinite power series, expanded about  $x=x_0$ ,

$$f(x) = f(x_0) + \left. \frac{df}{dx} \right|_{x=x_0} \Delta x + \frac{1}{2!} \left. \frac{d^2 f}{dx^2} \right|_{x=x_0} \Delta x^2 + \dots + \frac{1}{k!} \left. \frac{d^k f}{dx^k} \right|_{x=x_0} \Delta x^k + \dots \quad (1.9)$$

where  $\left. \frac{d^k f}{dx^k} \right|_{x=x_0}$  is the value of the  $k$ -th derivative of the nonlinear function  $f(x)$  with respect to  $x$ , evaluated at the point  $x=x_0$ , and  $\Delta x = (x - x_0)$ . Clearly, if  $|\Delta x|$  is small enough, or  $x$  is near  $x_0$ , then  $\Delta x^k \approx 0$ , for all  $k > 1$ . When, in (1.9), the sum of the terms of second and higher powers of  $\Delta x$  are negligible compared with the sum of the first two terms,  $f(x)$  can be rewritten, in the vicinity of  $x_0$ , as:

$$f(x) \approx f(x_0) + \left. \frac{df}{dx} \right|_{x=x_0} (x - x_0); \quad \text{for } |\Delta x| = |x - x_0| < \delta, \quad (1.10)$$

Equation (1.10) is a linear approximation of the nonlinear function  $f(x)$  in the vicinity of  $x_0$ . The

first derivative  $K = \left. \frac{df}{dx} \right|_{x=x_0}$ , or the slope of the linear approximation of  $f(x)$ , is a function of  $x_0$ .

If  $x$  is a function of time  $t$ , the linear approximation of  $f(x(t))$  is time dependent with a time-varying slope  $K(x(t))$ . If time is in discrete form, i.e.  $t = n\Delta t$ ,  $n = 0, 1, 2, \dots$ , and  $x_n = x(n\Delta t)$ , the dynamic trajectory  $y(t) = f(x, t)$ , at  $t = (n+1)\Delta t$ , can be approximated as:

$$\begin{aligned} y_{n+1} &= f(x_n, n\Delta t) + f'(x_n, n\Delta t)(x_{n+1} - x_n) \\ &= y_n + f'_n x_{n+1} + f'_n x_n \end{aligned} \quad (1.11)$$

where  $f'_n = f'(x_n, n\Delta t) = \left. \frac{df(x_n, n\Delta t)}{dx} \right|_{x=x_n}$ . The derivative  $f'_n$ , in (1.11), is time dependent. Thus,

$y_n = f(x, n\Delta t)$  can be treated as a first order time-varying autoregressive moving average process, i.e. an ARMA(1,1) process. More generally, a time invariant nonlinear multivariate dynamic equation  $Y(n\Delta t) = F(X, n\Delta t)$  can be approximated as a linear time-varying ARMA( $p, q$ ) process. The stronger the nonlinearity is, the faster the slope changes. If the dynamic system is nonlinear and time-varying, the tracking of its trajectory with a linear approximation is even more difficult.

From the given example, it is obvious that if a nonlinear model can be used, instead of a linear approximation, the time dependency of the model can be canceled out, or at least slowed down greatly (this is equivalent to including some higher powers of  $\Delta x$  in (1.9)). However, conventional nonlinear modeling suffers from many practical problems. To begin with, there is no systematic way to select the basis functions such that the linear relation between the observation  $y(t)$  and the basis functions can still be preserved. Then, when this relation does not hold, the merit functions for estimating the model parameters are no longer quadratic, so that the guarantee of a single minimum is lost. The end result is a computationally intensive search for a global minimum, from multiple local minima, for which final convergence is not guaranteed either.

One of the recent advances in nonlinear multivariate modeling is the development of Radial Basis Function (RBF) models [6, 15-17]. The RBF representation has the general form of

$$y = f(\mathbf{X}) = \sum_{i=1}^N \alpha_i \Phi_i(\|\mathbf{X} - \mathbf{C}_i\|) \quad (1.12)$$

where  $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$  is a vector in  $\mathbb{R}^n$ ,  $y = f(\mathbf{X})$  is a nonlinear multivariate function which maps from  $\mathbb{R}^n$  to  $\mathbb{R}$ , where  $\{\alpha_i; i=1,2,\dots,N\}$  is the parameter set,  $\Phi_i(\|\mathbf{X} - \mathbf{C}_i\|)$  is the  $i$ -th RBF centered at  $\mathbf{C}_i = \{c_{1,i}, c_{2,i}, \dots, c_{n,i}\}$ , and  $\|\mathbf{X} - \mathbf{C}_i\| = \sqrt{\sum_{j=1}^n |x_j - c_{j,i}|^2} \geq 0$  is the norm (distance) from  $\mathbf{X}$  to the center of the  $i$ -th basis function  $\mathbf{X}_i$ . Some examples of compactly supported RBFs are

$$\Phi_i(\|\mathbf{X} - \mathbf{C}_i\|) = (\|\mathbf{X} - \mathbf{C}_i\|^2 + c^2)^{-\frac{1}{2}} \quad (\text{inverse multiquadrics}) \quad (1.13a)$$

$$\Phi(\|\mathbf{X} - \mathbf{C}_i\|) = e^{-\frac{\|\mathbf{X} - \mathbf{C}_i\|^2}{\sigma^2}} \quad (\text{Gaussians}) \quad (1.13b)$$

where  $c$ , and  $\sigma$  are arbitrary positive constants. The common characteristic of the given examples is that these basis functions will be most effective only at the center  $\mathbf{C}_i$ , and they will quickly die out when  $\|\mathbf{X} - \mathbf{C}_i\|$  increases. This important feature makes it possible to construct a nonlinear function of any strangely shaped manifold by using RBFs of small dome-shaped pieces. Therefore, the RBFs work as building blocks. Instead of searching for a global match of the model parameter set, the model parameters of an RBF can be optimized relatively locally, which avoids the problems with multiple local minima.

In selecting the RBFs, in addition to the functional form, there are two important parameters, i.e. the dilation of the RBF, controlled by  $\sigma$  in (1.13b), and the center location  $\mathbf{C}_i$ . The first parameter affects how localized the RBF will be, and the second one determines where to put these building blocks.

A dynamic function, whether a differential or difference function, can also be written in the form of an RBF representation (see Figure 1.1). For a differential equation,  $\mathbf{X}$  is a vector of derivatives of the system response and derivatives of the driving force, i.e.

$$\mathbf{X} = \left\{ \frac{dy}{dt}, \frac{d^2y}{dt^2}, \dots, \frac{d^l y}{dt^l}; u, \frac{du}{dt}, \dots, \frac{d^{m-1}u}{dt^{m-1}} \right\}.$$

A differential equation, linear or nonlinear, can thus be represented in the form of an RBF. In the case of a difference equation,  $\mathbf{X}$  is a vector composed of  $l$

previous system outputs  $y_{n-k}; 0 < k \leq l$ , and the present and previous driving forces  $u_{n-k}; 0 \leq k \leq m - 1$ , i.e.  $\mathbf{X} = \{y_{n-1}, y_{n-2}, \dots, y_{n-k}; u_n, u_{n-1}, \dots, u_{n-m-1}\}$ . For a linear discrete dynamic system, the difference equation can be represented as:

$$y_n = F(\mathbf{X}) = \langle \Theta, \mathbf{X} \rangle$$

$$= \sum_{i=1}^l a_i y_{n-i} + \sum_{j=0}^{m-1} b_j u_{n-j} \tag{1.14}$$

where  $\Theta = \{a_1, a_2, \dots, a_l; b_0, b_1, \dots, b_m\}$  is the parameter set, and where  $\langle \Theta, \mathbf{X} \rangle$  denotes the inner product of  $\Theta$  and  $\mathbf{X}$ . This is the well-known ARMA( $l, m$ ) model.

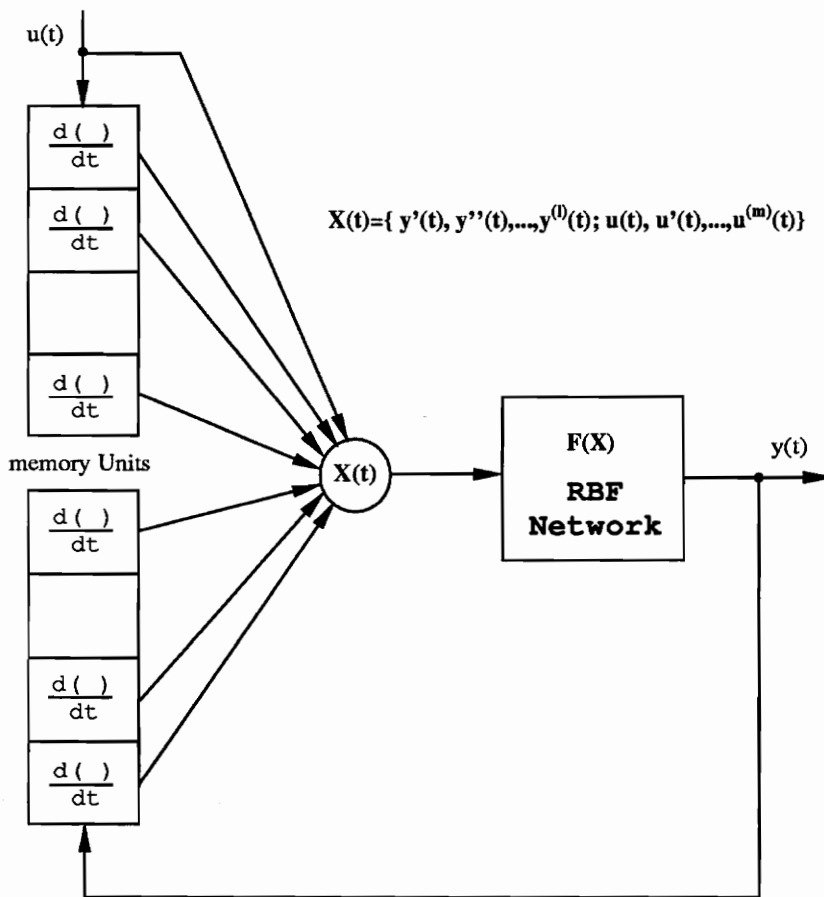
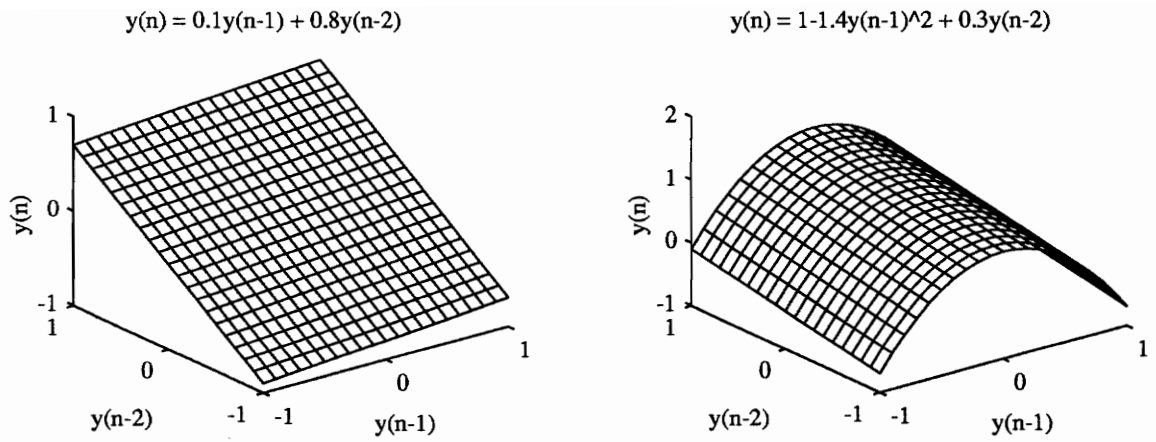


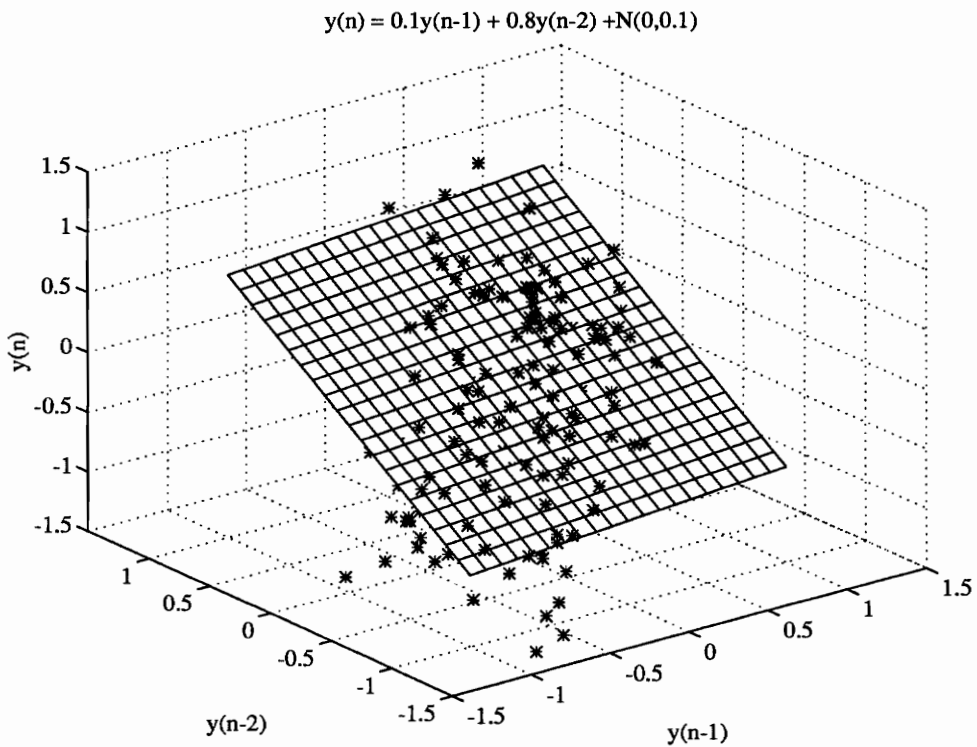
Figure 1.1 RBF model for dynamic system approximation.

The dynamic equation  $y = F(\mathbf{X})$  defines a projection from  $R^{(l+m)}$  to  $R$ . In the linear case, the function  $F(\mathbf{X})$  is a hyperplane in  $R^{(l+m)}$  that is expanded by the vector  $\left\{ \frac{dy}{dt}, \frac{d^2y}{dt^2}, \dots, \frac{d^l y}{dt^l}; u, \frac{d^2u}{dt^2}, \dots, \frac{d^{m-1}u}{dt^{m-1}} \right\}$ , and  $\{y_{n-1}, y_{n-2}, \dots, y_{n-l}; u_n, u_{n-1}, \dots, u_{n-m+1}\}$  for the continuous and discrete cases respectively. Given the coordinates of  $\mathbf{X}(t)$  at time  $t$ , the value of  $y(t)$  can be determined from the hyperplane  $F(\mathbf{X})$ . Figure 1.2a gives an example of a second order linear dynamic system. In a nonlinear system,  $F(\mathbf{X})$  represents a hypersurface in  $R^{(l+m)}$ . If this hypersurface is previously known, the output of the system  $y(t)$  can be determined from the hypersurface (see Figure 1.2b for an example). If the hypersurface is relatively 'flat', it can be approximated by a linear hyperplane, or equivalently one can say that the dynamic system can be approximately modeled as a linear system. Therefore, the time-varying linearization of a nonlinear system can be interpreted as using a tangential hyperplane to continuously approximate the hypersurface of a nonlinear system. Although the dynamic hypersurface of a time-invariant system is static, its linear approximation, i.e. the orientation of the tangent hyperplane, is time-varying as the dynamic trajectory moves around on the hypersurface.

Most system identification or signal modeling problems can be treated as the reconstruction of  $F(\mathbf{X})$  from observations on the hypersurface. Here our discussion was limited to a deterministic dynamic system, where there is no observation noise involved in the model construction. In a stochastic situation, the observations will form a Milky Way, where the dynamic hypersurface  $F(\mathbf{X})$  represents the statistical ensemble properties of the Milky Way (see Figure 1.3 for an example), and more data samples are required to estimate such a hypersurface than in the case of a deterministic system.



**Figure 1.2a,b** Examples of linear (a) and nonlinear (b) dynamic manifolds.



**Figure 1.3** The Milky way and its statistical ensemble of dynamic manifold.

## 1.5 Network representations for signal modeling and approximation

In signal modeling, the major task is to establish a proper mathematical representation called *function*. The function maps a set of independent variables (input variables) to a set of dependent variables (output variables). It defines a projection from the input domain to the output domain, and thus can be equivalently represented as a network. In constructing the model one often uses a set of input variable samples and a set of associated output variable samples to identify the unknown function or the network. Based on some optimal criteria and constraints or *a priori* knowledge, these samples will be linked by the function which gives the 'best' fit to these sample input-output pairs and a 'reasonable' estimate for the region between the observations. Mathematically, such a procedure is equivalent to interpolation.

The Artificial Neural Network (ANN) is a good example of a network representation for the modeling of a signal or an unknown dynamic system. The ANN is simply a nonlinear network which allows people to implement it for solving nonlinear problems without knowing the exact mathematical function. The common form of the ANN is based on a topological network with adjustable weighting functions, in which many simple common elements operate in parallel. The nonlinear sigmoid functions implanted in each element are critical for a neural network to perform nonlinearly. These elements are inspired by biological nervous systems. The ANN's function is determined largely by the connections between elements. Different structures, training rules, and criteria have been developed for training the network. We distinguish between the Widrow-Hoff learning rules [18-20], backpropagation [21-25], associative rules [26-28], and self-organizing Hopfield networks [29-31].

Even with all these efforts however, ANN research is still in its infancy. There are many problems associated with the conventional ANN, such as:

1. The first problem is the topological structure of the ANN. This kind of structure is usually fragile. If one or more nodes and weighting coefficients in the network are modified slightly, the function of the entire network can change dramatically. If one or more connections or nodes in the network malfunction or break, the function of the network becomes unpredictable. This certainly is not the case in a biological nervous system where neurons die frequently. Defects in

the structure of the conventional ANN are the main obstruction for the network to 'learn', to 'self-organize,' and later to 'adapt' to a sophisticated environment.

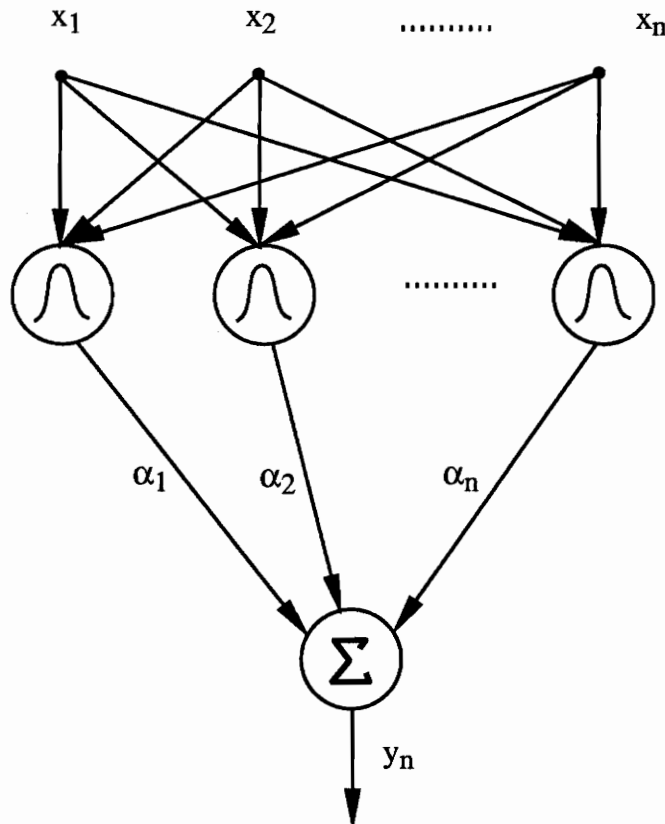
2. The ANN does not simulate biological neurons in function, merely in form. E.g. the sigmoid function of the ANN elements is used to imitate the function of the axon of a real neuron. The biological neuron fires randomly however, and only the associated statistical characteristics carry information. The ANN, on the other hand, is a fixed function network where the connections and elements work deterministically. The result is an imitation of the firing mechanism of the real neuron without imitating its statistical properties. Thus, the ANN is not a functional simulator *per se*.
3. The third problem is the existence of multiple local minima in solving for the best merit function during the training period, especially where the number of distinct objects in the training set is large. No method exists that can guarantee that the global minimum will be reached without being trapped by a local minimum.
4. The final problem in the ANN is that the knowledge of the network can not be accumulated. That is, the network can not learn things one after another. Due to its topological structure, any modification in the structure or weighting coefficients will cause unpredictable results. Thus, the ANN is not an adaptive network after it has been trained. Any modification or increase in the knowledge base will necessitate retraining of the network. The more target events are to be learned simultaneously, the more local minimums will be encountered. This is the main limitation to the learning of any complex problems by a conventional ANN.

In the literature, the radial basis function approach is also called Radial Basis Function Network (RBFN) because they have been developed mostly by the ANN research community. An RBFN can be treated as a two layer feed-forward ANN (See Figure 1.4) [32]. Since the first report of an RBF network for predicting the nonlinear quadratic map equation in 1988 [32], publications on theory and applications have increased dramatically. The main reason is that the RBFN partly avoids the existing problems that the conventional ANN suffers from, such as:



1. The RBF network introduces a field representation and partly gets rid of the topological structure of the conventional ANN. That is, the final form of the RBFN is equivalent to the superposition of many local fields. Each field only affects a small local area and its immediate neighbors (geometric neighbors). If one of these malfunctions, the result will be a hole in the general manifold, and the tails of its immediate neighbors could help to fill part of the empty space to reduce the error. In that sense, the RBFN is more robust than the conventional ANN.
2. The RBFN no longer imitates the firing mechanism of the real neuron. Instead, they directly simulate the statistical characteristics of the neuron, i.e. the synaptodendritic receptive fields of the axons. The receptive fields can be experimentally obtained by stimulating a receptor, or a set of receptors, of an animal (such as the retina), while recording the density of the response (units firing frequency) from the axons. When parameters of stimuli continuously vary, the density of the response will follow the changes. The procedure produces the tuning feature of neurons to their favorite parameter set of stimuli. In the experiment, each member of the parameter set occupies an independent dimension of input space. The receptive field is, thus, constructed by plotting the density of neuron responses against the associated parameter set in the input space [33]. The tuning shape of the receptive field is a multidimensional 'Gaussian' or 'Mexican hat' like manifold mapping the input (parameters of the stimuli) to its output (density of neuron responses). Therefore, understanding how to make the RBFN work, how to train the network step by step, how to realize self-organization and adaptation, may inspire neural scientists to understand the behavior and organization of real neural networks. The RBFN may thus serve as a more *realistic* model for a biological system than the ANN.
3. As the basis functions are locally supported, any basis function is virtually orthogonal to all others except for its immediate neighbors. When searching for a global minimum of the merit function, only one dominant basis function and a few of its immediate neighbors will get involved. This greatly reduces the search space and dimensions, as well as the risk of being trapped by local minimums.
4. The RBFNs offer the possibility of training the network gradually. Due to its special structure, the entire input space can be partitioned into segments. Each segment will be dominated by the basis function defined in the corresponding region. Therefore, the RBFN can learn one bit of

knowledge after the other without the necessity of retraining the entire network. Moreover, by learning gradually, the RBFN can possibly be used as a structure-adaptive signal processor, in which the center location and the radius of the basis functions could be updated dynamically, i.e. as the system under observation changes.



**Figure 1.4** The two-layer feed-forward ANN structure for RBFN.

RBFN research is in the preliminary stages. The theory and technology are more premature than the signal decomposition or modeling methods in the previous sections. It is part of the dissertation research goal to explore this new territory in modeling signals. The exciting problems associated with RBFN work will be discussed in the research survey section.

## 1.6 Survey of previous research

In the last two decades, signal decomposition methods have grown remarkably both in theory and in applications. The approaches include the nonparametric time-frequency representation methods, the parametric spectral estimation methods, and the nonlinear dynamic function approximations such as Artificial Neural Networks (ANNs) and Radial Basis Functions (RBFs). During the seventies and eighties, parametric modeling and spectral estimation received a lot of research attention in many fields, such as communication, signal processing (adaptive filtering, smoothing or predicting, as well as high resolution spectral estimation), system identification, and time series analysis. Efforts and achievements made it one of the most productive research areas during that period. From the middle eighties on, two new research territories began to emerge. One is the wavelet transform for time frequency analysis and the other is the ANN for function approximation. Since then, these have been widely explored in most scientific disciplines, and become part of today's hottest research areas.

The main focus of this brief survey is to point out what has been done and what remains unsolved. We home in on the new promising directions by cross-comparison of the features of these methods.

### 1.6.1 Research in linear time-frequency representation

Research efforts dealing with the localization of time and frequency characteristics of a signal are primarily in two directions. One is to develop a better time and frequency resolution estimator, so that a smaller time window can be used; the other is the development of multiresolution signal processing methods, such as multiresolution coding and the wavelet transform. The multiresolution techniques vary the time and frequency scale in different frequency bands, so that signals are processed with a constant  $Q$  factor instead of with constant frequency bandwidth.

The idea of looking at a signal in various scales and analyzing it with various resolutions took place as early as the beginning of this century. In 1910, Haar published the first complete orthonormal system of compactly supported functions in his dissertation [34]. Based on the mathematical principle developed by Heisenberg two decades earlier, Gabor discovered the limitations of measuring signals in time-frequency phase space in 1946 [35]. He partitioned the phase space into minimum information cells (units) named 'logons'. The size of the cell, the

product of time and frequency resolutions, is bounded below by  $1/2$ , known as the Heisenberg uncertainty principle. This principle “is at the root of the fundamental principles of communication [35]”.

The development of linear time-frequency signal representation has in fact emerged independently in many different fields, such as in mathematics [36], engineering [37, 38], and physics [39, 40]. The name “Ondelettes” (Wavelets) was first introduced in 1982, by the French geophysicist Morlet when he applied wavelets with unbounded support to seismology data [39]. The numerical success of Morlet’s method prompted Grossmann to make a more detailed study of the wavelet transform and build up a strong mathematical foundation for the wavelet transform [40]. Their work initiated unification of the framework of different forms of wavelet transforms and equivalent methods, under the theory of functional analysis.

In the discrete case, two methods were developed independently in the middle seventies and early eighties, in electrical engineering, where subband coding [37, 38] and multiresolution pyramid decomposition [41] were proposed. Since then, the multiresolution TFR methods have been successfully applied in many areas of signal processing, such as image processing [42-44], speech processing [45-47], subband coding and data compression [48], stochastic and nonstationary signal processing [49, 50], and multirate signal processing [51]. The work done by Smith and Barnwell on Conjugate Quadrature Filters (CQF) not only avoided aliasing of the band splitting filter but also provided asymptotically minimal computational complexity ( $O(n)$ ) [52]. Later, Vetterli and Vaidyanathan independently developed theories of the multi-rate digital signal processing filters that generalize Quadrature Mirror Filters (QMF) and CQFs [53, 54]. Both CQFs and QMFs were proved by Daubechies to be equivalent to the discrete wavelet transforms [55], which established the connections between wavelet transform theory and the techniques in discrete signal processing. Daubechies also contributed to the development of the theory for the generalized orthonormal compactly supported wavelet transform and provided conditions for smoothness and polynomial representation as well [55, 56].

Unlike the conventional Fourier transform, which suffers from fixed frequency and time resolutions, the wavelet transform is capable of trading off resolution between frequency and time domains within different frequency bands. That is, it can zoom-in in a high frequency band to catch fast "local" events and zoom-out in a low frequency band to observe slow "global" changes.

However, the wavelet transform is not a high resolution estimator. It has the same resolution problem as Fourier transform based methods, which are bounded by the uncertainty principle. It can be shown that the resolution of the Discrete Time Wavelet Series (DTWS), a popular discrete version of the wavelet transform, is much poorer than the lower bound of  $1/2$  (see the resolution study of DTWS in Sections 3.3 and 3.4). The DTWS also requires that the spectrum of the signal satisfies octave distribution [57], which is an undesirable limitation for many applications.

Meanwhile, there is no publication providing a numerical study of the resolutions for different forms of the WTs and the other equivalent TFR methods. In particular no comparison exists with those parametric spectral estimators which have been known to be superior in frequency resolution.

### **1.6.2 Research in parametric spectral estimation**

Parametric time series modeling and power spectral estimation (PSE) are old problems which can be traced back to the eighteenth century, when Prony applied an exponential model to a set of data representing pressure and volume relationships of gases in 1795 [58]. An important turning point for PSE is marked by the British statistician Yule who, in 1927, introduced a notable alternative analysis method [59]. His idea was to model a time series with a linear regression model, in order to find more accurate periodic components in sunspot data. The regression equation of Yule can be regarded as a prediction of the time series by a linear combination of past samples of the signal plus a Gaussian noise contribution. The least squares technique was, for the first time, applied to solve the regression model. The solution is a set of damped sinusoids. Later, Walker used the same regression method to investigate a damped sinusoidal time series [60]. The general equations for solving the least squares regression parameters have been named the *Yule-Walker equations* in honor of their pioneering research work.

There are many important discoveries and inventions in the research on PSE. For example, there are the discovery of the Fourier transform pairs made up of the autocorrelation function and the power spectral density function [61, 62], as well as the discovery of the special structure of the Yule-Walker equations and the invention of the Levinson recursion algorithm [63]. The prime motivator for the recent interest in high-resolution spectral estimation has been the work of Burg, who introduced the theory of maximum entropy into PSE [64].

For the last two decades, parametric spectrum estimation methods have attracted a lot of interest. Theory and algorithms have developed rapidly. One of the advantages is the potential for better frequency resolution than with Fourier based methods, in particular for short data window and high Signal-to-Noise-Ratio (SNR) conditions [65, 66], which are very important in analyzing time-varying signals. The AutoRegressive (AR) spectral estimator is popular because accurate AR parameter estimates can be found by solving a set of linear equations, while for the Moving Average (MA) and AutoRegressive Moving Average (ARMA) estimators a set of highly nonlinear equations must be solved. A limitation of the AR spectral estimator is that it can be used only for either narrowband or broadband signals, but does not perform very well when both are present simultaneously. Therefore, the AR method is not a multiresolution one *per se*. The frequency and time scales in different frequency bands are the same. Another limitation of the AR technique is that the theory and algorithms for parametric spectral estimators assume a (wide sense) stationary process. For nonstationary signals this assumption is violated. To overcome the nonstationarity, a time window is applied. The window should be short enough to ensure pseudo-stationarity of the input. In a high SNR situation, a short window suffices to estimate the AR parameters. In a low SNR situation, the window must be long enough to yield acceptable variance of the estimator, especially for a high order, multi-frequency process. As always, a trade off between the estimation variance and the inference on nonstationarity is required.

As a result of the stationarity assumption, there has been no study of the time resolution of the AR-PSE although its frequency resolution properties have been explored extensively [65, 66]. It becomes essential to understand the time characteristics of PSE when operating in a nonstationary environment.

### **1.6.3 Research in ANNs and RBFNs for function approximation**

The development of artificial neural networks gained speed during the eighties although they were first invented in the early sixties. Largely, during the last ten years, different theories and network structures have been developed. ANNs have been successfully implemented in various fields including applications in pattern recognition and classification [67-70], speech signal processing [71-73], image processing [74-79], artificial intelligence [80, 81], radar and sonar signal processing [82-85], and in nonlinear system identification, modeling and nonlinear control

[86-91]. Such activity surely marks ANN as one of the fastest growing and productive research areas.

The radial basis function network was first proposed as the method of potential functions in 1964 [92, 93]. The description of the method can be found in the book "Pattern Classification and Scene Analysis" by Duda and Hart [94]. In the later seventies and early eighties, mathematicians used RBFs for interpolation of multivariate scattering data and for function approximation [95-97]. The research goals at the time were to find good basis functions as interpolation spline functions for scattering data, so that one could avoid to sample the unknown process on a grid as required by most other spline functions. One of the important publications by Powell on RBFs for function approximation initiated research interest in many scientific fields [6]. His later paper on the approximation of polynomials with an RBF established the connection between these two decomposition methods [16]. Soon, Broomhead and Lowe found the connections between RBF approximation methods and feed-forward neural networks. They successfully implemented the RBF as a feed-forward neural network model for the "EXOR" (classification) and chaotic time series estimation problems (prediction of the trajectory of the quadratic map equation) [32]. This implementation quickly caught the attention of the neural network society. Since then, publications in both theory and applications have increased explosively. The main reason is that the RBFNs may be capable of solving the problems that conventional ANNs suffer from.

The research history for RBFNs is much shorter than for the conventional ANN. There are many problems in constructing an RBFN. The main problem is the lack of an efficient and effective way to select the parameters of the basis functions. There are more free parameters in an RBFN than in a conventional ANN. These free parameters are the number of the basis functions and their center locations, the dilation in different directions, and the weighting coefficients. Among these parameters, the most important one is the number of basis functions to use for any given dynamic system, together with their center locations. A popular method for selecting center locations is to use all the input vectors in the training set as the center locations for the basis functions. Since many input vectors in the training set are close to each other, using their location as the centers for the basis functions will bring about a lot of redundancy in the network, and cause severe basis function overlap problems. The result is that just as for the conventional ANN, the network hardly converges because of the local minimums. A technique proposed by Chen et al. applies the

Orthogonal Least Squares algorithm (OLS) to determine which input vector is suited to be a center location [98]. After exhaustive recursion, the final set of the *optimal* number and locations of centers for the RBFN is determined from the set of input vectors. This procedure is computationally intensive, especially when the number of members in the training set is large. There is almost no possibility to update the existing network later. As for the conventional ANN, the RBFN has to be retrained each time some new knowledge needs to be incorporated. Vector quantization has reportedly also been used to partition the input vector space and then the center locations of basis functions were determined by the center of gravity of each partition. This method has the same problems as the OLS, in that it is computationally intensive and lacks adaptation capability. In addition, partitioning of the input vectors without consideration of the output is unreasonable for many applications, where the dynamic manifold can have both ‘flat’ surfaces and sharp edges.

## 1.7 Objectives of this effort

Localization is the main focus for this dissertation research. The characteristics of a nonstationary or nonlinear signal are decomposed onto a set of basis functions, either in the *phase space* composed by time-frequency coordinates as Gabor proposed, or in the *phase space* spanned by a set of derivatives of different degree (or time delayed observations for a difference equation) as defined in physics. The latter is also equivalent to the *state space* in linear and nonlinear system theory.

Localization in time and frequency of a signal is of interest in, for example, filtering, detection, and enhancement; it also facilitates reduction of signal information for the purpose of transmission or storage. As mentioned in the previous section, over the last several decades, many researchers have worked on finding improvements in the theory and technology for signal analysis and synthesis, especially under time-varying conditions. To track the changes in a signal, a sliding window is usually applied to the time series, based on the assumption that the signal properties are nearly constant during this window period. A better method is thought to produce better time and frequency resolution.



Research efforts dealing with the localization of time and frequency characteristics of a signal are primarily in two directions. One is to develop a better time and frequency resolution estimator, so that a smaller time window can be used; the other is the development of multiresolution signal processing methods, such as multiresolution coding and the wavelet transform.

Frequency resolution is known to be inversely proportional to the length of the time window used, i.e. a short time window results in poor frequency resolution and vice versa. The Heisenberg uncertainty principle gives a lower bound for the product of the time and frequency resolutions, known as the *resolution cell size*. Wavelet transforms are important since they can trade off the time and frequency resolutions in different frequency bands. Their resolution cell size, however, remains the same as for other Fourier transform based methods. For short records at higher signal-to-noise ratios, the model-based parametric methods have better frequency resolution than those that are Fourier transform based. Still, they rely on an assumption of stationarity or wide sense stationarity, and their time characteristics have remained unknown. The parametric methods are not multiresolution methods either.

To use the superior frequency estimation characteristics of parametric estimation techniques, and also have the flexibility of changing the resolution scale in time-frequency space, the Multiresolution Parametric Spectral Estimator (MPSE) is proposed. It is a hybrid of a parametric spectral estimator and a multiresolution signal processor. Meanwhile, the time characteristics of the parametric spectral estimators are studied in order to derive their time resolution. The resolutions of the conventional parametric methods and the proposed MPSE will be compared with Fourier transform-based nonparametric methods, such as wavelet transforms, especially those for which the lower bound is given by the uncertainty principle. It is also important to know whether or not this bound can be exceeded. If it can, then what are the conditions, and is there any trade-off involved. Finding applications for MPSE is another important part of the research.

In the past, most research efforts concentrated on developing the theory and technology for linear systems and models. The main reason is that in a linear system the local properties reflect its global properties, so that there are general methods to systematically solve a linear problem, and there are universal laws which can be used for analyzing and modeling the behavior of the system. However, the theory and techniques for nonlinear systems were neglected by most of the scientific and engineering society because of the lack of general applicability. To use the linear model and

technology in a nonlinear environment, researchers have studied how to model a nonlinear system as a piecewise linear system, or a linear time-varying system. However, these kinds of approximation can make a simple nonlinear problem become too complicated. With the increase in computational power, it is time for researchers to find a more general mathematical representation which can fit both the linear and nonlinear model.

RBFs have shown promise in constructing an unknown dynamic manifold, from a local one to a global one. However, RBF research is still in a very early stage, and the theory and techniques are immature, as discussed in the survey. There are more free parameters in an RBF network than in other modeling methods and, in particular, there is no good method to determine the number of basis functions and their center locations. The other challenge for RBF realization is an adaptive learning or self-organization feature. Instead of training the RBF network with all of the observations simultaneously, we like the network to learn gradually, i.e. its knowledge accumulates and refreshes during operation without the necessity to retrain from scratch.

Therefore, part of this research is to seek possible effective ways to determine the center location of the RBFs. The performance of the RBFN in modeling both linear and nonlinear dynamic processes (deterministic or stochastic processes) will be compared with those of linear parametric methods. It is important for us to understand the benefits and costs for both the nonlinear modeling method and the piecewise linear modeling method. Adaptive learning and knowledge accumulation is also a partial goal for this research. The purpose here is to design experiments which can demonstrate the idea of adaptive learning, and to explore the issue in order to understand what difficulties may be encountered in future research.

The dissertation is composed of two parts, the first, or main part is to present the Multiresolution Parametric Spectral Estimator (MPSE) decomposition method and its applications, as well as to study the time and frequency resolutions of different signal decomposition methods. There are five chapters in this part. Chapter 2 provides the overview of the MPSE and technical issues regarding the design of the MPSE. Chapter 3 focuses on the resolution of MPSE. The theoretical and experimental results are then compared with those for Fourier transform based methods. The generality of the lower bound on resolution cell size given by the Heisenberg uncertainty principle is discussed as well. Three applications of MPSE are proposed in the next three consecutive chapters. The first application is to use the MPSE for optimal Wiener filter parameter adaptation

(Chapter 4). The second one is the time-frequency analysis of the EEG signals using MPSE (Chapter 5). The third application is optimal wavelet design via MPSE (Chapter 6).

The second part of the dissertation is to develop the technique for the Radial Basis Function (RBF) decomposition method. An equal distance sample rule is developed for constructing the RBF in Section 7.2. Experiments are presented in Section 7.3 and 7.4 for modeling deterministic chaos as well as stochastic processes, be it linear or nonlinear. Meanwhile, Section 7.5 initiates the idea of structure adaptation, selflearning, and knowledge accumulation for RBF neural networks.

The author feels that the most exciting aspect of this dissertation research is that the newly developed methods open new avenues towards overcoming difficulties in the processing of nonstationary signals. Although these methods may be still immature in different respects, they all have great potential in future research in theory, technique, and applications. The research work is concluded in Chapter 8. Some future research directions are listed in this chapter and serve as minnows for catching a whale.

## **2. MULTIREOLUTION PARAMETRIC SPECTRAL ESTIMATION (MPSE)**

Multiresolution techniques were initially developed for subband coding and multiresolution signal processing [37, 52]. The main idea is to split the entire signal frequency band into multiple subbands with a series of filtering and resampling processes. The method has been successfully implemented in speech compression and image pyramid coding, and is equivalent to the discrete time wavelet series (DTWS), a discrete form of the wavelet transform [55, 42]. Multiresolution signal processing is concerned with the issues of signal decomposition and signal reconstruction. The Multiresolution Parametric Spectral Estimation (MPSE) algorithm is developed based on multiresolution signal decomposition.

### **2.1 Overview of MPSE**

Like the discrete wavelet transform, the MPSE method divides the signal frequency band into multiple octave bands. In addition to the band-splitting, MPSE uses an AR spectral estimator in each octave band to estimate the spectral shape of the signal components, such that both the energy of the signal components and their spectral shapes (via the AR parameters) are preserved in this decomposition process [99, 100].

The MPSE process starts by splitting the signal frequency band into the lower halfband and the upper halfband using a lowpass filter and a highpass filter. The output of the highpass filter is forwarded to an AR spectral estimator with a sliding window, such as the Burg algorithm [64] or the Windowed Recursive Least Squares Algorithm (WRLS) [101]. The result is the first octave band spectrum. The output of the lowpass filter is decimated by a factor of two, and sent to the next processing block. Analogous to the first stage, the second processing block consists of highpass and lowpass filters, which filter the decimated signal from the first stage into another set of lower and upper halfbands. The output of the highpass filter is used to estimate the second octave band spectrum, while the output of the lowpass filter is decimated and sent to the

subsequent stage. The processor recursively divides the entire frequency band from high to low frequency into multiple octave bands (Figure 2.1). This "splitting" process reaches its end when one of two conditions is met. The first condition is whether or not the output of the lowpass filter contains only a low order simple spectrum. This condition indicates that the remaining signal can be modeled well by a low order AR process. The second condition is whether or not the output energy of the lowpass filter falls below some minimal energy threshold. If so, all the signal components are accounted for in previous operating stages. After the final stage is thus reached, the signal has been completely decomposed.

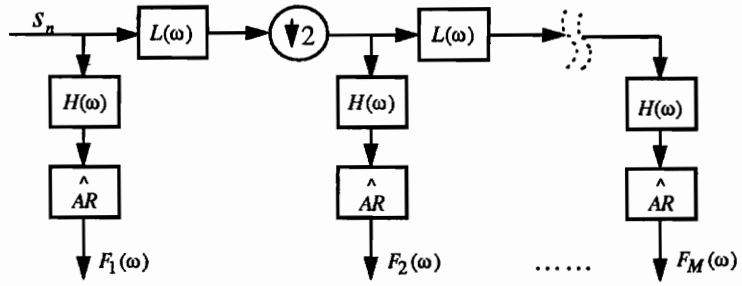
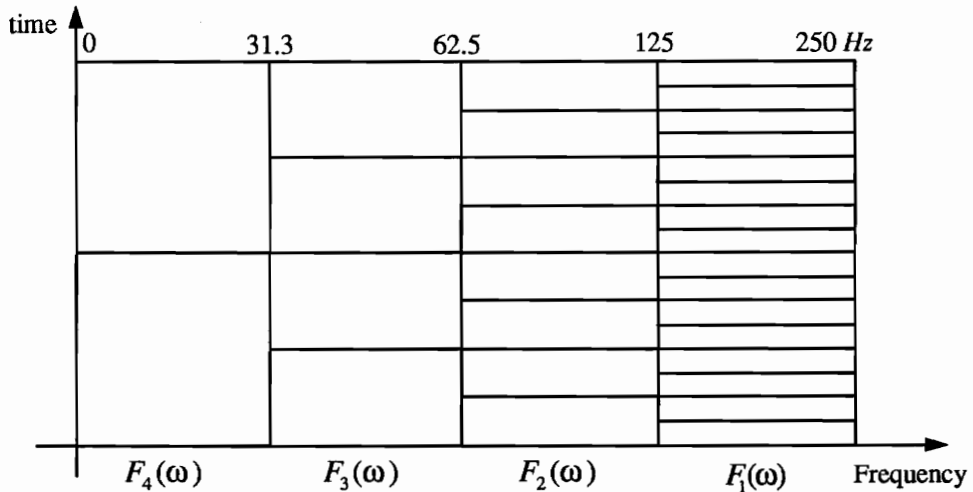


Figure 2.1 System diagram for MPSE.

In applications, the signal often occupies only a few of the octave bands. Computational savings result when only subbands with significant signal power are actually used for spectral estimation. A below-energy-threshold subband signal can be treated as a zero-th order process.

The output rate of each subband signal is different. The first octave band has the highest rate, equal to the sample rate of the original signal. The second octave band has an output rate one half that of the first octave band, and its time scale doubles compared with the first octave band. The composite spectrum plotted in time-frequency space shows a multiresolution time-varying spectrum with a logarithmic scale between octave frequency bands, and a linear scale within each octave frequency band. Figure 2.2 provides an example of the phase space for MPSE.



**Figure 2.2 Example of MPSE phase space, where the entire frequency band is split into four octave bands,  $F_i(\omega)$ ,  $i = 1, 2, 3, 4$ , with a sampling frequency  $f_s = 500$  Hz.**

## 2.2 Octave band-splitting filter design for MPSE

Although the highpass and lowpass filters are identical in each processing stage for frequency band-splitting, there is a variety of ways of designing the octave band-splitting filters. As filter forms the band-splitting filters can be characterized as FIR or IIR realizations. From the occupation of the frequency band the filters can be characterized as symmetric or asymmetric forms. Sometimes a third auxiliary bandpass filter is introduced to improve the performance of the octave band-splitting, especially to reduce information loss on the edges of the band-splitting filters. In this section we will provide the design of symmetric and asymmetric band-splitting filters, as well as of the auxiliary bandpass filter for edge compensation. The performance of these filters and the IIR and FIR realizations are compared and discussed here as well.

### 2.2.1 Symmetric band-splitting filter design

One of the important features of the symmetric band-splitting filter is that the poles and zeros of the highpass and lowpass filters are mirror symmetric about the imaginary axis in the  $z$ -plane. Thus, the parameters of the lowpass filter can be directly transformed from those for the highpass filter by modulating the parameters with  $f_s/2$  or, equivalently, by multiplying the highpass parameters by  $(-1)^n$ . For instance, for an FIR realization, the filter parameters  $h(n)$  of the highpass filters and  $l(n)$  of the lowpass filters, for a filter length  $L$ , are related by

$$l(n) = (-1)^n h(n) \quad n=0,1,2,\dots,L-1 \quad (2.1)$$

In an IIR realization, the parameters of the lowpass filter,  $b_L(n)$  and  $a_L(n)$ , and those for the highpass filter,  $b_H(n)$  and  $a_H(n)$ , are related by

$$\begin{cases} b_L(n) = (-1)^n b_H(n) \\ a_L(n) = (-1)^n a_H(n) \end{cases} \quad n=0,1,\dots,M \quad (2.2)$$

where  $M$  is the order of the IIR filters.

It is well known that FIR filters can preserve linear phase or constant group delay characteristics, while IIR filters do not [102]. When studying MPSE resolution properties, it was observed that an FIR filter produces better time resolution than an IIR filter, given comparable magnitude characteristics and frequency resolution. The linear phase property and better time resolution are important for applications where accurate time localization is required. However, an IIR filter offers much shorter group delay in most of its passband than an FIR filter. This particular feature is extremely important for on-line signal detection.

Figure 2.3 provides an example of the simultaneous FIR realization of both of the symmetric band-splitting filters. From the diagram, one can easily discover the computational advantage of the symmetric design of the band-splitting filter. That is, the highpass and the lowpass filters can share the same tap-weight array. The only difference is in the linear summation. The output of the highpass filter can be computed by summing the tap-weighted incoming signal directly while the output of the lowpass filter can be obtained by summing the tap-weighted incoming signal with alternated signs as in (2.1). When FIR filters satisfy the linear phase property, their coefficients are

symmetric about  $0.5L$ . That is, each incoming signal will be multiplied by the same coefficient twice during the operation. Another half of the computation can be saved by using a bank of tap-delay register arrays.

Unlike the filters used in subband coding or the orthonormal discrete wavelet transform, which require the filter impulse responses to form an orthonormal set [55], the filters used for symmetric band-splitting only require that their passbands be broad enough to cover the frequency halfband with a "reasonable" transition band. Here the term "reasonable" reflects the considerations of trading off between energy leakage and time resolution, as well as group delay. The detriment of a too broad passband and transition band may cause signal energy to leak from one octave band into another, and thereby lower the SNR. However, the high frequency resolution characteristics of AR-PSE will greatly reduce such interference, especially when the energy leakage from an adjacent band is not significant. It is important to keep in mind that the closer these filters are to ideal filters, the longer their impulse response length. A longer impulse response length causes poorer time resolution as well as a larger group delay relative to shorter impulse response length.

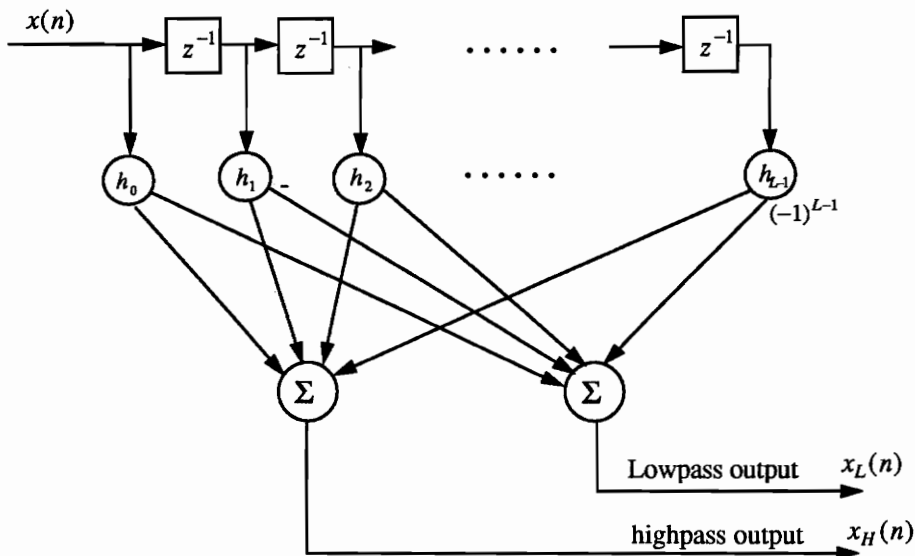


Figure 2.3 FIR realization of octave band-splitting filters specifications.



### 2.2.2 Asymmetric band-splitting filter design

In subband coding, if the Quadrature Mirror Filters (QMF) satisfy certain conditions, the magnitude distortion and aliasing caused by the band-splitting can be canceled out later in the composition process [54]. The combined decomposition and composition operations can be a lossless process. Since MPSE is a spectral estimator, the magnitude distortion and aliasing will affect the representation of the signal in terms of frequency components. Therefore the design of the band-splitting filter requires minimization of the distortion caused by the filtering operation. The distortion caused by the characteristics of the passbands and stopbands of the band-splitting filters can be easily controlled, while the signal energy leakage and distortion at the edges of the octave bands are hard to overcome.

The advantage of the symmetric band-splitting filter design is the associated computational simplicity, while the disadvantage of this design is the unavoidable signal loss at the edges of the band-splitting filters. To prevent aliasing, the passbands of the highpass and lowpass filters do not overlap. There is always a gap between two octave bands, so that the signal frequency components located in or moving across that gap will be seriously attenuated by the roll-off of the band-splitting filters. This edge signal loss can not be reduced by increasing the order or the length of the filters, to make them closer to the ideal filter shapes. Instead, increasing the order or the length of the filter will increase its transient and the group delay, which brings about extra loss caused by poor time resolution. Especially when signal components are moving from one octave band to another, the long transient of high order IIR or long FIR filters will severely attenuate signal power in the edge area.

To prevent edge signal loss in the band-splitting operation, we introduce the design of asymmetric band-splitting filters (see Figure 2.4). The idea of the asymmetric filter design is to make the passbands of the highpass and lowpass filters overlap. That is, the cutoff frequency of the highpass filter  $f_{hcut}$  is designed to be lower than that of the lowpass filter  $f_{lcut}$ . Since the output of the lowpass filter will be decimated later, its highest cutoff frequency is required to fall below 0.25 (fraction of sample frequency) to prevent aliasing. The difference between the two cutoff frequencies,  $\Delta f = f_{lcut} - f_{hcut}$ , is the frequency band shared by both filters. This overlap area helps to reduce the signal loss caused by the filter's edge attenuation and the filter transient when the signal component is moving from one octave band to another. That is, the overlap in the frequency

bands provides a buffer zone that gives extra time for the filter to warm up. Although the highpass filter will occupy more of the low frequency band than the symmetric design, the spectral estimator after the highpass filter will only estimate the frequency components from  $f_{edge}$  to  $f_H$  (Figure 2.4), where  $f_{edge} = (f_{hcut} + f_{lcut})/2$  and  $f_H = 2f_{edge}$ , except in the first and last of the frequency bands. The first octave band covers from  $f_{edge}$  to 0.5, and the last octave band covers from 0 to  $f_{edge}$

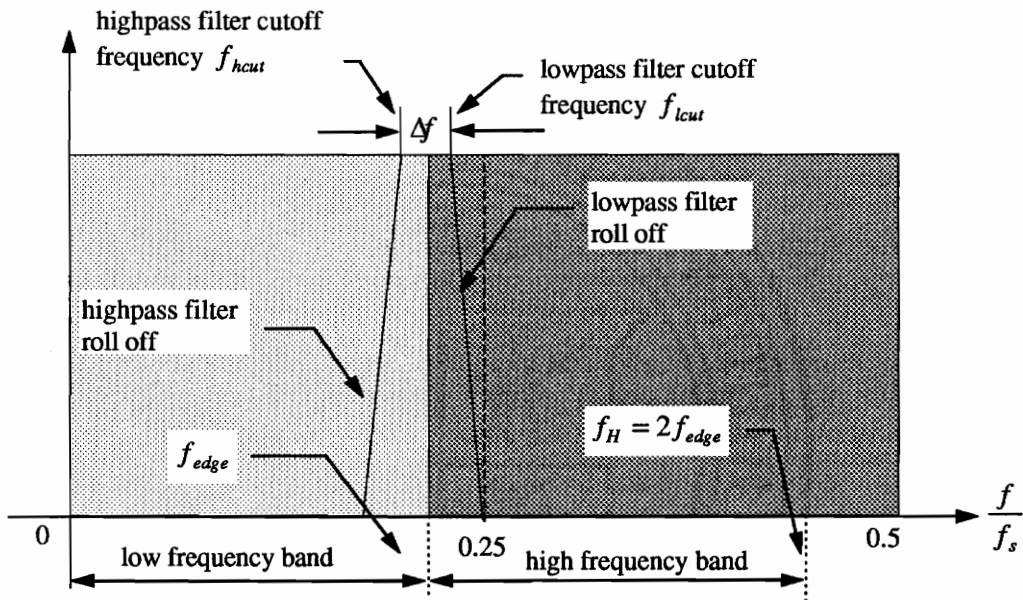
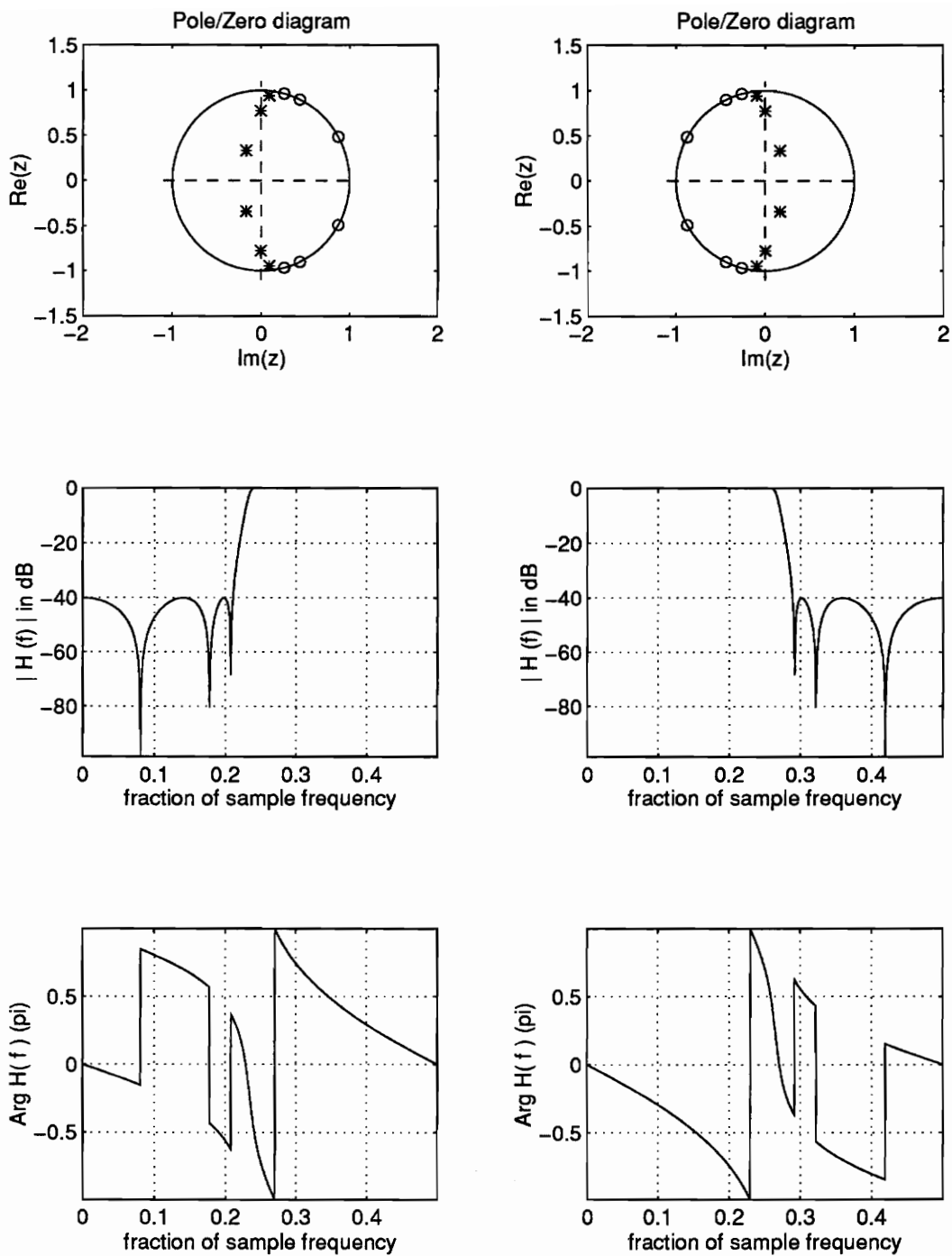
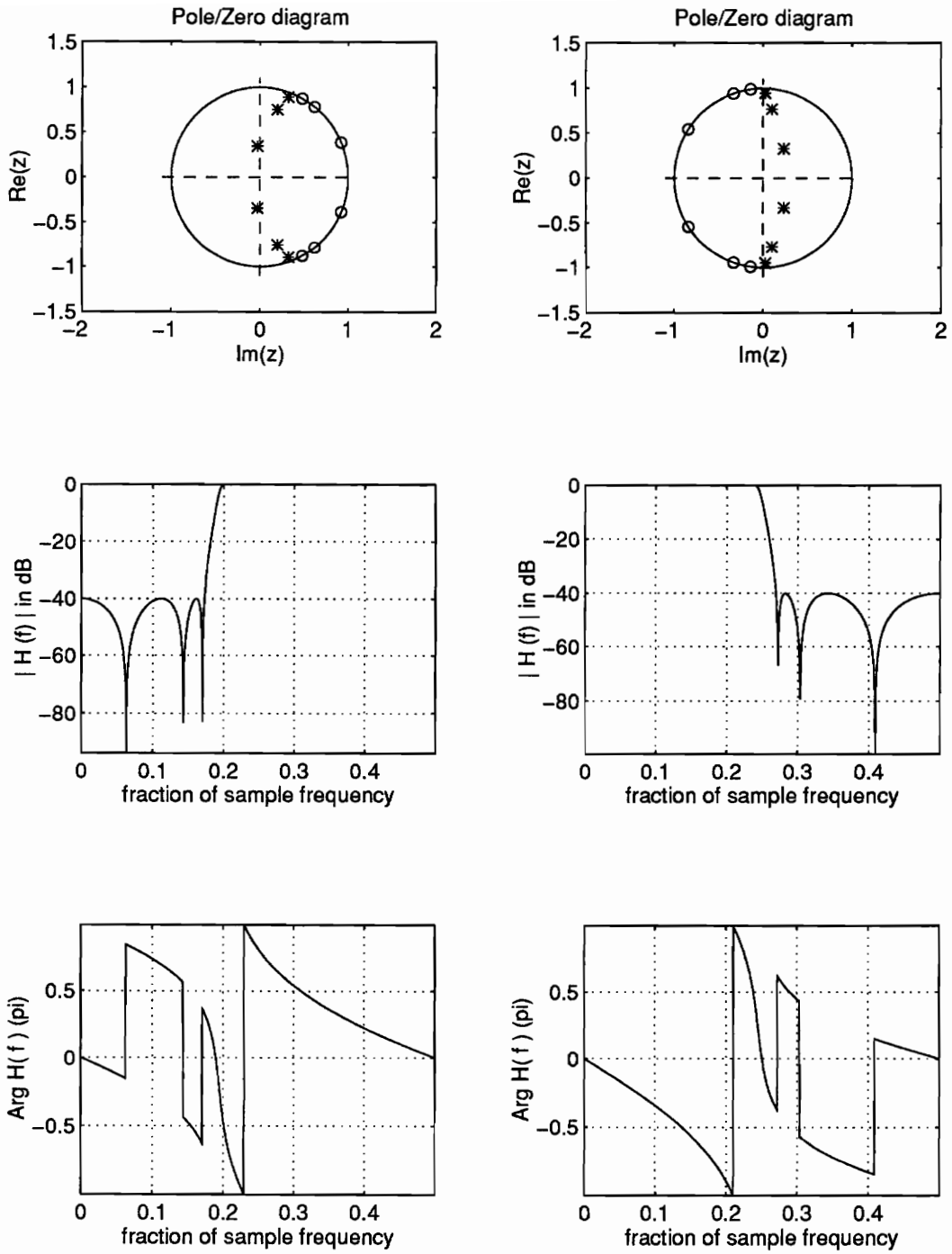


Figure 2.4 Asymmetric band-splitting filter design specifications.

Two experiments are designed to test and compare the performance of the symmetric and asymmetric band-splitting filters. In the symmetric design, the cutoff frequency for the highpass filter is 0.26 (fraction of sample frequency). In the asymmetric design, the cutoff frequencies for the highpass and lowpass filters are chosen as  $f_{hcut} = 0.2$  and  $f_{lcut} = 0.24$  for both experiments. The related  $f_{edge} = 0.22$  and  $f_H = 0.44$ . Both the symmetric and asymmetric filters are of the 6th order elliptic IIR form. Figure 2.5 a-f shows the pole-zero plots and the frequency characteristics of symmetric band-splitting filters, and Figure 2.6 a-f provides the same information for the asymmetric band-splitting filter.

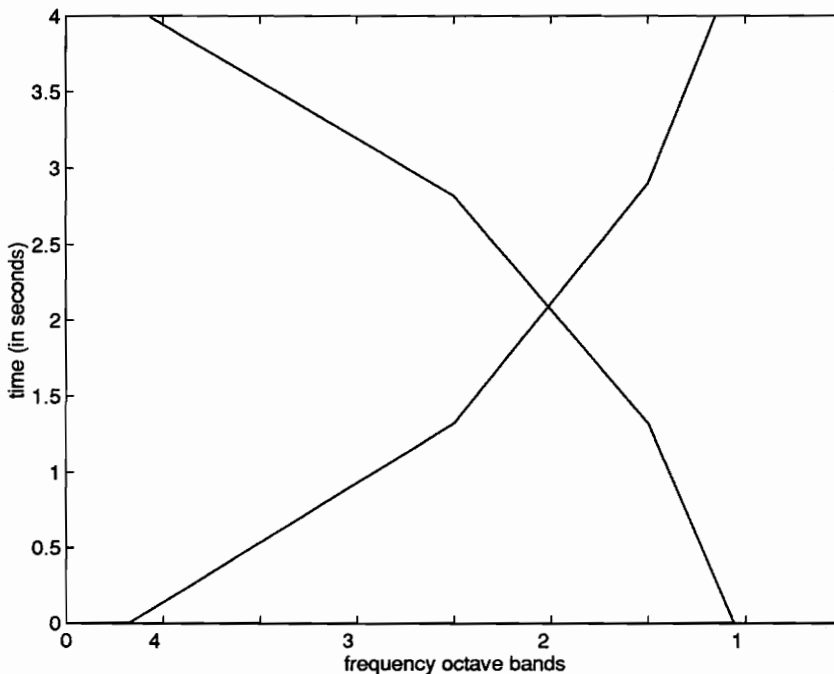


**Figure 2.5** Frequency characteristics for symmetric band-splitting filters (2.5a,b,c in the first column, and 2.5d,e,f in the second column).

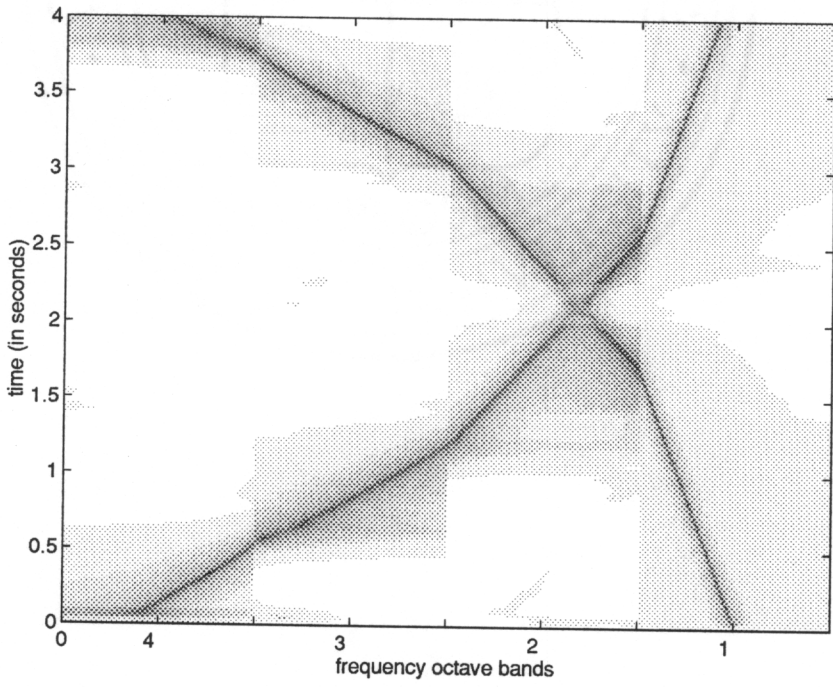
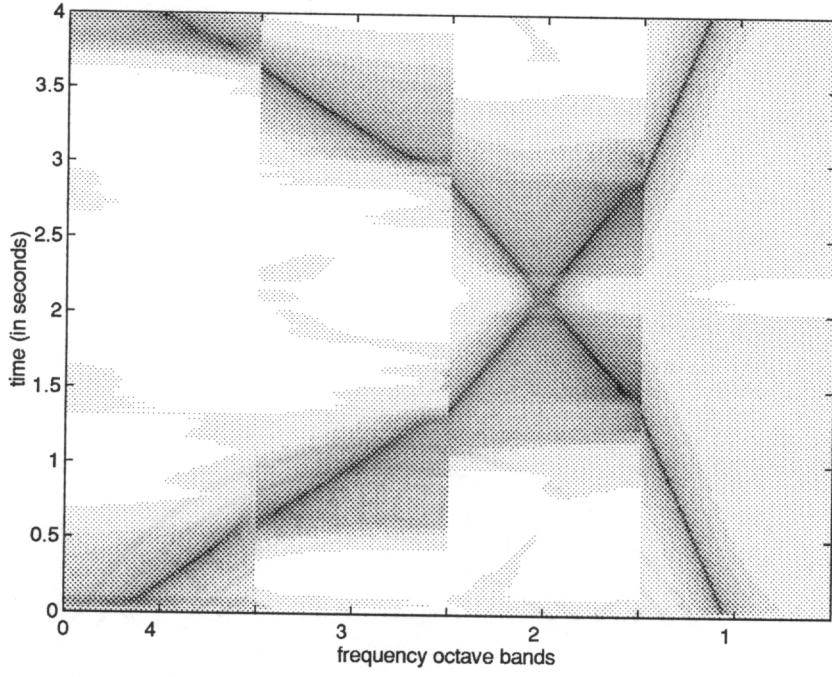


**Figure 2.6** Frequency characteristics for asymmetric band-splitting filters (2.6a,b,c in the first column, and 2.6d,e,f in the second column).

The test signal in the first experiment is composed of two chirp signals. The central frequency of these two signal components changes in an opposite direction, one varies from low to high frequency and the other changes from high to low frequency (Figure 2.7a). In Figure 2.7a, the bending of the frequency traces at the boundaries between the first and the second octave bands; and between the second and third octave bands is the result of decimation and frequency rescaling. Since the sub-sampling frequencies in the last two octave bands, the third and the fourth, are the same, there is no bending of the frequency trace between them. Figure 2.7b,c provides the MPSE estimates based on symmetric and asymmetric band-splitting filters respectively. From the graphics, one can easily see that there are frequency gaps between each octave band in the symmetric filter design, while the asymmetric band-splitting filters have overcome this problem. However, there is a visible frequency banding between the last two octave bands which is caused by the nonlinear phase characteristics of the IIR filters. The group delay of the nonlinear phase IIR filter is not constant, especially at the edges of its passband, where the group delay is larger than in the middle of the passband. This frequency bending can be corrected after knowing the phase characteristics of the IIR filters, or can be avoided by using FIR filter designs.

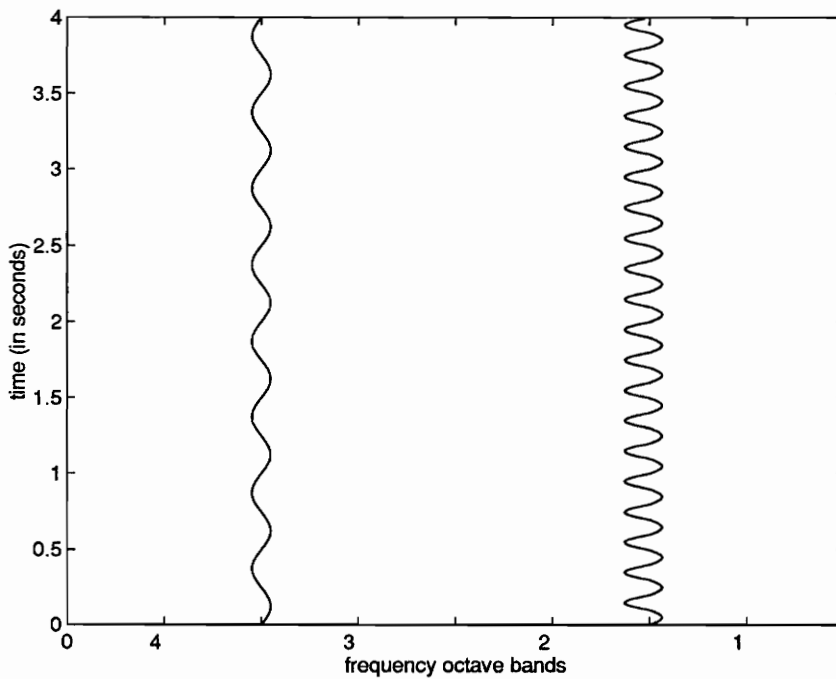


**Figure 2.7a** Two chirp signals used for testing symmetric and asymmetric filters.

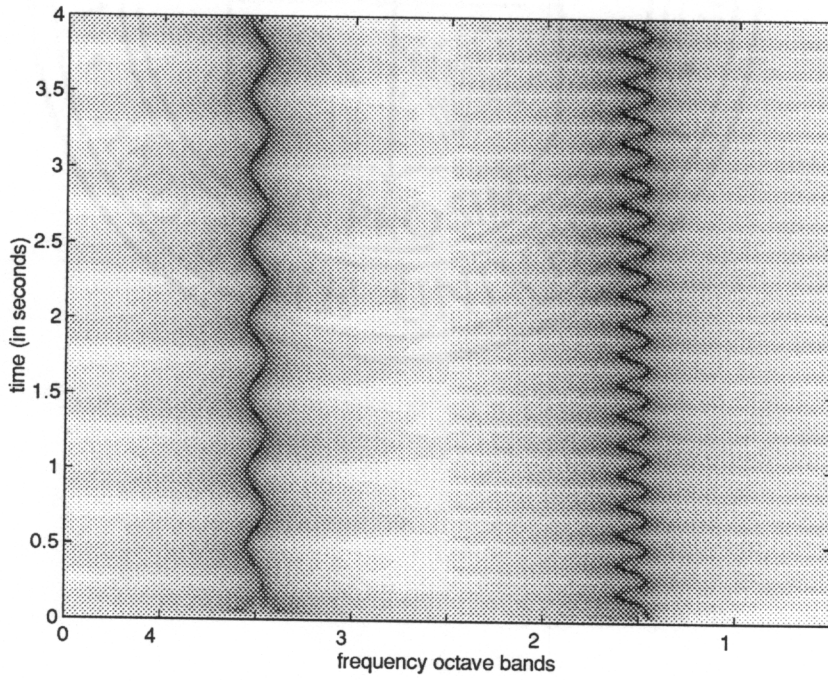
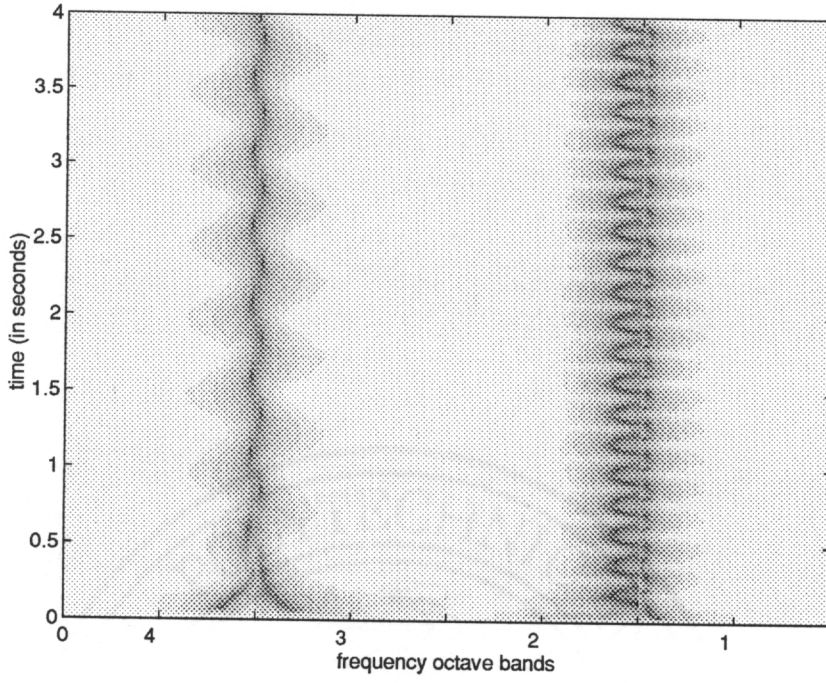


**Figure 2.7b,c** Performance of the symmetric (top) and asymmetric (bottom) band-splitting filters.

The test signal in the second experiment consists of two FM signals modulated by sinusoids (Figure 2.8a). The carrier frequencies are chosen such that one is located at the edge between the first and second octave bands while the other is located at the boundary of the third and fourth octave bands. Since the signal energies are concentrated in the boundary area, the symmetric band-splitting filters suffer significant signal information loss (Figure 2.8b). The asymmetric band-splitting filters avoid such edge information loss (Figure 2.8c) and provide excellent estimates of the test signal.



**Figure 2.8a** Two FM signals used for testing symmetric and asymmetric filters.



**Figure 2.8b,c** Performance of the symmetric (top) and asymmetric (bottom) band-splitting filters.



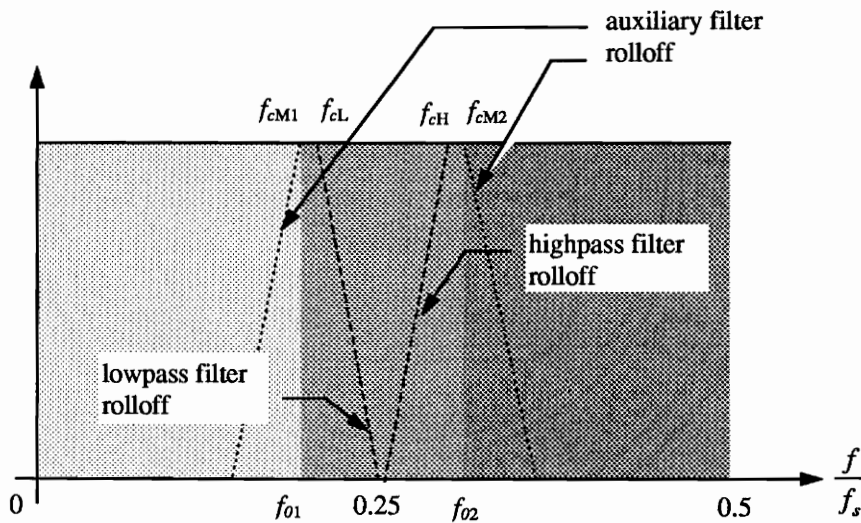
### 2.2.3 Auxiliary band-splitting filter design

Designing symmetric band-splitting filters gives the advantage of computational simplicity, while the use of asymmetric band-splitting filters largely overcomes the energy leakage at the boundaries of the octave bands. However, to prevent aliasing and to minimize the bandwidths of the filters without making gaps between the octave bands, these two kinds of band-splitting filters are designed to have as small a transition band as possible. That is, the frequency responses of the filters are close to the ideal filter, and thus they are filters of relatively high order (or long length). In Sections 3.3 and 3.4 we will see that the time resolution and the resolution cell size of such high order filters are usually poorer than for low order ones. When the time resolution and the resolution cell size of the band-splitting filter is the main concern, an auxiliary bandpass filter can be used to reduce the requirement on the transition bands.

The auxiliary bandpass filter is placed between the highpass and lowpass band-splitting filters. In each of the band-splitting stages the signal is filtered into three frequency bands. Instead of making the cutoff frequencies of the highpass and lowpass filters, in the two band design, as nearly the same as possible, the auxiliary band-splitting filter design allows a large gap between the main band-splitting filters. Therefore, the transition bands of the highpass and lowpass filters can be broader than those in the two band design. The large gap is then occupied by the third, or auxiliary, bandpass filter with a slow rolloff in its transition bands as well. The slow rolloff or broad transition band provides the possibility of using lower order, or shorter length, band-splitting filters, which results in better resolutions. Figure 2.9 shows the specifications for an auxiliary band-splitting filter design. Since there is a large overlap in the boundary areas, the signal energy loss in the auxiliary band-splitting filter design is less than with the asymmetric design. Another advantage of the auxiliary band-splitting filter design is the improvement in signal-to-noise ratio within every frequency band; since the bandwidths of the band-splitting filters are narrower than in two band designs, the spectral estimators can be focused on a smaller frequency region. On the other hand, these improvements are paid for by an almost doubling of the computations in performing the MPSE, since a middle frequency band is introduced in every band-splitting stage, requiring extra filtering and spectral estimation.

In the auxiliary band-splitting filter design, the outputs of the high and middle band are forwarded to AR spectral estimators, and the output of the lowpass filter is forwarded to the next

band-splitting stage. Except in the first octave band, the spectral estimator of the highpass filter analyzes the frequency region from  $f_{02}$  to  $2f_{01}$ , where the lower bound  $f_{02} = (f_{cH} + f_{cM2})/2$  and the upper bound is rescaled from the upper bound,  $f_{01} = (f_{cM1} + f_{cL})/2$ , of the lowpass filter in the last band-splitting operation (see Figure 2.9). The spectral estimator of the auxiliary filter analyzes the frequency band from  $f_{01}$  to  $f_{02}$  (see Figure 2.9). The frequency band of the AR estimator for the highpass filter of the first octave band is from  $f_{02}$  to 0.5 (as a fraction of the sampling frequency), and the frequency band of the AR estimator for the lowpass filter in the last octave band is from 0 to  $f_{01}$ . Since the cutoff frequencies of the main band-splitting filters, highpass and lowpass, do not overlap, they can be designed to be symmetric, which leads to identical filter coefficients with alternating signs. Such a design can reduce the computational load.



**Figure 2.9** Auxiliary band-splitting filter design specifications.

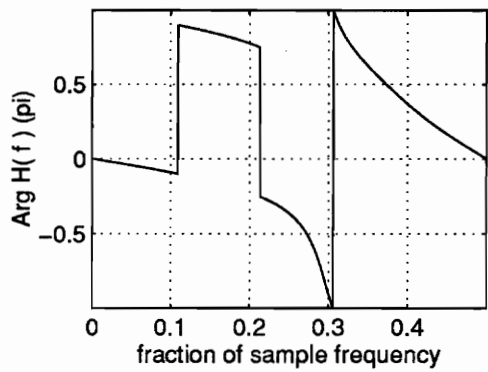
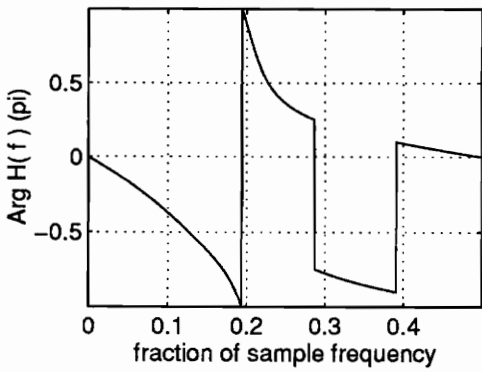
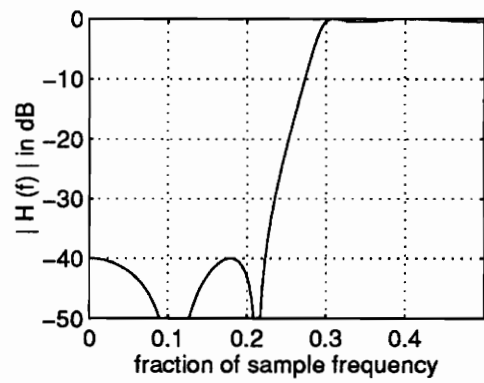
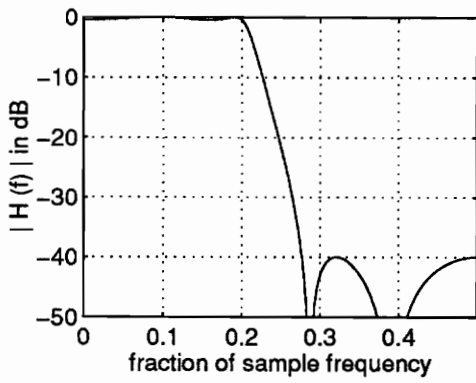
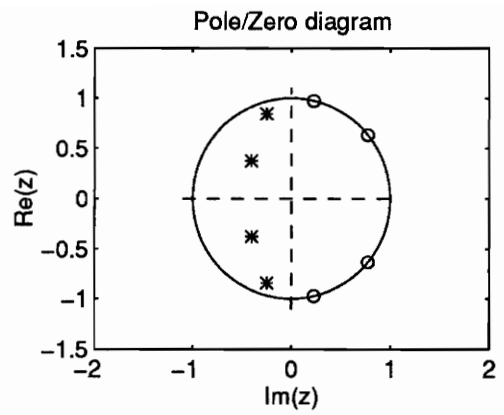
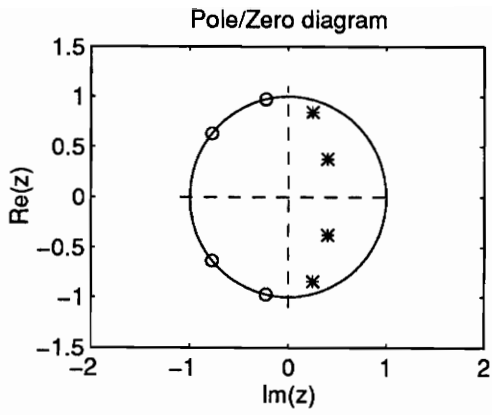
To compare the resolution characteristics of the three band-splitting filter designs, the resolutions of these filters are evaluated. The computational details for these resolutions will be discussed in Section 3.1. The parameters of the symmetric and asymmetric band-splitting filters are the same as those used in Section 2.2.2. They are all 6th order elliptic IIR filters. In the auxiliary band-splitting

filter design, the cutoff frequency of the highpass filter is 0.3 (as a fraction of the sampling frequency); the cutoff frequency of the low pass filter is 0.2; the cutoff frequencies of the auxiliary filter are 0.18 and 0.32. The highpass and lowpass band-splitting filters are now fourth order elliptic IIR filters while the auxiliary bandpass filter is a sixth order elliptic IIR filter. Figure 2.10 provides pole/zero plots and frequency characteristics for these filters. The results of the evaluation of the resolutions for the three band-splitting filter designs are given in Table 2.1.

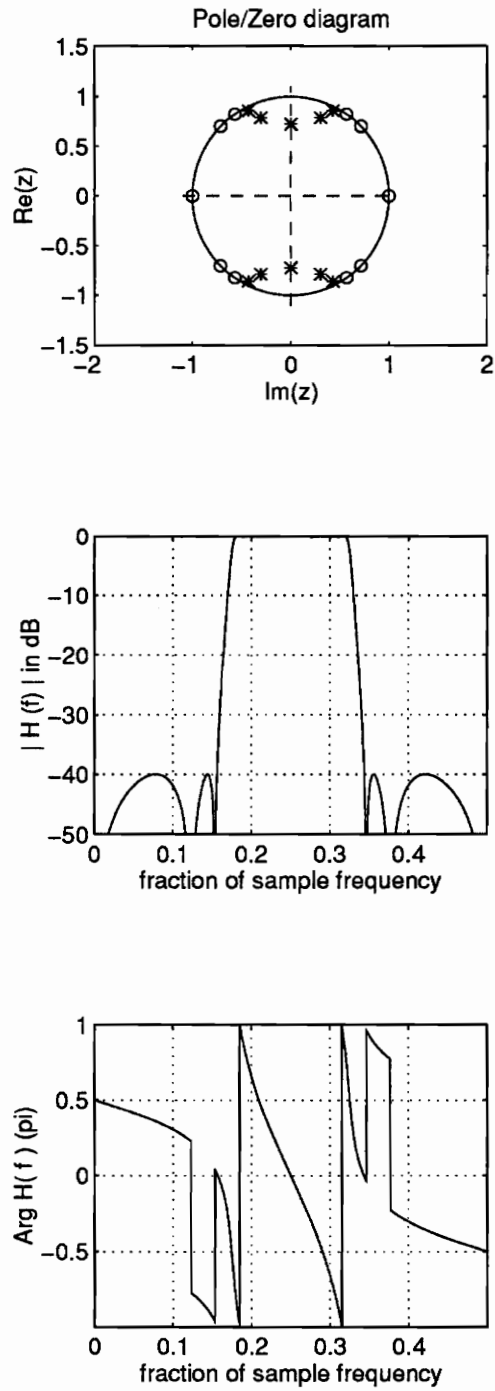
**Table 2.1 Resolutions of three different designs of band-splitting filters.**

Resolution	Auxiliary band-splitting filter design			Symmetric design		Asymmetric design	
	Lowpass	Bandpass	Highpass	Lowpass	Highpass	Lowpass	Highpass
$\Delta t$ (samples)	6.3721	4.8040	4.8040	11.2019	11.2019	11.4519	11.0866
$\Delta f$ (cycles/sample)	0.1384	0.3300	0.3300	0.3875	0.3875	0.3583	0.4448
$\Delta c$	0.8821	1.5855	1.5855	4.3410	4.3410	4.1030	4.9314

The time resolutions and the resolution cell sizes for the symmetric and asymmetric band-splitting filter design approaches are almost three times as large as for the auxiliary approach. The frequency resolutions of the main band-splitting filters in all three designs are closely matched. The auxiliary bandpass filter has the best time resolution, frequency resolution, and resolution cell size of all the band-splitting filters.



**Figure 2.10a,b** Frequency characteristics for main band-splitting filters in auxiliary band-splitting filter design (2.10a,b,c in the first column, and 2.10d,e,f in the second column).



**Figure 2.10c** Frequency characteristics of the auxiliary bandpass filter in auxiliary band-splitting filter design.

## 2.3 Realization of AR-PSE in MPSE

### 2.3.1 Overview

Theory and algorithms for AR-PSE assume a (wide sense) stationary process. However, signals to be processed by MPSE are assumed to have time-varying characteristics. To overcome the nonstationarity, a time window is applied. The window should be short enough to ensure pseudo-stationarity of the input. In a high SNR situation, a short window suffices to estimate the AR parameters. In a low SNR situation, the window must be long enough to yield acceptable variance of the estimator. As always, a trade off between estimation variance and nonstationarity inference is required.

There are two basic approaches to AR-PSE. One is the block estimation approach, such as in the Burg and Covariance algorithms. The incoming data are divided into individual blocks of length  $N_w$ . The parameters are estimated according to information within each block, and typically vary from one block to another. The output rate of the estimator is one  $N_w$ -th of its input rate, if the blocks are non-overlapping. To observe a continuous change in the spectrum, spectral estimation can alternatively be approached recursively, such as in the Recursive Least Squares (RLS) algorithm, where a sliding window is applied to the incoming data. The advantage of a block algorithm is the flexibility of selecting different time windows to improve performance in a nonstationary environment. However, block algorithms usually require more computation and storage space than recursive algorithms. The computational complexity of a block algorithm is  $O(N_w^2)$ . The recursive algorithm can update its parameter estimates more efficiently. Its computational complexity is  $O(\rho^2)$ , where  $\rho$  is the AR model order ( $\rho \ll N_w$ ). However, the recursive algorithm has a limitation working in a nonstationary environment, in that it can only apply an exponential window to the time series, controlled by the forgetting factor, which not only has an infinitely long tail but also penalizes near present data too much (see Figure 2.11a,b). To keep the computational simplicity of RLS, and improve its window function, we use a Windowed Recursive Least Squares (WRLS) algorithm for MPSE. WRLS applies an additional rectangular window to the exponential window to "forget" past data completely (see Figure 2.11). The computational complexity for WRLS is still  $O(\rho^2)$ .

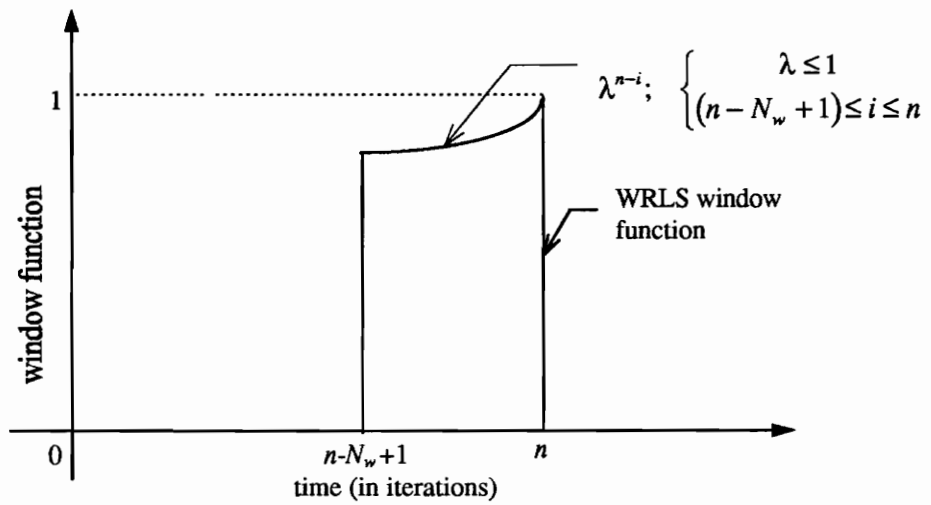
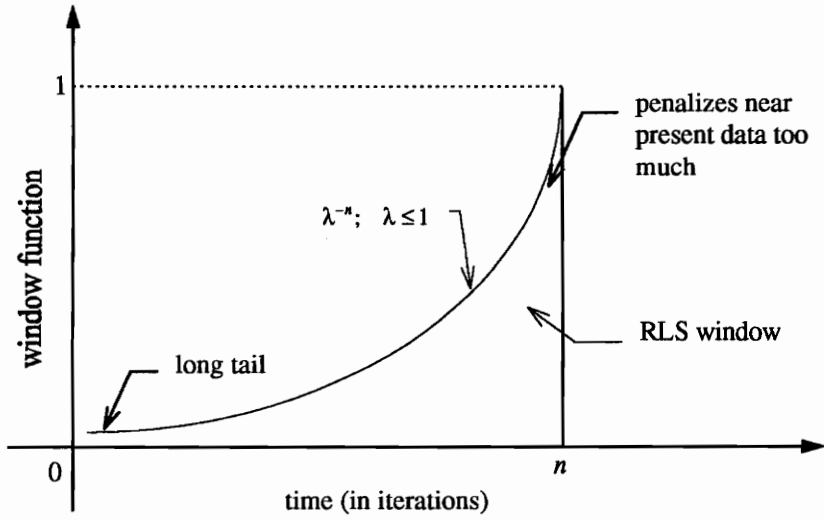


Figure 2.11a,b The window functions of RLS (top) and WRLS (bottom).

The limitations of the exponential window in the RLS have been well known and studied [103]. It is an essential problem for all those algorithms based on RLS with the exponential forgetting coefficient as the only way of handling nonstationarity. Therefore, the importance of the newly developed WRLS algorithm is not only for improving the performance of MPSE but also for the general purposes of nonstationary adaptive filtering, time-varying system identification, and nonstationary signal spectral estimation. Since the RLS algorithm is the basic core for many fast adaptive filtering algorithms [104] and system identification algorithms [105], as well as for array processing algorithms such as the systolic array algorithms [104, 106], there will exist equivalent algorithms for the WRLS.

The WRLS algorithm was first derived as part of this dissertation research. Later I found the same approach in the literature [101], but the only publication that I found provides no detail about the algorithm and its derivation. Thus, in this section a complete derivation is provided. A summary and a signal flow graph of the algorithm can be found in Table 2.2 and Figure 2.13 respectively. Meanwhile, in this section the performance of the RLS and WRLS algorithms will be compared under nonstationary conditions; the special issue of how to properly initialize the WRLS algorithm will also be discussed.

### 2.3.2 Derivation of the WRLS algorithm

The recursive least squares algorithm provides a systematic method for updating of the adaptive transversal filter and to estimate the desired signal  $d(n)$  based on the least squares criterion. At time  $n$  the desired signal  $d(n)$  is estimated by an adaptive transversal filter using a linear combination of the present observation  $u(n)$  and  $m$  previous observations  $u(n-i)$ ,  $i= 1, 2, \dots, m$ , as in Figure 2.12. That is, the estimate  $y(n)$  of the desired signal can be represented as:

$$y(n) = \mathbf{w}^H(n)\mathbf{u}(n) \quad (2.3)$$

where the superscript  $H$  denotes the Hermitian transpose,  $\mathbf{u}(n)$  is the observation vector, and  $\mathbf{w}(n)$  is the tap-weight vector of the transversal filter at time  $n$ , which are defined as

$$\mathbf{u}(i) = [u(i), u(i-1), \dots, u(i-\rho+1)]^T \quad \text{for } i= 0, 1, 2, \dots, n \quad (2.4a)$$



$$\mathbf{w}(i) = [w_0(n), w_1(n), \dots, w_\rho(n)]^T \quad (2.4b)$$

The superscript  $T$  in above equations denotes the transpose and  $\rho$  is the order or the length of the transversal filter.

As in Figure 2.11a, the cost function  $\xi(n)$  in the RLS algorithm is minimized over the entire observation interval,  $[0, n]$ , weighted by the forgetting coefficient  $\lambda$ ,  $0 < \lambda \leq 1$ , i.e.

$$\xi(n) = \sum_{i=0}^n \lambda^{n-i} |d(i) - y(i)|^2 \quad (2.5)$$

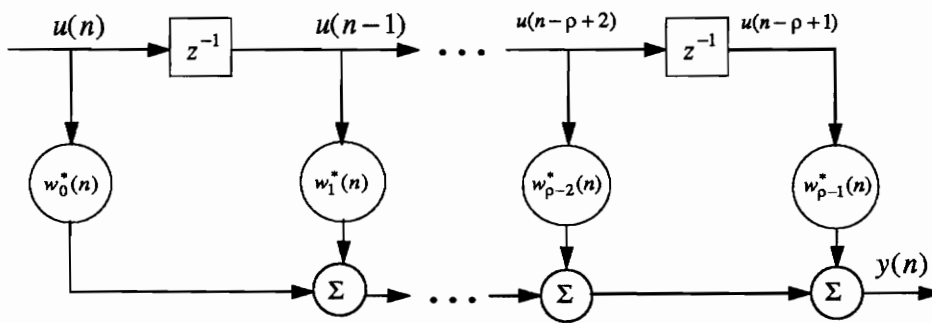


Figure 2.12 Transversal filter.

The optimum value of the tap-weight vector,  $\hat{\mathbf{w}}(n)$ , is thus defined by the *normal equations*, in matrix form [104]:

$$\Phi(n)\hat{\mathbf{w}}(n) = \mathbf{p}(n) \quad (2.6)$$

where  $\Phi(n)$  is the  $\rho \times \rho$  exponentially weighted correlation matrix defined by

$$\Phi(n) = \sum_{i=0}^n \lambda^{n-i} \mathbf{u}(i)\mathbf{u}^H(i) \quad (2.7a)$$

and where  $\mathbf{p}(n)$  is the  $\rho \times 1$  exponentially weighted cross-correlation vector defined by

$$\mathbf{p}(n) = \sum_{i=0}^n \lambda^{n-i} \mathbf{u}(n) d^*(i) \quad (2.7b)$$

The asterisk denotes complex conjugation.

The new WRLS algorithm introduces a rectangular window in addition to the exponential window of the RLS algorithm. The rectangular window cuts the past information abruptly while the exponential window is used only to penalize the near present information. The cost function for the WRLS algorithm is defined by

$$\xi(n) = \sum_{i=n-N_w+1}^n \lambda^{n-i} |d(i) - y(i)|^2 \quad (2.8)$$

where  $N_w$  is the rectangular window width (see Figure 2.11b), and  $N_w \gg \rho$ . The optimum value of the tap-weight vector,  $\hat{\mathbf{w}}(n)$ , has the same matrix form as in (2.6). However, the correlation matrix in (2.6) is redefined by

$$\Phi(n) = \sum_{i=n-N_w+1}^n \lambda^{n-i} \mathbf{u}(i) \mathbf{u}^H(i) \quad (2.9a)$$

and the cross-correlation vector is redefined by

$$\mathbf{p}(n) = \sum_{i=n-N_w+1}^n \lambda^{n-i} \mathbf{u}(i) d^*(i) \quad (2.9b)$$

From (2.9a) the correlation matrix  $\Phi(n)$  can be rewritten in terms of  $\Phi(n-1)$  and the new observation vector  $\mathbf{u}(n)$ , i.e.

$$\begin{aligned} \Phi(n) &= \lambda \sum_{i=n-N_w}^{n-1} \lambda^{n-i-1} \mathbf{u}(i) \mathbf{u}^H(i) - \lambda^{N_w} \mathbf{u}(n-N_w) \mathbf{u}^H(n-N_w) + \mathbf{u}(n) \mathbf{u}^H(n) \\ &= \lambda \Phi(n-1) - \lambda^{N_w} \mathbf{u}(n-N_w) \mathbf{u}^H(n-N_w) + \mathbf{u}(n) \mathbf{u}^H(n) \end{aligned} \quad (2.10)$$

From (2.9b) the cross-correlation vector  $\mathbf{p}(n)$  can also be represented in terms of  $\mathbf{p}(n-1)$  and the new observation vector  $\mathbf{u}(n)$  and desired signal  $d(n)$ , i.e.

$$\begin{aligned}\mathbf{p}(n) &= \lambda \sum_{i=n-N_w}^{n-1} \lambda^{n-i-1} \mathbf{u}(n) d^*(i) - \lambda^{N_w} \mathbf{u}(n-N_w) d^*(n-N_w) + \mathbf{u}(n) d^*(n) \\ &= \lambda \mathbf{p}(n-1) - \lambda^{N_w} \mathbf{u}(n-N_w) d^*(n-N_w) + \mathbf{u}(n) d^*(n)\end{aligned}\quad (2.11)$$

The derivation of the WRLS algorithm makes use of the *matrix inversion lemma* which can be represented as follows [107]:

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C} \mathbf{D} \mathbf{C}^H \quad (2.12a)$$

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B} \mathbf{C} (\mathbf{D} + \mathbf{C}^H \mathbf{B} \mathbf{C})^{-1} \mathbf{C}^H \mathbf{B} \quad (2.12b)$$

First we define the first two terms of (2.10) as the matrix inverse of  $\Psi(n)$ , i.e.

$$\Psi^{-1}(n) = \lambda \Phi(n-1) - \lambda^{N_w} \mathbf{u}(n-N_w) \mathbf{u}^H(n-N_w) \quad (2.13)$$

Since the correlation matrix  $\Phi(n-1)$  and  $\mathbf{u}(n-N_w) \mathbf{u}^H(n-N_w)$  are positive definite, the  $\rho \times \rho$  matrix  $\Psi(n)$  can be solved by letting

$$\begin{aligned}\mathbf{A} &= \Psi^{-1}(n) \\ \mathbf{B}^{-1} &= \lambda \Phi(n-1) \\ \mathbf{C} &= \lambda^{\frac{N_w}{2}} \mathbf{u}(n-N_w) \\ \mathbf{D} &= -1\end{aligned}\quad (2.14)$$

Then, substituting (2.14) into the matrix inverse lemma of (2.12b), we obtain the inverse of (2.13)

$$\begin{aligned}\Psi(n) &= \mathbf{A}^{-1} \\ &= \lambda^{-1} \Phi^{-1}(n-1) - \frac{\lambda^{N_w-2} \Phi^{-1}(n-1) \mathbf{u}(n-N_w) \mathbf{u}^H(n-N_w) \Phi^{-1}(n-1)}{\lambda^{N_w-1} \mathbf{u}^H(n-N_w) \Phi^{-1}(n-1) \mathbf{u}(n-N_w) - 1}\end{aligned}\quad (2.15)$$

For simplicity let

$$\mathbf{P}(n) = \Phi^{-1}(n) \quad (2.16a)$$

$$\mathbf{h}(n) = \frac{\lambda^{N_w-1} \mathbf{P}(n-1) \mathbf{u}(n - N_w)}{\lambda^{N_w-1} \mathbf{u}^H(n - N_w) \mathbf{P}(n-1) \mathbf{u}(n - N_w) - 1} \quad (2.16b)$$

Thus, (2.15) can be rewritten as:

$$\begin{aligned} \Psi(n) &= \lambda^{-1} \mathbf{P}(n-1) - \lambda^{-1} \mathbf{h}(n) \mathbf{u}^H(n - N_w) \mathbf{P}(n-1) \\ &= \lambda^{-1} [\mathbf{I} - \mathbf{h}(n) \mathbf{u}^H(n - N_w)] \mathbf{P}(n-1) \end{aligned} \quad (2.17)$$

where  $\mathbf{I}$  is the identity matrix of size  $\rho \times \rho$ . Using (2.10) and (2.13), we have

$$\Phi(n) = \Psi^{-1}(n) + \mathbf{u}(n) \mathbf{u}^H(n) \quad (2.18)$$

Letting

$$\begin{aligned} \mathbf{A} &= \Phi(n) \\ \mathbf{B} &= \Psi(n) \\ \mathbf{C} &= \mathbf{u}(n) \\ \mathbf{D} &= 1 \end{aligned} \quad (2.19)$$

we obtain the inverse of the correlation matrix by applying the matrix inversion lemma to (2.18):

$$\begin{aligned} \mathbf{P}(n) &= \Phi^{-1}(n) = \mathbf{A}^{-1} \\ &= \Psi(n) - \frac{\Psi(n) \mathbf{u}(n) \mathbf{u}^H(n) \Psi(n)}{\mathbf{u}^H(n) \Psi(n) \mathbf{u}(n) + 1} \end{aligned} \quad (2.20)$$

Let

$$\mathbf{k}(n) = \frac{\Psi(n) \mathbf{u}(n)}{\mathbf{u}^H(n) \Psi(n) \mathbf{u}(n) + 1} \quad (2.21)$$

Equation (2.20) can then be rewritten as:

$$\begin{aligned}
\mathbf{P}(n) &= \Psi(n) - \mathbf{k}(n)\mathbf{u}^H(n)\Psi(n) \\
&= [\mathbf{I} - \mathbf{k}(n)\mathbf{u}^H(n)]\Psi(n)
\end{aligned} \tag{2.22}$$

From (2.21) we have

$$\mathbf{k}(n)[1 + \mathbf{u}^H(n)\Psi(n)\mathbf{u}(n)] = \Psi(n)\mathbf{u}(n) \tag{2.23}$$

Using (2.22), we can rewrite  $\mathbf{k}(n)$  as follows:

$$\begin{aligned}
\mathbf{k}(n) &= \Psi(n)\mathbf{u}(n) - \mathbf{k}(n)\mathbf{u}^H(n)\Psi(n)\mathbf{u}(n) \\
&= [\Psi(n) - \mathbf{k}(n)\mathbf{u}^H(n)\Psi(n)]\mathbf{u}(n) \\
&= \mathbf{P}(n)\mathbf{u}(n)
\end{aligned} \tag{2.24}$$

Equation (2.16b) can be rewritten as:

$$\mathbf{h}(n)[\lambda^{N_w-1}\mathbf{u}^H(n-N_w)\mathbf{P}(n-1)\mathbf{u}(n-N_w) - 1] = \lambda^{N_w-1}\mathbf{P}(n-1)\mathbf{u}(n-N_w) \tag{2.25a}$$

Using (2.17),  $\mathbf{h}(n)$  can be reorganized as

$$\begin{aligned}
\mathbf{h}(n) &= -\lambda^{N_w}[\lambda^{-1}\mathbf{P}(n) - \lambda^{-1}\mathbf{h}(n)\mathbf{u}^H(n-N_w)\mathbf{P}(n-1)]\mathbf{u}(n-N_w) \\
&= -\lambda^{N_w}\Psi(n)\mathbf{u}(n-N_w)
\end{aligned} \tag{2.25b}$$

Using (2.6) and (2.11), we can estimate the optimal tap-weight vector from

$$\begin{aligned}
\hat{\mathbf{w}}(n) &= \Phi^{-1}(n)\mathbf{p}(n) \\
&= \mathbf{P}(n)[\lambda\mathbf{p}(n-1) - \lambda^{N_w}\mathbf{u}(n-N_w)d^*(n-N_w) + \mathbf{u}(n)d^*(n)] \\
&= \lambda\mathbf{P}(n)\mathbf{p}(n-1) - \lambda^{N_w}\mathbf{P}(n)\mathbf{u}(n-N_w)d^*(n-N_w) + \mathbf{P}(n)\mathbf{u}(n)d^*(n)
\end{aligned} \tag{2.26}$$

From (2.22) and (2.17) the first term of (2.26) can be written as:

$$\begin{aligned}
\lambda \mathbf{P}(n)\mathbf{p}(n-1) &= [\mathbf{I} - \mathbf{k}(n)\mathbf{u}^H(n)] [\mathbf{I} - \mathbf{h}(n)\mathbf{u}^H(n - N_w)] \hat{\mathbf{w}}(n-1) \\
&= \hat{\mathbf{w}}(n-1) - \mathbf{k}(n)\mathbf{u}^H(n)\hat{\mathbf{w}}(n-1) - \mathbf{h}(n)\mathbf{u}^H(n - N_w)\hat{\mathbf{w}}(n-1) \\
&\quad + \mathbf{k}(n)\mathbf{u}^H(n)\mathbf{h}(n)\mathbf{u}^H(n - N_w)\hat{\mathbf{w}}(n-1)
\end{aligned} \tag{2.27a}$$

From (2.22) and (2.25b) the second term of (2.26) can be rewritten as:

$$\begin{aligned}
-\lambda^{N_w} \mathbf{P}(n)\mathbf{u}(n - N_w)d^*(n - N_w) &= -\lambda^{N_w} [\mathbf{I} - \mathbf{k}(n)\mathbf{u}^H(n)] \Psi(n)\mathbf{u}(n - N_w)d^*(n - N_w) \\
&= [\mathbf{I} - \mathbf{k}(n)\mathbf{u}^H(n)] \mathbf{h}(n)d^*(n - N_w)
\end{aligned} \tag{2.27b}$$

Using (2.24) the last term of (2.26) can be represented as:

$$\mathbf{P}(n)\mathbf{u}(n)d^*(n) = \mathbf{k}(n)d^*(n) \tag{2.27c}$$

Substituting (2.27a), (2.27b), and (2.27c) back into (2.26), we can reorganize (2.26) in the following form:

$$\begin{aligned}
\hat{\mathbf{w}}(n) &= \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)[d^*(n) - \mathbf{u}^H(n)\hat{\mathbf{w}}(n-1)] + \mathbf{h}(n)[d^*(n - N_w) \\
&\quad - \mathbf{u}^H(n - N_w)\hat{\mathbf{w}}(n-1)] - \mathbf{k}(n)\mathbf{u}^H(n)\mathbf{h}(n)[d^*(n - N_w) - \mathbf{u}^H(n - N_w)\hat{\mathbf{w}}(n-1)]
\end{aligned} \tag{2.28}$$

or

$$\begin{aligned}
\hat{\mathbf{w}}(n) &= \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)[d^*(n) - \mathbf{u}^H(n)\hat{\mathbf{w}}(n-1)] \\
&\quad + [\mathbf{I} - \mathbf{k}(n)\mathbf{u}^H(n)] \mathbf{h}(n) [d^*(n - N_w) - \mathbf{u}^H(n - N_w)\hat{\mathbf{w}}(n-1)]
\end{aligned} \tag{2.29}$$

We define:

$$\begin{aligned}
\alpha_1(n) &= d^*(n) - \mathbf{u}^H(n)\hat{\mathbf{w}}(n-1) \\
\alpha_2(n) &= d^*(n - N_w) - \mathbf{u}^H(n - N_w)\hat{\mathbf{w}}(n-1) \\
\mathbf{k}_1(n) &= \mathbf{k}(n) \\
\mathbf{k}_2(n) &= [\mathbf{I} - \mathbf{k}(n)\mathbf{u}^H(n)] \mathbf{h}(n)
\end{aligned} \tag{2.30}$$

Substituting (2.30) into (2.29), we obtain the following recursive equation for the tap-weight vector:

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \alpha_1(n)\mathbf{k}_1(n) + \alpha_2(n)\mathbf{k}_2(n) \quad (2.31)$$

Figure 2.13 provides the signal flow graph and Table 2.2 provides the summary for the WRLS algorithm.

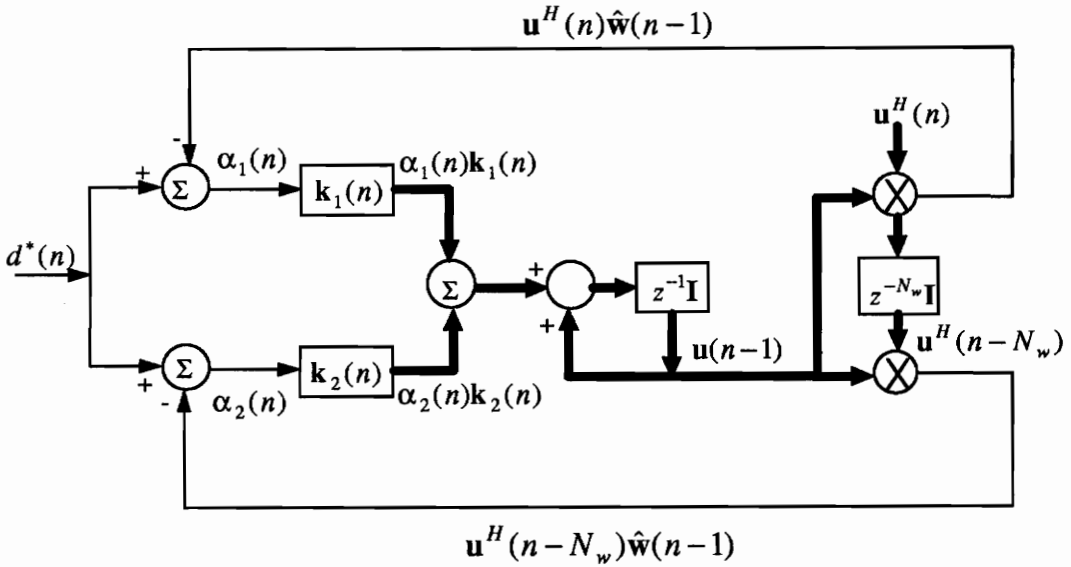


Figure 2.13 Signal flow graph of WRLS.

There is logical meaning associated with each of the newly defined variables in (2.30). The scalar  $\alpha_1(n)$  is the estimation error generated by using the inner product  $\mathbf{u}^H(n)\hat{\mathbf{w}}(n-1)$  to estimate the desired signal  $d(n)$ . Since the estimation uses the old tap-weight vector  $\hat{\mathbf{w}}(n-1)$  and the desired signal  $d(n)$  is at the right end of the sliding window, the estimation error  $\alpha_1(n)$  is called the *a priori forward estimation error*. The scalar  $\alpha_2(n)$  is the *a priori backward estimation error*, since it is the estimation error produced when estimating the desired signal at  $d(n-N_w)$  by using the observation vectors  $\mathbf{u}(i)$ , for  $i = n-1, n, \dots, n-N_w+1$ .  $\mathbf{k}_1(n)$  is the forward error feedback gain vector, and  $\mathbf{k}_2(n)$  is the backward error feedback gain vector.

Table 2.2 Summary of the WRLS Algorithm.

**Initialization**

starts from  $n = 0$  after padding  $N_w$  zeros to the observation at beginning.

$$\mathbf{P}(0) = \delta^{-1} \mathbf{I}$$

$$\mathbf{u}(0) = [u(0), 0, \dots, 0]^T \text{ and } \mathbf{u}(i) = \mathbf{0}, i = -1, -2, \dots, -N_w + 1$$

$$\hat{\mathbf{w}}(0) = \mathbf{0}$$

**For each instant of time,  $n = 1, 2, 3, \dots$  compute**

$$\mathbf{h}(n) = \frac{\lambda^{N_w-1} \mathbf{P}(n-1) \mathbf{u}(n-N_w)}{\lambda^{N_w-1} \mathbf{u}^H(n-N_w) \mathbf{P}(n-1) \mathbf{u}(n-N_w) - 1}$$

$$\Psi(n) = \lambda^{-1} \mathbf{P}(n-1) - \lambda^{-1} \mathbf{h}(n) \mathbf{u}^H(n-N_w) \mathbf{P}(n-1)$$

$$\mathbf{k}_1(n) = \frac{\Psi(n) \mathbf{u}(n)}{1 + \mathbf{u}^H(n) \Psi(n) \mathbf{u}(n)}$$

$$\mathbf{k}_2(n) = \mathbf{h}(n) - \mathbf{k}_1(n) \mathbf{u}^H(n) \mathbf{h}(n)$$

$$\mathbf{P}(n) = \Psi(n) - \mathbf{k}_1(n) \mathbf{u}^H(n) \Psi(n)$$

$$\alpha_1(n) = d^*(n) - \mathbf{u}^H(n) \hat{\mathbf{w}}(n-1)$$

$$\alpha_2(n) = d^*(n-N_w) - \mathbf{u}^H(n-N_w) \hat{\mathbf{w}}(n-1)$$

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}_1(n) \alpha_1(n) + \mathbf{k}_2(n) \alpha_2(n)$$

$$y(n) = \hat{\mathbf{w}}^H(n) \mathbf{u}(n)$$

$$e(n) = d(n) - y(n)$$



The transversal adaptive filter output, the *a posteriori* estimate  $y(n)$ , can be computed from

$$y(n) = \hat{\mathbf{w}}^H(n)\mathbf{u}(n) \quad (2.32a)$$

and the *a posteriori* estimation error  $e(n)$  can be computed from

$$e(n) = d(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n) \quad (2.32b)$$

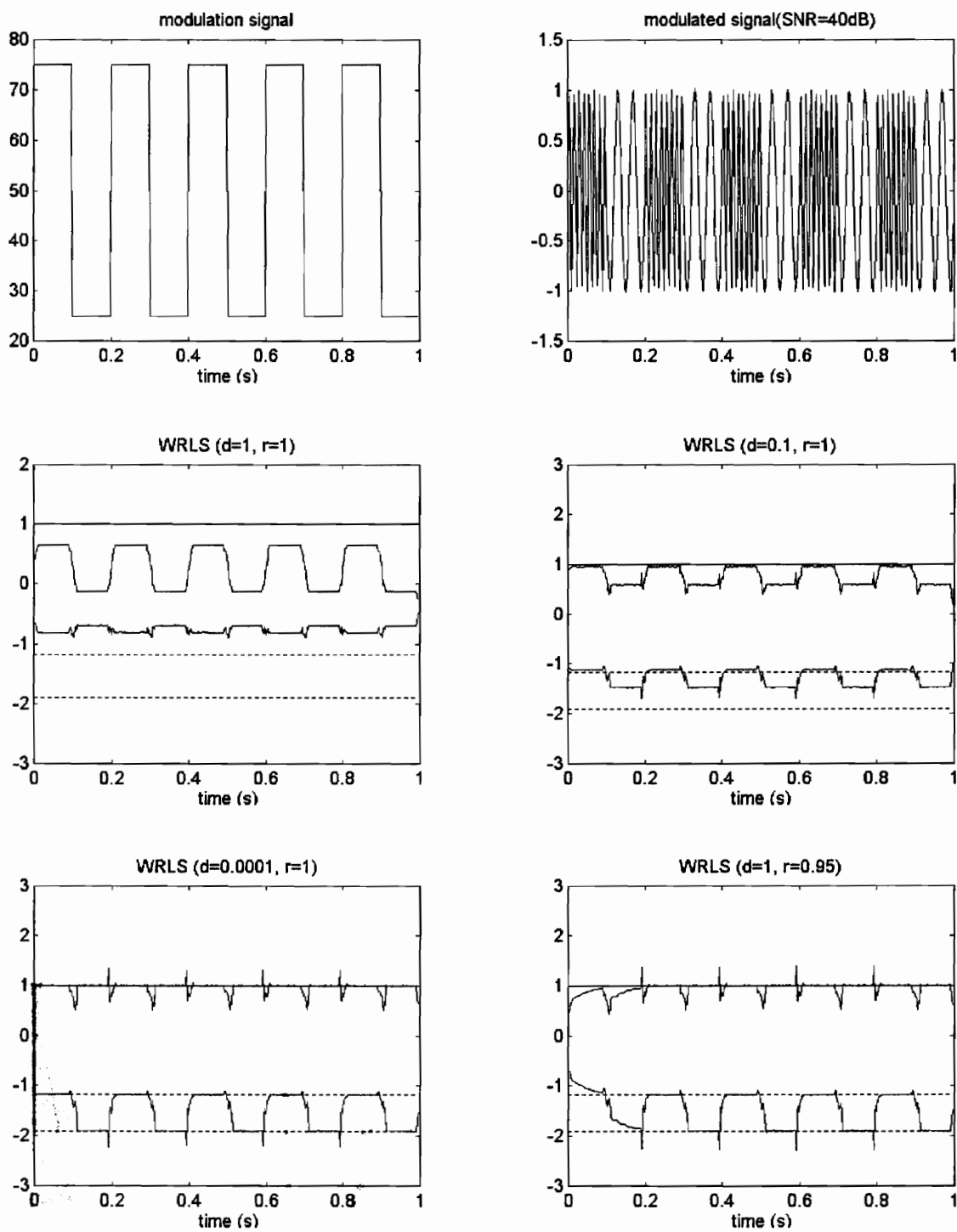
### 2.3.3 Initialization of the WRLS algorithm

The initialization of WRLS is an important step in the algorithm. Unlike for RLS, where the initialization condition only affects the initial performance of the adaptive filter, the initialization condition affects the performance of WRLS at all times if it is not started properly. The general initialization procedure for WRLS is similar to that for RLS. First, pad  $N_w - 1$  zeros to the observation such that  $\mathbf{u}(i) = \mathbf{0}$ ,  $i = -1, -2, \dots, -N_w + 1$ , and  $\mathbf{u}(0) = [\mathbf{u}(0), 0, \dots, 0]^T$ , where the  $\mathbf{0}$  is a  $\rho \times 1$  zero vector. Then, choose a small number  $\delta$  to build an initial correlation matrix and initialize the tap-weight vector at time  $n = 0$ , i.e.

$$\mathbf{P}(0) = \delta^{-1} \mathbf{I} \quad (2.33a)$$

$$\hat{\mathbf{w}}(0) = \mathbf{0} \quad (2.33b)$$

The special consideration that needs to be taken with WRLS is to choose the initial correlation matrix  $\Phi(0) = \delta \mathbf{I}$ . In the RLS algorithm, if the forgetting coefficient  $\lambda = 1$ , the effect of the initial correlation matrix will die out quickly as the power of the correlation matrix  $\Phi(n)$  accumulates with iterations. However, the power of the correlation matrix  $\Phi(n)$  in WRLS will not accumulate with iteration after  $n > N_w$ . Thus, the initialization matrix  $\delta \mathbf{I}$  will remain in the correlation matrix, which can produce biased estimates. Figure 2.14 provides examples of initialization with different combinations of  $\delta$  and forgetting coefficient  $\lambda$ . The signal is a single sine wave of amplitude  $A=1$  whose frequency is modulated by a square wave (Figure 2.14a), such that the signal frequency changes between 25 Hz and 75 Hz, with a period of 200 ms (Figure 2.14b). A real sine wave can be represented by a complex conjugate pair of poles on the unit circle, with corresponding AR parameters



**Figure 2.14 (a,b top row) Modulation and Modulated signals; (c-f) Estimates  $\hat{a}_1(t)$ ,  $\hat{a}_2(t)$  (solid lines) & true values  $a_1(t)$ ,  $a_2(t)$  (dotted lines).**

$$\{a_0, a_1, a_2\} = \begin{cases} \{1, -1.9020, 1\} & \text{for } f = 25 \text{ Hz} \\ \{1, -1.1755, 1\} & \text{for } f = 75 \text{ Hz} \end{cases} \quad (2.34)$$

Zero-mean white noise of variance  $\sigma_n^2$  is added to the sine wave to yield 40 dB SNR, defined as  $A^2/2\sigma_n^2$ . The time window used in WRLS is 10 sample points. In Figure 2.14c with  $\delta = 1$  and  $\lambda = 1$ , the model parameter estimates  $\hat{a}_1(t)$  and  $\hat{a}_2(t)$  (solid lines) have a large bias compared with the true values of the parameters (dotted lines). In Figure 2.14d with  $\delta = 0.1$  and  $\lambda = 1$ , the estimates  $\hat{a}_1(t)$  and  $\hat{a}_2(t)$  (solid lines) have a smaller bias than in the previous case. In Figure 2.14e with  $\delta = 0.0001$  and  $\lambda = 1$ , the estimation bias is invisible. Figure 2.14f provides an example with  $\delta = 1$  and  $\lambda = 0.95$ , where the estimation bias dies out exponentially. Therefore, to prevent estimation bias from occurring, special care is needed, such as setting the forgetting coefficient  $\lambda$  to less than 1 in the initialization period, and/or making  $\delta$  small enough.

### 2.3.4 Performance of WRLS in a nonstationary environment

The experiment is designed to compare the performance, under noisy conditions, of the WRLS and conventional RLS algorithms for AR estimation. The signal is the same as that in the initialization experiment (Figure 2.14a,b). Zero-mean white noise of variance  $\sigma_n^2$  is added to the sine wave to yield either 40 or 20 dB SNR, defined as  $A^2 / 2\sigma_n^2$ . The time window used in WRLS is 10 sample points and the forgetting factor for RLS is 0.5, chosen so that both have the same transient time (Figure 2.15). The experimental results in Figure 2.16 suggest that WRLS has better noise immunity, for lower SNR conditions, than RLS.

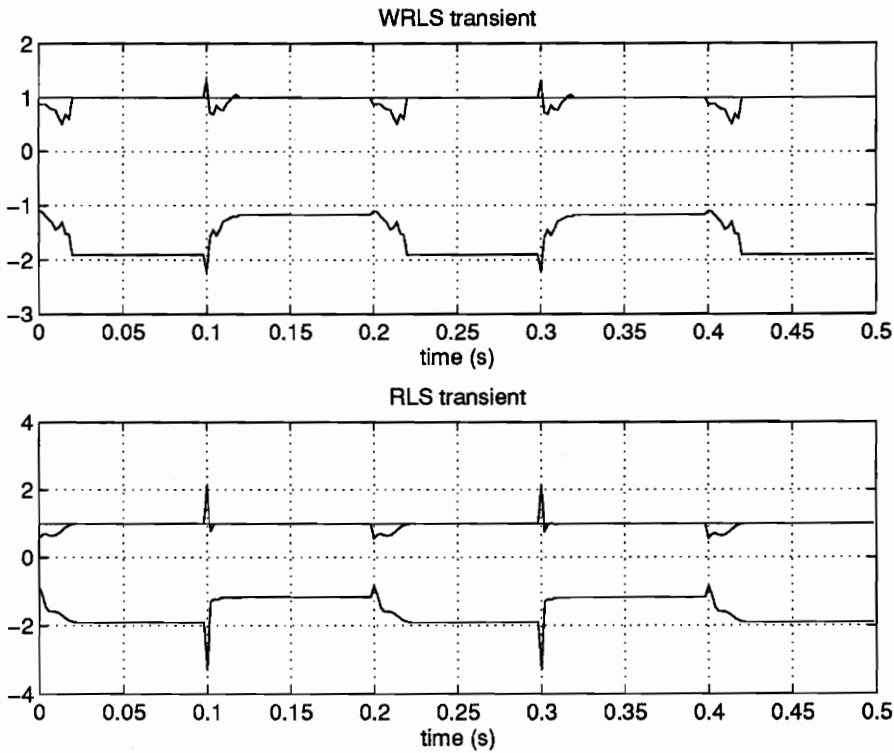
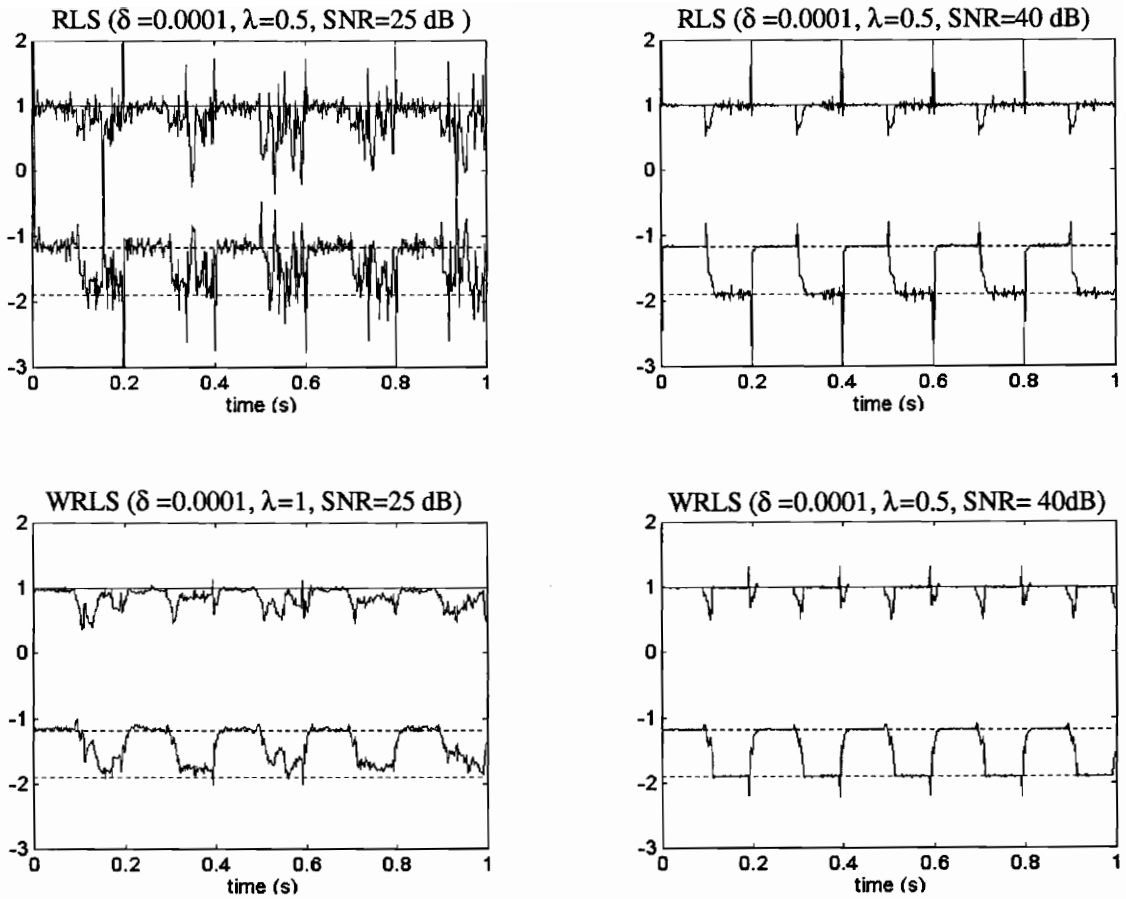


Figure 2.15a,b Transients of WRLS with  $N_w = 10$ ,  $\lambda = 1$  (top), and RLS with  $\lambda = 0.5$  (bottom).



**Figure 2.16a-d RLS (top) and WRLS (bottom) estimates (solid lines) and true values of  $a_1(t)$ ,  $a_2(t)$  (dotted lines).**

## 2.4 Magnitude estimation with auxiliary minimum variance spectral estimator

Although the Parametric Spectral Estimator (PSE) is known to yield high frequency resolution, its signal component amplitude estimation performance is relatively poor. Usually the estimates exhibit large variance. Since the PSE is based on the all-pole model for the signal, the amplitude of the signal component is inversely proportional to the distance from the poles of the AR model to the unit circle. The error in the estimated radius of each pole will be significantly magnified when

the poles are close to the unit circle. The closer the pole is to the unit circle, the larger the magnification is. When the  $i$ -th signal frequency component is close to the unit circle, the amplitude of this frequency component is mainly determined by the dominant pole at this frequency, i.e.

$$\hat{A}_i \propto \frac{1}{1 - \hat{r}_i} \quad (2.35)$$

where  $\hat{r}_i$  is the estimated radius of the dominant pole. The sensitivity of the amplitude estimation of the PSE to an error in  $\hat{r}_i$  can be approximated as

$$S_i = \frac{\partial A_i}{\partial r_i} \propto (1 - r_i)^{-2} \quad (2.36)$$

Assuming that  $\hat{r}_i = 0.9$ , we find that  $S_i \approx 100$ ; i.e. the error in  $\hat{r}_i$  is magnified 100 times in estimating the amplitude of the frequency component. Therefore, when the signal components are sinusoids in background noise, magnitude estimation using PSE will incur large variance.

In many practical cases, the error in the estimated amplitude of the frequency components is not critical, such as in the situation where the main concern is to detect or track the signal components. In some applications, the magnitude or the power of the signal components plays an important role. To improve the amplitude estimation in PSE, an auxiliary amplitude estimator can be used in conjunction with the parametric spectral estimator; an example is the Minimum Variance Spectral Estimator (MVSE) [108], which has been successfully used to estimate the signal power from narrow band filters [109].

The MVSE can be introduced from a simple signal model in which the observed signal is a complex sinusoid in background noise [66], i.e.

$$x(n) = A \exp(j2\pi f_0 n \Delta t + j\varphi) + N(n) \quad n = 0, 1, \dots, M-1 \quad (2.37)$$

where  $A$  is the amplitude of the complex sinusoid,  $\varphi$  is the initial phase, and  $N(n)$  is the background noise with zero mean and covariance matrix  $R_{NN}$ . Among the parameters in (2.37), the frequency for the sinusoid is a known parameter, while the amplitude  $A$  and the initial phase  $\varphi$  of the sinusoid are unknown parameters. To simplify the notation, we define the complex amplitude  $A_c$  as

$$A_c = A \exp(j\phi) \quad (2.38a)$$

We also define the observation vector  $\mathbf{x}$  and the normalized sinusoid vector  $\mathbf{s}$  as

$$\mathbf{x} = [x(0), x(1), \dots, x(M-1)]^T \quad (2.38b)$$

$$\mathbf{s} = [1, \exp[2\pi f_0 \Delta t], \exp[2\pi f_0 2\Delta t], \dots, \exp[2\pi f_0 (M-1)\Delta t]]^T \quad (2.38c)$$

It has been proved that the Maximum Likelihood Estimate (MLE) of the complex amplitude has the following form [66]:

$$\hat{A}_c = \frac{\mathbf{s}^H R_{NN} \mathbf{x}}{\mathbf{s}^H R_{NN} \mathbf{s}} \quad (2.39)$$

This MLE has also been proved to be the Linear Minimum Variance Unbiased (LMVU) estimate of  $A_c$ . When the background noise in (2.37) is a white Gaussian process of zero mean and variance  $\sigma_N^2$ , the covariance matrix of the background noise  $R_{NN}$  can be simplified as:

$$R_{NN} = \sigma_N^2 \mathbf{I} \quad (2.40)$$

Then (2.39) can be represented as:

$$\hat{A}_c = \frac{\mathbf{s}^H \mathbf{x}}{\mathbf{s}^H \mathbf{s}} = \frac{\sum_{n=0}^{M-1} s^*(n)x(n)}{\sum_{n=0}^{M-1} s^*(n)s(n)} \quad (2.41a)$$

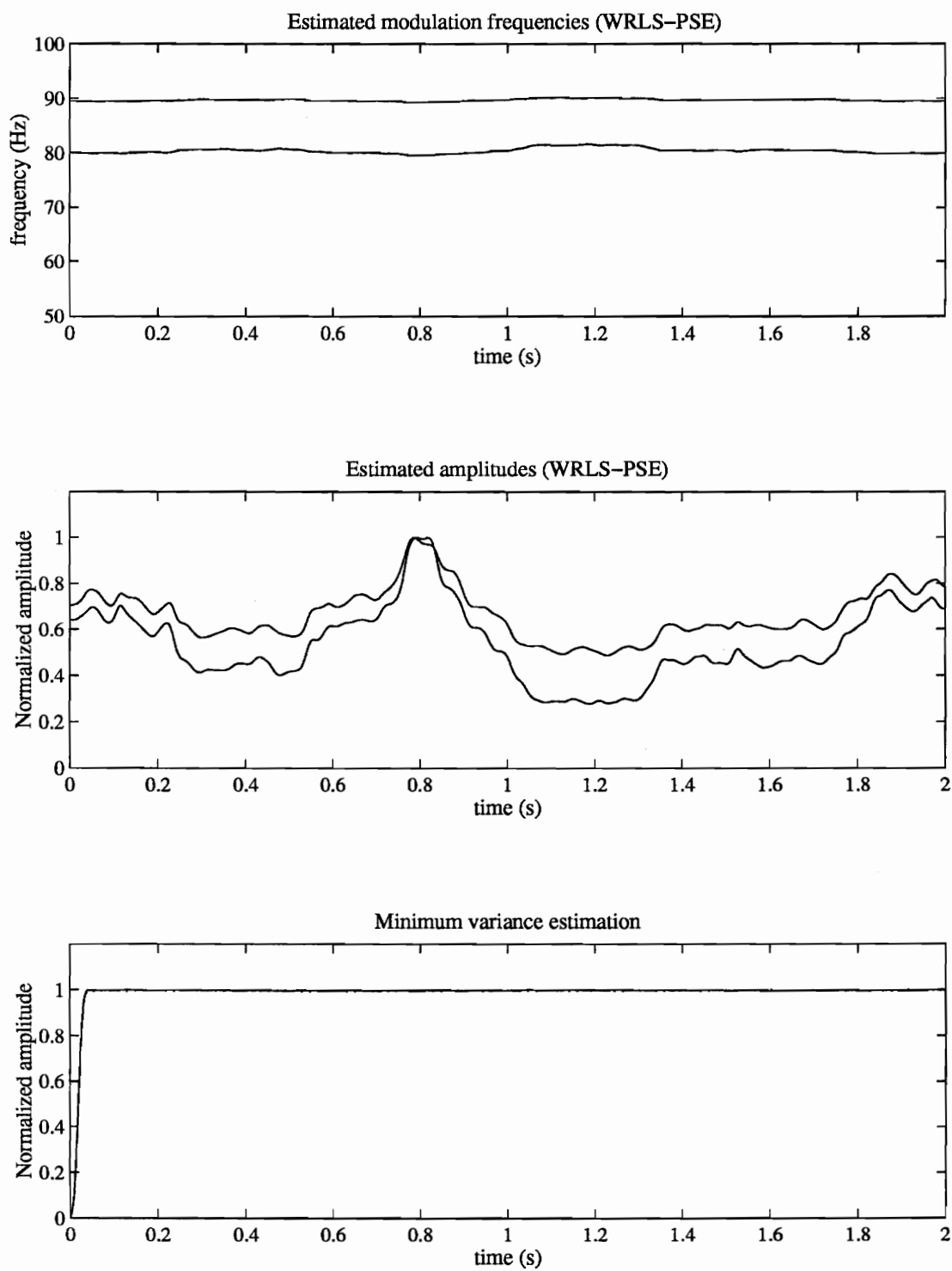
An important factor in (2.41a) is that the estimate of  $A_c$  does not require additional information about the background noise other than the white noise and zero mean assumptions. Under the white noise model, the MLE in (2.41a) is also equivalent to the Least Squares estimate. Note that the  $s(n)$  are samples of a normalized complex sinusoid and that  $s^*(n)s(n) \equiv 1$ , so that (2.41a) can be simplified as

$$\hat{A}_c = \frac{1}{M} \sum_{n=0}^{M-1} e^{-j2\pi f_0 n \Delta t} x(n) \quad (2.41b)$$

Examining (2.41b), one will recognize the equation for the Discrete Fourier transform of the data samples at frequency  $f_0$ . That is, knowing the exact frequency of the signal, the Discrete Fourier transform yields the 'best' estimate of the signal amplitude under the signal model that the observation consists of a sinusoid in a white noise background. Equation (2.41b) can also be interpreted as a linear FIR filtering operation with  $s^*(n)/M$  as filter coefficients. The frequency response of  $s(n)$  is a narrow bandpass filter. The bandwidth is controlled by the data window width  $M$ . Different window functions can be used to help reduce the sidelobe effects. Equation (2.41b) can be interpreted as a similarity measure as well. It measures the similarity between the observation and the basis function, the normalized complex sinusoid at the signal frequency.

The MVSE can be used in conjunction with the PSE. The operation consists of two steps. First the PSE estimates the frequencies of the signal components and then MVSE is used to determine the complex amplitudes of the signal components. The second step helps to reduce the variance in estimated signal component amplitudes. To demonstrate this idea, a test signal was designed, composed of 2 sinusoids with frequencies of 80 Hz and 90 Hz respectively. Both sinusoids are unit amplitude. White noise is added to yield a 20 dB signal-to-noise ratio. The frequencies of the test signal were estimated with the RLS-PSE algorithm. The amplitudes of the signal components were estimated by applying (2.41b) at the estimated frequencies with a Hamming window of 100 samples. Figure 2.17 provides the experimental results. The WRLS algorithm produces high resolution with a small variation in frequency (Figure 2.17a), but its magnitude estimates show relatively large variance (Figure 2.17b). The amplitudes estimated by the auxiliary MVSE are all equal to unity, without visible variations (Figure 2.17c). This example indicates that the auxiliary magnitude estimator can help to improve the performance of the PSE.





**Figure 2.17 a-c Performance of PSE with MVSE as auxiliary magnitude estimator for a time-invariant signal.**

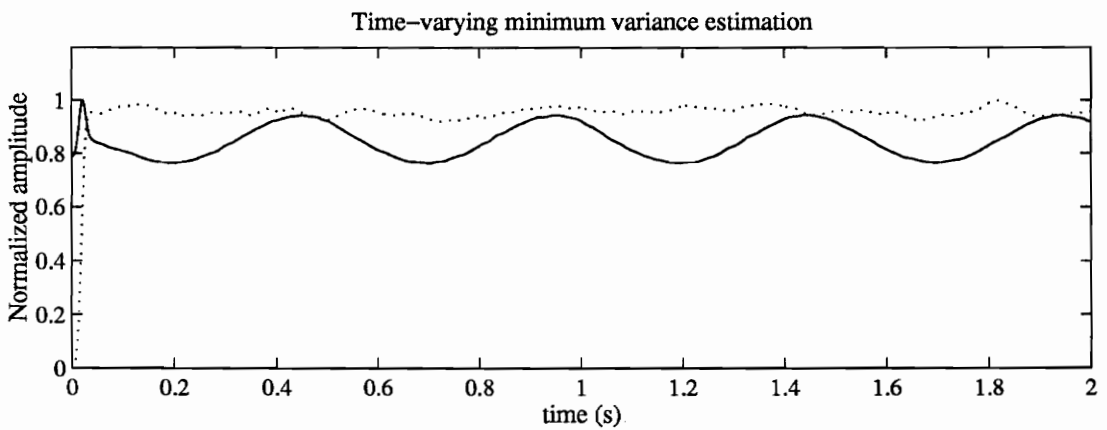
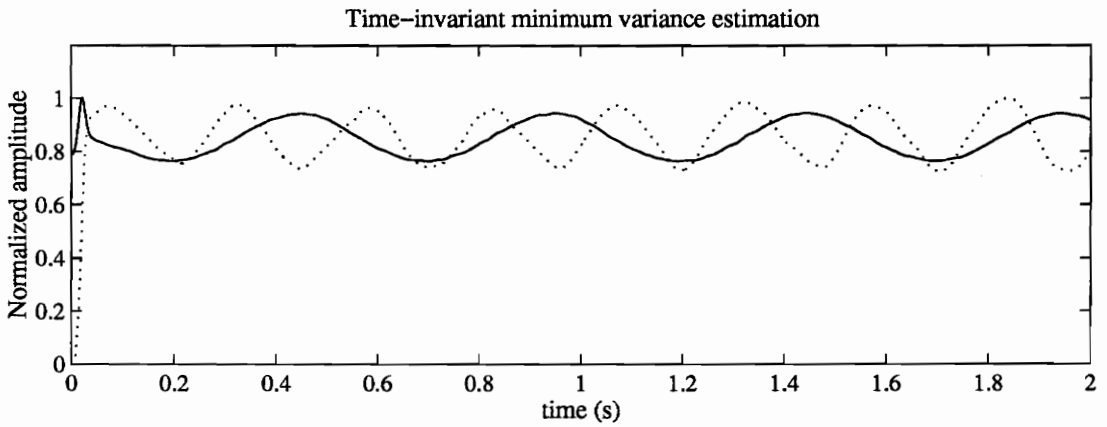
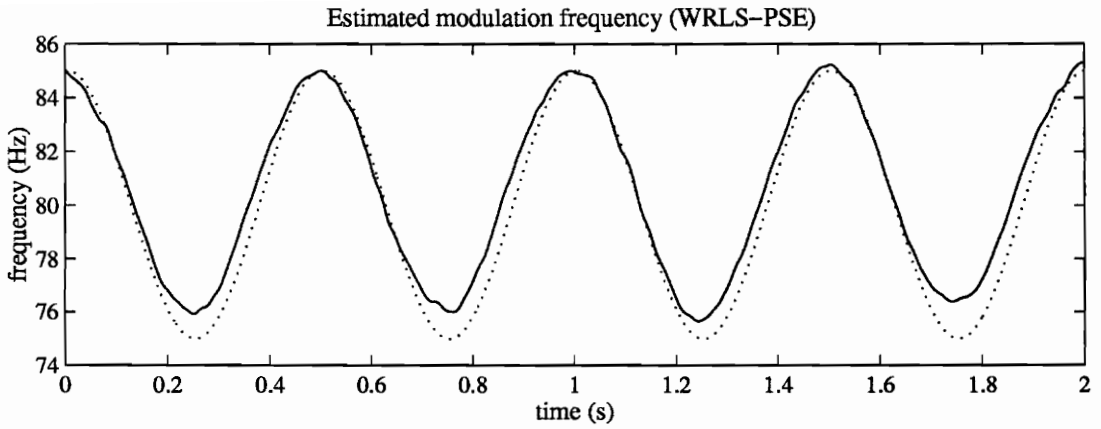
However, the MVSE in (2.39) or (2.41b) is derived based on the assumption that the signal and the background noise are stationary or time-invariant. When the signal is time-varying, the estimator in (2.41b) can bring about serious distortion. Good amplitude estimation by MVSE requires a relatively large time window (large  $M$ ) to produce a narrow bandpass filter and to reduce the sidelobe energy leakage. Since the MVSE is time-invariant, it can not follow the changes in the signal frequency. When the signal component moves in and out of the passband of the MVSE filter, the estimated magnitude will be modulated. Figure 2.18 provides a time-varying example. The signal used is a Frequency Modulated (FM) signal which is modulated by a 2 Hz sinusoid on a unit amplitude 80 Hz carrier with a modulation depth of  $\pm 5$  Hz (dotted line in Figure 2.18a). After addition of white noise, the SNR of the FM signal is 20dB. The time varying frequency of the signal is tracked by the WRLS algorithm. The estimated frequency (solid line in Figure 2.18a) shows a close match with the original, while the estimated amplitude from WRLS reveals a large variation (solid line in Figure 2.18b). Using (2.41b) the time-invariant MVSE with a 40 sample Hamming window can not follow the changes in signal frequency, so that its amplitude estimate is modulated by the frequency changes (dotted line in Figure 2.18b). The variations in both estimates in Figure 2.18b are of the same order of magnitude. This example indicates that the conventional time-invariant MVSE can not perform well in a nonstationary environment.

The reason that the time-invariant MVSE can not work well in a nonstationary environment is that the frequency of the basis function,  $f_0$  in  $s(n) = e^{j2\pi f_0 n \Delta t}$ , is time-invariant. Since (2.41b) provides a similarity measure between the observation and the basis function, the similarity between a fixed frequency basis function and the time-varying signal varies depending on the time-averaged result. Therefore a statistically consistent decomposition requires that the frequency of the basis function has the same time-varying pattern as the signal component, i.e.

$$s(t) = \exp(j\theta(t)) \tag{2.42}$$

where the time-varying phase of the normalized complex sinusoid is defined by

$$\theta(t) = \int_0^t 2\pi f_0(\tau) d\tau \tag{2.43a}$$



**Figure 2.18 a-c Performance comparison of time-invariant and time-varying MVSE for estimating the amplitude of an FM signal.**

Or the instantaneous frequency of the basis function is defined by

$$\frac{1}{2\pi} \frac{d\theta(t)}{dt} = f_0(t) \quad (2.43b)$$

where  $f_0(t)$  is the time-varying frequency of the signal component, which can be estimated by using PSE or, in our design, by using MPSE.

The continuous form of (2.42) and (2.43) can be easily modified into discrete form by substituting for the time  $s(t)$  with  $s(n\Delta t)$  (simply written as  $s(n)$ ) and replacing the integration with summation. That is,

$$\hat{s}(n) = \exp(j\hat{\theta}(n)) \quad (2.44a)$$

$$\hat{\theta}(n) = \sum_{k=0}^n 2\pi \frac{\hat{f}_0(k)}{f_s} \quad (2.44b)$$

where  $f_s$  is the sampling frequency. The time-invariant MVSE in (2.41b) can thus be modified to a time-varying MVSE in the following form (assuming  $M$  is even):

$$\hat{A}_c(n) = \frac{1}{M} \sum_{k=0}^{M-1} e^{-j\hat{\theta}(n)} x(n - M/2) \quad (2.45)$$

Equation (2.45) is still a representation of an FIR filtering operation, but with the time-varying coefficients  $s^*(n) = e^{-j\hat{\theta}(n)}$ . The passband of the time-varying MVSE follows the changes in the signal component to be decomposed. To demonstrate the performance of the time-varying MVSE in conjunction with WRLS, we applied (2.45) to the signal in Figure 2.18a. The frequency of the signal used is the estimate from WRLS shown as the dotted line in Figure 2.18a. The amplitude estimate from the time-varying MVSE is provided as the dotted line in Figure 2.18c, while the solid line is the original amplitude estimate from the time-invariant MVSE and WRLS. Comparing the time-varying and time-invariant amplitude estimates (dotted lines in Figure 2-18b,c, the time-varying MVSE yields a better estimate than its time-invariant version, in spite of the estimation error in  $\hat{f}_0(t)$  used for constructing the basis function  $s^*(n) = e^{-j\hat{\theta}(n)}$ .

## 2.5 Some MPSE realization issues

A peculiar issue in the MPSE implementation is the selection of the time window width for the AR-PSE. There are two major considerations here. The first consideration is the trade-off between time resolution and frequency resolution. A long time window offers good frequency localization and frequency resolution, but results in poor time localization and time resolution, and vice versa. The other consideration is the trade-off between the statistical consistency of the estimates and the stationarity of the signal. A more consistent estimate requires more data, especially in a noisy environment. However, a long window will smooth out all the time-varying information within the window. One of the advantages of MPSE is that the time windows in different frequency bands can be selected independently. That is, a long time window can be applied to a noisy frequency band, while a short window can be used in a subband where the signal is strongly nonstationary. Therefore, the designer has to balance resolution and statistical consistency according to the resolution requirements and signal conditions of the application at hand. This issue will be discussed further in the next section.

The AR-PSE estimates are sensitive to the initial phase of a signal consisting of sinusoids [110, 66]. In computer simulations to detect the time-varying spectrum of an FM signal, it was observed that the modulation frequency estimates exhibited large variations correlated with the position of the sliding window. That is, the estimated frequency had a sinusoid-like bias as a function of the initial phase of the carrier frequency. When the time window slides along the data record, the bias varies with the change of initial phase. These variations can be as much as 16% of  $1/(N_w T_s)$  [111]. However, the modified covariance AR-PSE method is known to be less sensitive than most other AR-PSE algorithms [112]. The spectrum of the estimated modulation frequency variations in our computer experiment shows that its power is concentrated around the carrier frequency, which is usually much higher than the modulation frequency. Thus, the variations can be removed by a post-smoothing filter, such as an FIR filter or a simple boxcar moving average filter. In the next section, a 40 point FIR lowpass filter with cutoff at  $f = 0.2$  cycles/sample is used in all the resolution studies of MPSE. In the later computer simulations, all the MPSE results involve boxcar moving average smoothing filters of half the lengths of the sliding windows.

### 3. MPSE RESOLUTION ANALYSIS

One of the important characteristics of a signal processor is its resolution in time-frequency space, which measures how close two signal components can be located in the phase space so that the processor can still produce the time and the frequency of each individual component. Basically there are three resolution measures, i.e. the *time (space) resolution*, the *frequency resolution*, and the product of the time and frequency resolution called the *resolution cell size*.

Frequency resolution is known to be inversely proportional to the length of the time window used, i.e. a short time window results in poor frequency resolution and vice versa. The product of the time and frequency resolution, the resolution cell size, of a signal processor is bounded below by  $1/2$ , known as the Heisenberg uncertainty principle, "which is at the root of the fundamental principles of communication [35]."

In this chapter, the resolutions of the MPSE are analyzed in detail. The results are compared with the fixed resolution AutoRegressive Parametric Spectral Estimators (AR-PSE) and with the Fourier transform based methods. Both the analytical and experimental results show that, in many situations, the MPSE resolves at higher resolution than the conventional fixed resolution AR-PSE and the Fourier transform based methods. MPSE can even break the resolution lower bound of the Heisenberg uncertainty principle. The explanations are provided in Section 3.5, in terms of trading off between the different resolutions and the proper use of *a priori* knowledge.

#### 3.1 Computation of the resolutions in time-frequency space

The computation of time and frequency resolution requires integration [35]. To numerically compute these resolutions, we replace integrations with summations in the original definitions. The time resolution is defined as the effective (r.m.s.) duration or window width of an operator in the time domain. This is a measure of how close two signal components can be spaced in time, and still be correctly separated in time by the operator. For an operator with a time representation  $h(n)$ , such as the unit pulse response of a filter or the window function of a spectral estimator, the normalized time resolution  $\Delta n$  is then defined by:

$$\Delta n = \sqrt{2\pi \frac{\sum_{n=0}^{N-1} (n - \bar{n})^2 h^2(n)}{\sum_{n=0}^{N-1} h^2(n)}} \quad (\text{in samples}) \quad (3.1a)$$

where  $\bar{n}$  is the normalized center of  $h(n)$ . If the unit pulse response  $h(n)$  has a finite length  $N$ , the normalized center  $\bar{n}$  is defined as

$$\bar{n} = \frac{\sum_{n=0}^{N-1} nh^2(n)}{\sum_{n=0}^{N-1} h^2(n)} \quad (\text{in samples}) \quad (3.1b)$$

For non-finite length  $h(n)$ ,  $N$  is selected as the smallest time  $n$  such that  $|h(n)| < \delta \approx 0$  for all  $n \geq N$ . When the sample period is  $T_s$ , the center and the time resolution of  $h(n)$  can be written as follows:

$$\bar{t} = \bar{n} T_s \quad (\text{in seconds}) \quad (3.1c)$$

$$\Delta t = \Delta n T_s \quad (\text{in seconds}) \quad (3.1d)$$

The frequency resolution proposed by Gabor [35] is the effective (r.m.s.) bandwidth of an operator in the frequency domain. In our study, we propose the frequency resolution to be the minimum frequency separation, of two equi-amplitude sinusoids embedded in white noise, for which the two sinusoids can be discerned. The difference between the latter definition and Gabor's definition for a windowed Fourier transform is that Gabor's frequency resolution is computed over the entire frequency range while for our definition the computation is over the mainlobe of the spectral window function only.

Let  $H(f_n)$  be the Fourier transform of  $h(n)$ . The normalized frequency resolution  $\overline{\Delta f_n}$  is defined by

$$\overline{\Delta f_n} = \sqrt{2\pi \frac{\sum_{n=n_1}^{n_2} (f_n - \overline{f_n})^2 |H(f_n)|^2}{\sum_{n=n_1}^{n_2} |H(f_n)|^2}} \quad (\text{in cycles/sample}) \quad (3.2a)$$

where  $n_1$  and  $n_2$  are the limits of summation, and  $\overline{f_n}$  is the normalized center frequency of  $H(f_n)$  defined by

$$\overline{f_n} = \frac{\sum_{n=n_1}^{n_2} f_n |H(f_n)|^2}{\sum_{n=n_1}^{n_2} |H(f_n)|^2} \quad (\text{in cycles/sample}) \quad (3.2b)$$

When the sampling frequency is  $f_s = 1/T_s$ , the frequency resolution  $\Delta f$  and its center  $f_n$  can be written as follows:

$$\Delta f = \overline{\Delta f} f_s \quad (\text{in Hz}) \quad (3.2c)$$

$$f_n = \overline{f_n} f_s \quad (\text{in Hz}) \quad (3.2d)$$

In computing the frequency resolution, zero padding will smooth the spectrum  $H(f_n)$ , and reduce the numerical error caused by replacing integration with summation.

The resolution cell size in time-frequency space,  $\Delta c$ , is defined as the product of the time resolution  $\Delta t$  and the frequency resolution  $\Delta f$ , i.e.

$$\begin{aligned} \Delta c &= \Delta t \Delta f \\ &= \Delta n \overline{\Delta f_n} \end{aligned} \quad (3.3)$$

Note that the resolution cell size  $\Delta c$  is not a function of the sampling frequency.



### 3.2 Frequency resolution

In MPSE the frequency resolution is equal to the frequency resolution of the spectral estimator, which for an AR spectral estimator is approximately [65]

$$\overline{\Delta f_{AR}} = \frac{1.03}{p[\eta(p+1)]^{0.31}} \quad (3.4)$$

for  $p\eta > 10$ , where  $p$  is the order of the AR estimator, and  $\eta$  is the SNR of one sinusoid.

For Windowed Fourier Transform (WFT) based methods, the frequency resolution is the effective frequency bandwidth computed over the mainlobe of  $H(f_n)$ , instead of being computed over  $[-0.5f_s, 0.5f_s]$  as for Gabor's definition. Table 3.1 gives some examples of the frequency resolution of the WFT, in which  $\Delta f_{GB}$  denotes Gabor's frequency resolution, and  $\Delta f_{MLB}$  denotes the mainlobe-based frequency resolution. Among all the given resolutions, the rectangular window has the best  $\Delta f_{MLB}$ , while  $\Delta f_{MLB}$  for the Gaussian window is the worst. Comparing  $\Delta f_{GB}$  with  $\Delta f_{MLB}$ , we note that they are approximately equal, unless the spectral window sidelobes are very large, in which case  $\Delta f_{GB} \gg \Delta f_{MLB}$ . It is also shown in Table 3.1 that  $\Delta f_{MLB}$  is inversely proportional to the window width  $N_w$ , a relationship that holds for different  $N_w$ .

**Table 3.1 The time and frequency resolutions of windowed Fourier transform.**

window	$\Delta f_{GB}$ in Hz	$\Delta f_{MLB}$ in Hz	$\Delta f_{MLB}(N_w)$	$\Delta n_{WFT}(N_w)$	$\Delta c_{GB}$	$\Delta c_{MLB}$
<i>Gaussian</i>	0.0245	0.0245	$2.45/N_w$	$0.2045 N_w$	0.5003	0.5003
<i>Rectangular</i>	0.0665	0.0084	$0.84/N_w$	$0.7239 N_w$	2.1481	0.6079
<i>Bartlett</i>	0.0139	0.0129	$1.29/N_w$	$0.3925 N_w$	0.5425	0.5093
<i>Hamming</i>	0.0156	0.0133	$1.33/N_w$	$0.3799 N_w$	0.5170	0.5038
<i>Hanning</i>	0.0143	0.0143	$1.43/N_w$	$0.3583 N_w$	0.5136	0.5110
<i>conditions</i>	$N_w=100$ samples, padded with 1900 zeros, $T_s=1$ s.		tested for different $N_w$			

### 3.3 Time resolution

As the AR-PSE is based on the assumption of stationarity, the properties derived on that basis are the properties of the steady state response of the AR-PSE. In using MPSE, the stationarity assumption presumably no longer holds. It becomes imperative to know the temporal characteristics of the AR-PSE in order to formulate its time resolution.

To study the temporal characteristics of the AR-PSE, a virtual transfer function is introduced, as belonging to the system with the true time-varying spectrum of a signal as its input and the corresponding estimate as its output (Figure 3.1). The temporal characteristics are analyzed by studying both the impulse response and the frequency response of this virtual transfer function.

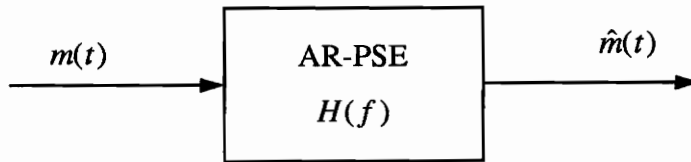


Figure 3.1 Virtual transformation function.

A nonstationary test signal  $x(t)$  is designed to derive a model of the virtual transfer function in the frequency domain. The signal  $x(t)$  is an FM signal with a carrier frequency of 160 Hz. The modulation signal  $m(t)$  is a single frequency sinusoid, i.e.

$$m(t) = 8 \sin(2\pi f_m t) \quad (3.5a)$$

The FM signal  $x(t)$ , which can be approximated as a second order time-varying AR process, is

$$\begin{aligned} x(t) &= \sin\left(2\pi \int_0^t 160 + m(\tau) d\tau\right) \\ &= \sin\left[2\pi 160t - \frac{8}{f_m} \cos(2\pi f_m t)\right] \end{aligned} \quad (3.5b)$$

The time-varying spectrum of  $x(t)$  has a peak located at  $160+m(t)$  Hz, which is also the instantaneous frequency  $f_i(t)$ . Using AR-PSE, the time-varying part of the spectrum is reflected in the AR parameter estimates

$$\begin{aligned}\hat{A}(t) &= \{1, \hat{a}_1(t), \hat{a}_2(t)\} \\ &= \{1, -2\hat{r}(t) \cos[2\pi\hat{f}_i(t) / f_s], \hat{r}^2(t)\}\end{aligned}\quad (3.6a)$$

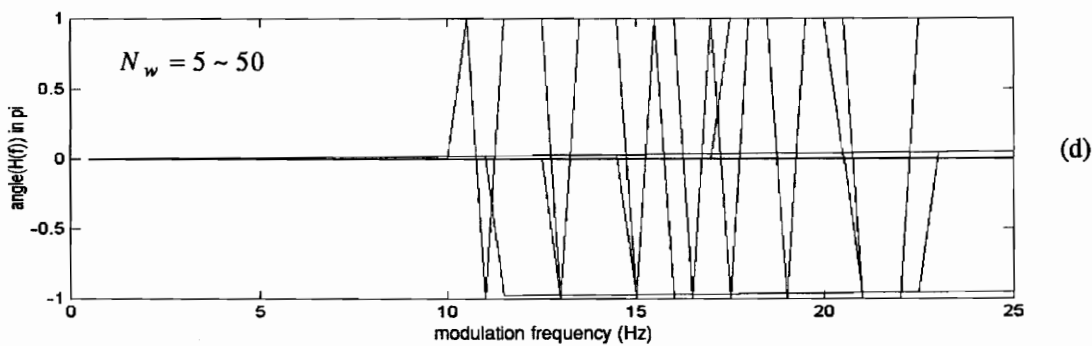
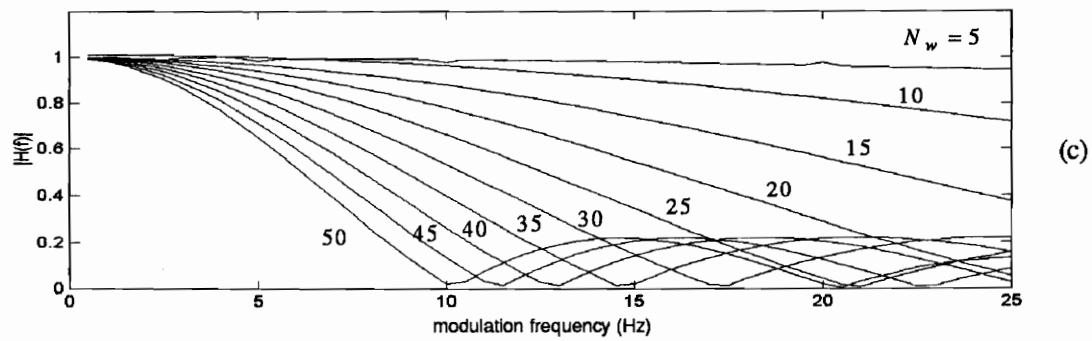
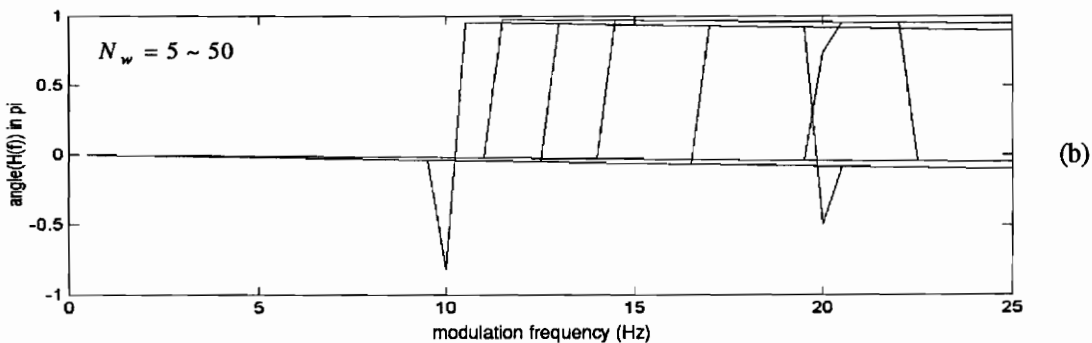
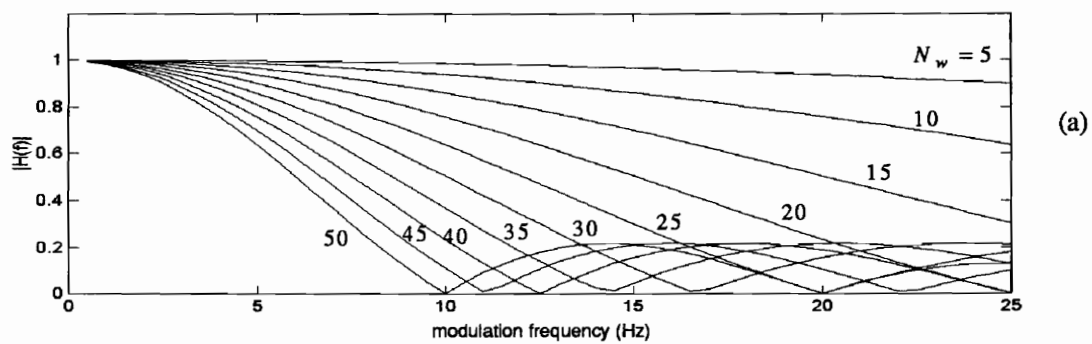
where  $|\hat{r}(t)| \leq 1$  and  $|\hat{f}_i(t)| \approx 1$ . These, in turn, lead to  $\hat{m}(t)$ , the following estimate of  $m(t)$ ,

$$\hat{m}(t) = \frac{f_s}{2\pi} \arccos\left[\frac{-\hat{a}_1(t)}{2\sqrt{\hat{a}_2(t)}}\right] - 160 \quad (\text{in Hz}) \quad (3.6b)$$

By scanning the modulation frequency  $f_m$  from low to high, the virtual transfer function in the frequency domain can be constructed experimentally from

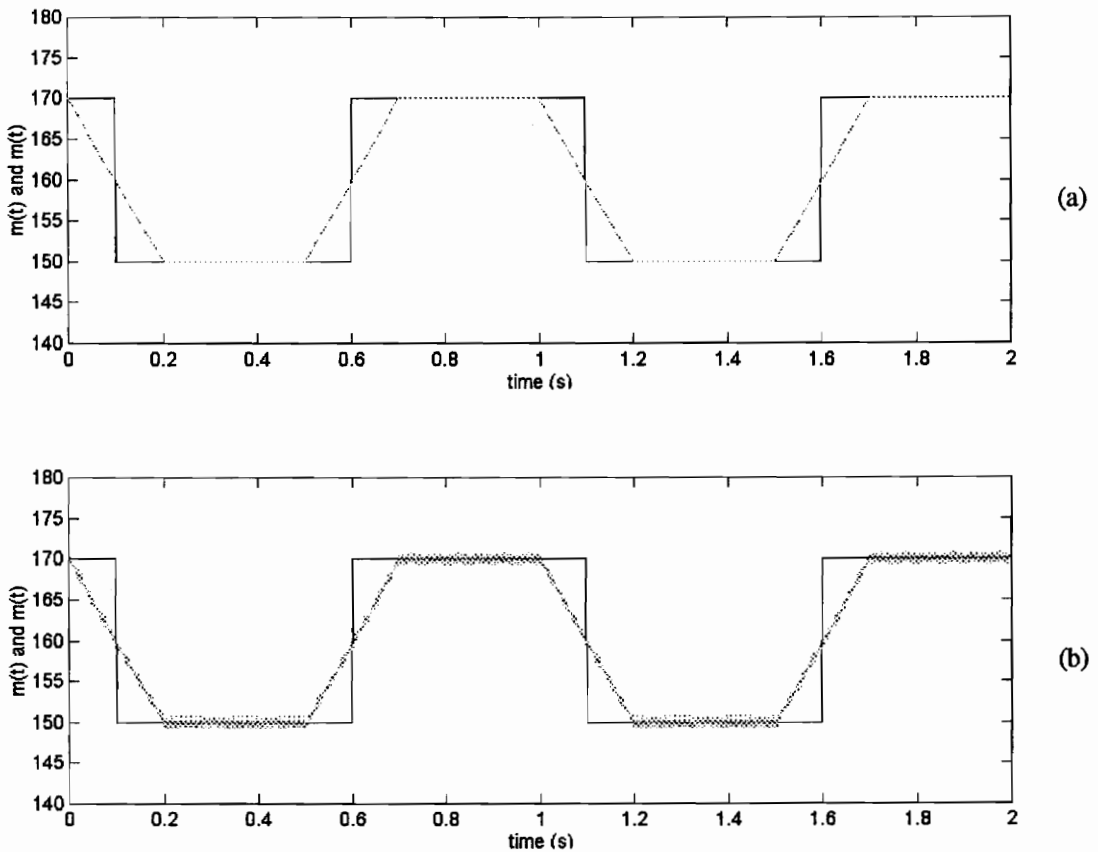
$$H_{AR-PSE}(f_m) = \frac{\hat{M}(f_m)}{M(f_m)} \quad (3.7)$$

where  $M(f_m)$  is the Fourier transform of  $m(t)$  at  $f_m$ , and  $\hat{M}(f_m)$  is the Fourier transform of  $\hat{m}(t)$  at  $f_m$ . In our study,  $f_m$  scans from 0.5 Hz to 25 Hz with a step size of 0.5 Hz. Figures 3.2a and 3.2b show the magnitude and phase responses of  $H_{AR-PSE}(f_m)$  using WRLS for various window widths  $N_w$ . Figures 3.2c and 3.2d show the magnitude and phase responses of  $H_{AR-PSE}(f_m)$  using the Burg algorithm for various window widths  $N_w$ . These figures suggest that  $H_{AR-PSE}(f_m)$  is a SINC function, i.e. its time representation  $h_{AR-PSE}(t)$  acts as a boxcar moving average filter.



**Figure 3.2** Frequency responses of the virtual transformation functions of WRLS (a,b) and Burg (c,d) algorithms.

To explore the transient response of the virtual transfer function, a square wave modulation signal  $m(t)$  is used. The frequency of the testing signal  $x(t)$  is switching instantaneously between 150 Hz and 170 Hz, with a period of 1 second. The sampling rate is 500 Hz. The AR estimators used are the second-order WRLS and Burg algorithms for different window widths. Examples of  $m(t)$  (solid lines) and  $\hat{m}(t)$  (dotted lines) are given in Figure 3.3a-b for WRLS and Burg algorithms respectively, in which the window width  $N_w$  for these AR estimators equals 100 samples. The transient behaviors of both algorithms are boxcar moving average filters. However, the estimates with the Burg algorithm show larger variance than with the WRLS algorithm. The estimation variance is caused by the fact that the Burg algorithm is sensitive to the initial phase of the windowed sinusoidal signal [110, 66].



**Figure 3.3 Temporal characteristics of the Virtual transformation functions of WRLS (3.3a) and Burg algorithms (3.3b).**

The results in Figure 3.2, and Figure 3.3 thus indicate that the AR-PSE acts as a boxcar moving average filter in the time domain, i.e.,

$$h_{AR-PSE}(n) = \begin{cases} \sum_{k=-\frac{W_{boxcar}}{2}}^{\frac{W_{boxcar}}{2}-1} \frac{\delta(n-k)}{W_{boxcar}} & \text{for even } W_{boxcar} \\ \sum_{k=-\frac{W_{boxcar}-1}{2}}^{\frac{W_{boxcar}-1}{2}} \frac{\delta(n-k)}{W_{boxcar}} & \text{for odd } W_{boxcar} \end{cases} \quad (3.8a)$$

and in the frequency domain

$$H_{AR-PSE}(f) = \begin{cases} \text{sinc}(\pi f W_{boxcar} T_s) e^{-j\pi f T_s} & \text{for even } W_{boxcar} \\ \text{sinc}(\pi f W_{boxcar} T_s) & \text{for odd } W_{boxcar} \end{cases} \quad (3.8b)$$

where  $W_{boxcar}$  is the length of the boxcar filter, which relates to the window length  $N_w$  and the AR order  $p$  as follows

$$W_{boxcar} = \begin{cases} N_w - 1 & \text{for the Burg algorithm} \\ N_w - 1 + p & \text{for the WRLS algorithm} \end{cases} \quad (3.8c)$$

Using (3.1a-d) the time resolution  $\Delta n_{AR-PSE}$ , for the AR estimator, can now be represented as follows:

$$\Delta n_{AR-PSE} = 0.7236 W_{boxcar} \quad (3.9a)$$

or, alternatively,

$$\Delta t_{AR-PSE} = 0.7236 W_{boxcar} T_s \quad (3.9b)$$

The time resolution of MPSE is determined by both the band-splitting filters and the time windows used in parametric spectral estimation. At each stage, filtering and spectral estimation are

cascaded operations, so that the total temporal characteristics of MPSE are determined by the convolution of the impulse responses of the frequency band-splitting filter  $h(n)$  and the AR-PSE  $h_{AR-PSE}(n)$ . Figures 3.4 give examples of time resolutions, frequency resolutions and resolution cell size of IIR (Figure 3.4a-c) and FIR (Figure 3.4d-f) band-splitting filters. To simplify the discussion, only the highpass filter will be considered. Figure 3.5 shows the time resolutions of MPSE for different lengths of the FIR band-splitting filters. Note that the time resolution  $\Delta n_{MPSE}$  has little to do with the length of the band-splitting filter, for it is almost exclusively proportional to the window width of AR estimation, i.e.

$$\Delta n_{MPSE} \approx 1.25W_{boxcar} \quad (3.10)$$

For comparison, some time resolutions of the WFT are given in Table 3.1. The time resolution for MPSE is poorer than that for the WFT under the same window width condition. However, the AR-PSE has better frequency resolution and localization than Fourier transform-based methods, especially in a short data window environment. Meanwhile, lowering the order of the AR process, by means of the band-splitting in MPSE, can reduce the required data window width. The loss in time resolution is relatively small compared with the improvement in frequency resolution.

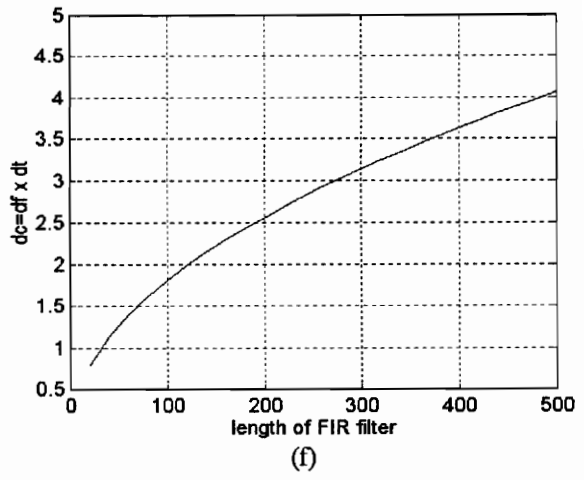
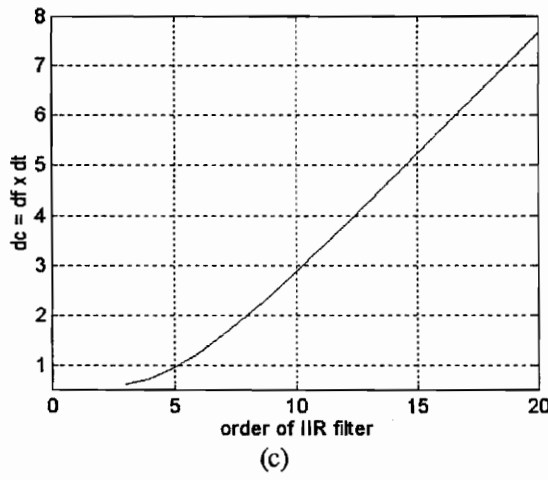
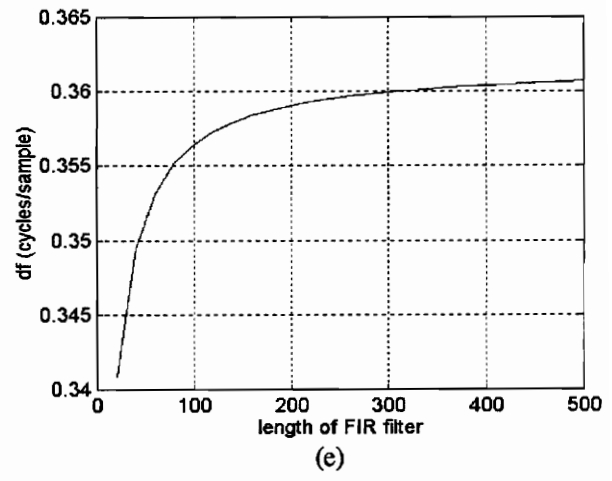
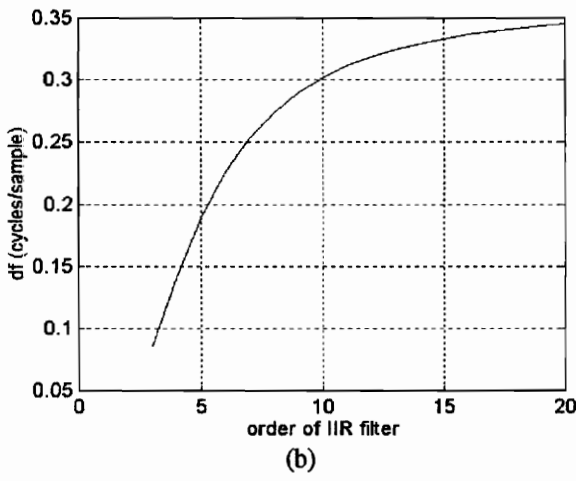
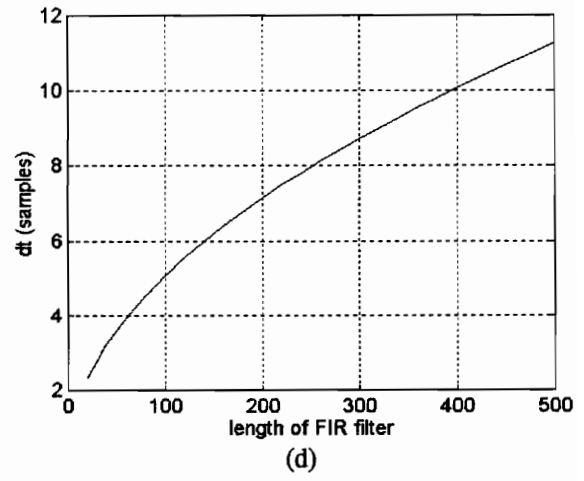
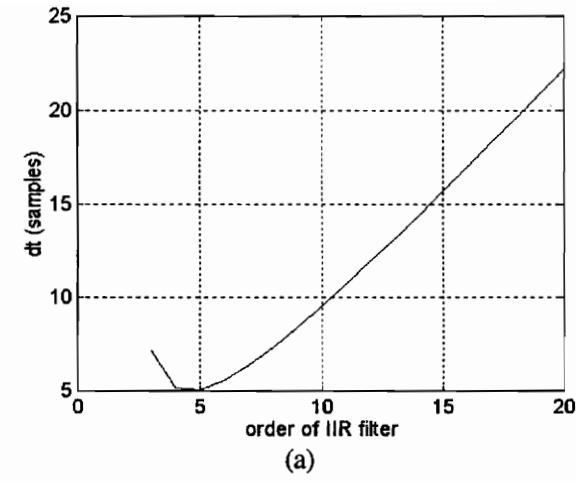


Figure 3.4 Time and frequency resolutions for IIR (a,b) and FIR (c,d) band-splitting filters.



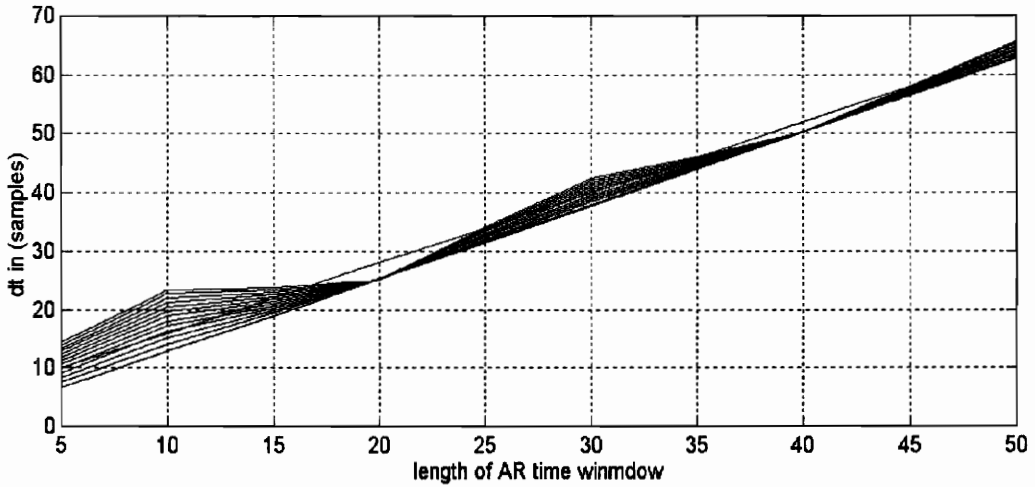


Figure 3.5 Time resolutions of the AR-PSE for FIR band-splitting filters of various lengths.

### 3.4 Resolution cell size in time-frequency space

Using (3.4) and (3.10) in (3.3), the resolution cell size of MPSE can be represented as:

$$\begin{aligned} \Delta c_{MPSE} &= \Delta n_{MPSE} \overline{\Delta f_{MPSE}} \\ &\approx \frac{1.2875 N_w}{p[\eta(1+p)]^{0.31}} \end{aligned} \quad (3.11)$$

For example, if the signal-to-noise ratio  $\eta$  equals 10,000 (40 dB), the window width  $N_w$  used for the AR-PSE equals 20 samples, and the minimum order  $p$  used for the AR-PSE to separate two real frequencies equals 4, then the frequency resolution of MPSE is 0.009 cycles/sample, the time resolution is 25 samples, and the resolution cell size is 0.2249. Note that  $\Delta c_{MPSE}$  is smaller than 0.5, the Heisenberg lower bound.

Table 3.1 gives examples of WFT resolution cell sizes, in which the Gaussian window has the best resolution in time-frequency space. As derived by Gabor [35],  $\Delta c_{Gaussian} = 0.5$ , which is at the Heisenberg lower bound. The other windows all produce resolution cell sizes greater than 0.5. Figure 3.6 provides examples of resolution cell sizes for the IIR and FIR filters.

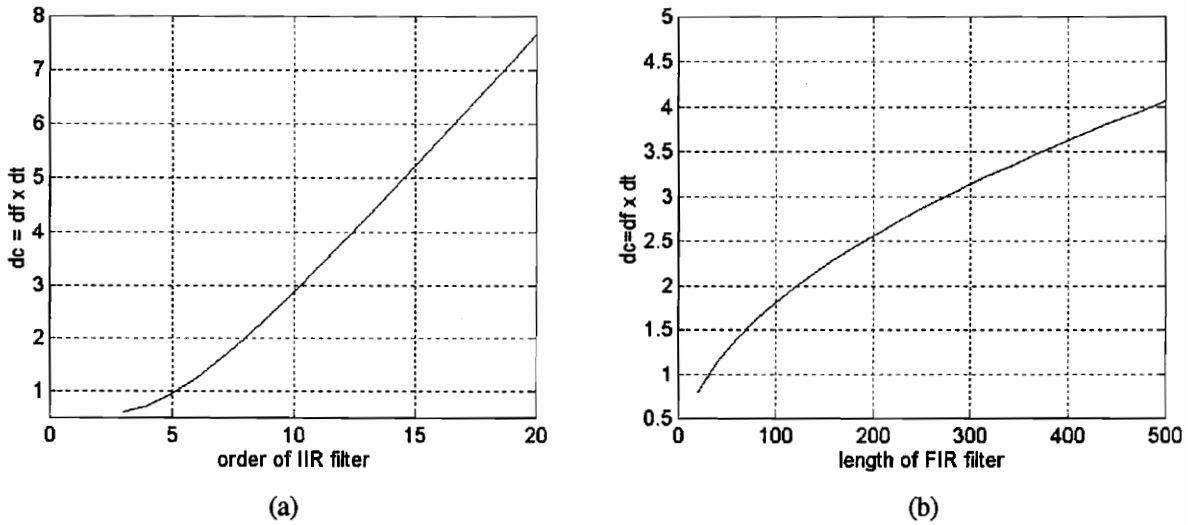


Figure 2.6 Resolution cell sizes for IIR (a) and FIR (b) filters.

### 3.5 Interpretation of MPSE in terms of the wavelet transform

A popular technique in multiresolution digital signal processing is the Discrete Time Wavelet Series (DTWS), or Discrete Wavelet Transform (DWT), a discrete form of the wavelet transform [42, 55]. The DTWS decomposes the signal into multi-octave bands by means of sequential processing blocks, each with the same structure consisting of both a lowpass halfband filter and a highpass halfband filter. The output of each filter is decimated by a factor of 2. The lowpass filters act as scaling operations; their output is sent to the next processing block for further frequency band-splitting and decimation. The highpass filters act as the mother wavelet matching operation; their outputs are the wavelet coefficients of the wavelet at a particular scale.

The advantage of the DTWS is its simplicity. The multiresolution techniques developed in subband coding make it easy to implement. The first disadvantage of the DTWS is that it assumes that the elementary signals in each subband have a simple bandpass-shape spectrum. If the subband signal has a sophisticated spectrum, with very sharp peaks or peaky shapes, that information can not be retrieved correctly by a simple bandpass-shaped mother wavelet. A more

sophisticated mother wavelet is then required to properly decompose the signal [57]. Secondly, each simple wavelet represents a unit of information, which is a windowed tone centered in each of the octave bands. In most applications the signal does not appear naturally as an octave distribution, requiring additional fine scale steps for a proper decomposition into unit information, i.e. a wavelet representation.

The MPSE method is, thus, motivated by the desire to construct the optimal wavelet (sophisticated form) for each frequency band by estimating the subband spectra. Instead of being used as a single DTWS coefficient, the output of each highpass filter is passed to a spectral estimator. The latter gives more detail of the signal in each subband than a single DTWS coefficient. Therefore, MPSE results in higher frequency resolution than the DTWS.

In the DTWS both the time and frequency resolutions are determined mainly by the highpass band-splitting filter (FIR). That is, its frequency resolution  $\Delta f_{DTWS}$  is equal to the effective mainlobe bandwidth of each octave band, and its time resolution equals the effective duration, i.e.

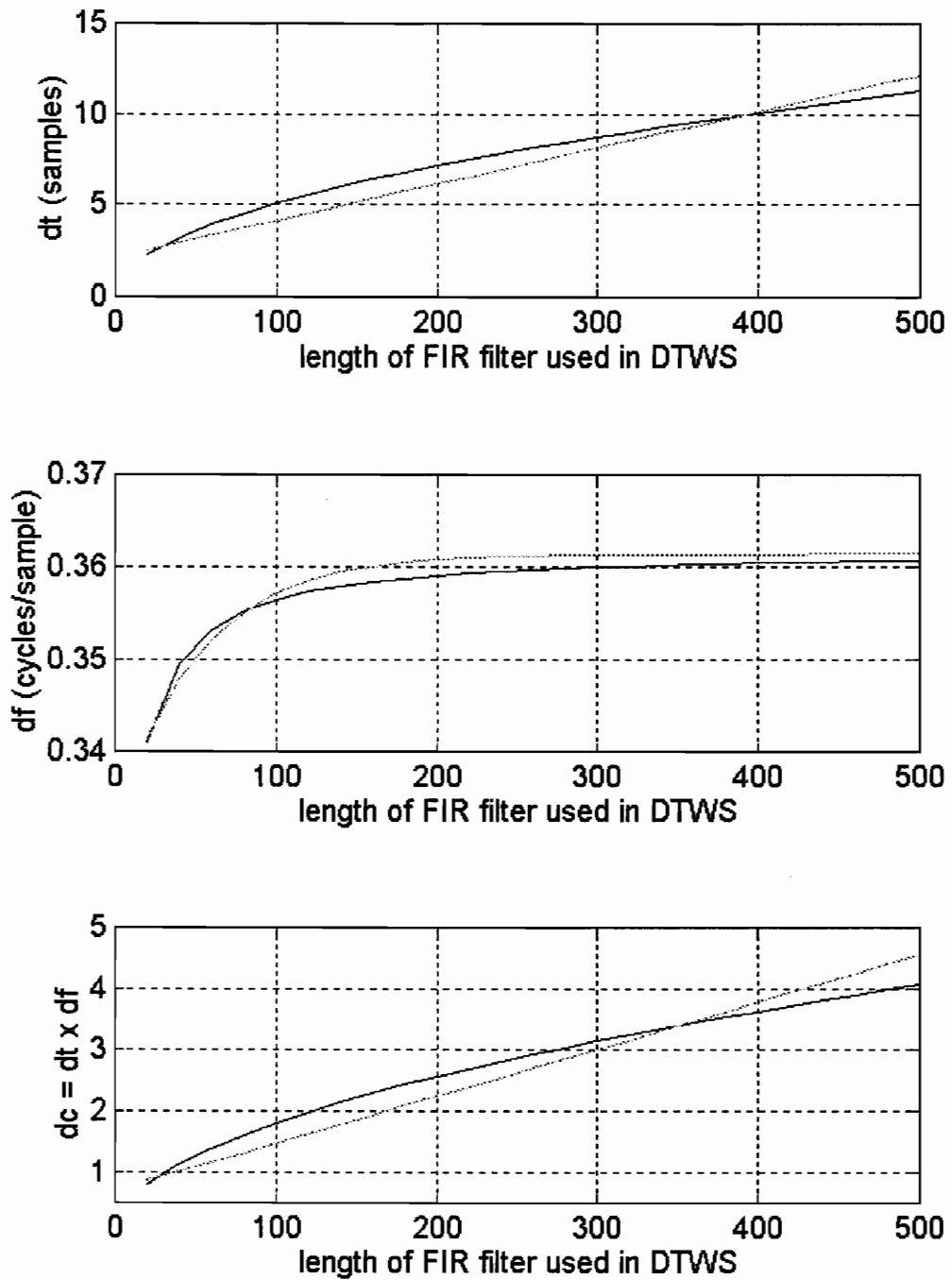
$$\begin{aligned}\Delta f_{DTWS} &= 2^{-(n-1)} f_s \overline{\Delta f}_{MLB} \\ \Delta t_{DTWS} &= 2^{n-1} T_s \Delta n_{WFT}\end{aligned}\tag{3.12}$$

where  $n$  is the octave band index,  $n \geq 1$ , numbered from high to low frequency, and  $\overline{\Delta f}_{MLB}$  and  $\Delta n_{WFT}$  are the normalized resolutions for the highpass filter (See (3.2a) and Table 3.1). Similarly, the time resolution of the DTWS equals the effective length of the unit pulse response of the highpass filter (See (3.1a), and Table 3.1). Figures 3.7a-c show examples of resolutions for DTWS.

The time resolution (Figure 3.7a, solid line) is approximately proportional to the length of the filter. Using a least squares polynomial fitting method, the time resolution can be approximated (dotted line) as:

$$\Delta n_{WFT}(L) \approx 2.1508 + 0.02L \quad (\text{in samples}) \tag{3.13}$$

The frequency resolution (Figure 3.7b, solid line) exponentially approaches the frequency resolution of an ideal filter, which can be fitted to an exponential form (dotted line).



**Figure 3.7** Time resolution (a), frequency resolution (b), and resolution cell size (c) of the DTWS (The solid lines are the actual value and the dashed lines are their polynomial approximations).

$$\begin{aligned}\overline{\Delta f}_{MLB}(L) &= (\overline{\Delta f}_{MLB}(10) - \overline{\Delta f}_{ideal})e^{-0.0194(L-10)} + \overline{\Delta f}_{ideal} \quad (\text{in cycles/sample}) \\ &\approx 0.3614 - 0.02946e^{-0.0194L}\end{aligned}\quad (3.14)$$

where  $L$  is the filter length, for  $10 \leq L \leq 500$ . Note in Figure 3.7b that the frequency resolution becomes insensitive to the filter length  $L$ , and can then be approximated as the constant 0.3614.

The resolution cell size (Figure 3.7c, solid line) of the DTWS can also be approximately fitted to a linear function of  $L$ , i.e.,

$$\Delta c_{MLB}(L) \approx 0.7033 + 0.0077L \quad (3.15)$$

Note that  $L$  is relatively large, so that  $\Delta c_{MLB}(L) \gg 0.5$ , which indicates that the DTWS is not a high resolution operator.

### 3.6 Computer simulations for MPSE

Three computer experiments were executed to explore the performance of MPSE compared with other methods, such as the fixed resolution AR spectral estimator and the wavelet transform. The performance of the WRLS, RLS, and Burg algorithms in the MPSE method have been compared as well. For all experiments, the signal sample interval  $T_s$  is 0.002 s, since  $f_s$  is 500 Hz.

The first experiment is designed to compare the performance of MPSE with either the WRLS realization or the Burg realization, as well as with a conventional fixed resolution AR spectral estimator. The signal for this experiment consists of four frequency components, which can be represented as:

$$y(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t) + \sin[2\pi f_3(t)t] + \sin[\theta_4(t)] \quad (3.16)$$

where  $f_1 = 20$  Hz,  $f_2 = 22$  Hz,  $f_3(t) = 155 + 10 \sum_{k=0}^{\infty} (-1)^k U(t - 0.1k)$ , and

$$\theta_4(t) = 2\pi 50t - \frac{\cos(2\pi 4t)}{4},$$

or, in frequency terms,

$$\begin{aligned} f_4(t) &= \frac{1}{2\pi} \frac{d\theta_4}{dt} \\ &= 50 + \sin(2\pi 4t) \end{aligned}$$

That is,  $f_1$  is 20 Hz,  $f_2$  is 22 Hz,  $f_3$  is switching instantaneously between 155 and 165 Hz with a period of 0.2 second, and  $f_4$  is changing sinusoidally between 49 and 51 Hz with a period of 0.25 seconds (Figure 3.8a). Zero-mean white noise is added to yield SNR equal to 40 dB. The frequency components  $f_1$  and  $f_2$  are used to test the frequency resolution of the estimator;  $f_3$  is for testing the time resolution; and  $f_4$  is for testing both time and frequency resolution in the middle band.

In MPSE, the whole frequency band is divided into subbands 1, 2, 3, and 4, whose frequency regions are: 250 to 125 Hz, 125 to 62.5 Hz, 62.5 to 31.25 Hz, and 31.25 to 0 Hz, respectively. The AR parameter estimators used in the experiment are WRLS and Burg with orders 2, 0, 2, and 4 in the respective subbands. The time windows used are 20, 40, 15, and 40 data points of the subband outputs. To be comparable to the results of WRLS, the Burg algorithm operates in a sliding window approach instead of on a block-by-block basis. The composite MPSE spectrum plotted in time-frequency space shows a multiresolution time-varying spectrum with a logarithmic scale between octave frequency bands, and a linear scale within each octave frequency band. For all images the top 70 dB of the dynamic range is shown. Resolution cell sizes have been computed, and superimposed on the test signal as solid rectangles; for MPSE these are based on local SNR and an order  $p$  of 4. The simulation results in Figures 3.8b and 3.8c show that all four frequency components are correctly estimated by MPSE, and both the WRLS results and the Burg results demonstrate the capability of MPSE to track the frequencies of a time-varying signal. Note that MPSE-Burg shows more phase sensitivity than MPSE-WRLS. In addition, the computational complexity for Burg is much higher than for WRLS, making WRLS more attractive for use in the MPSE method.

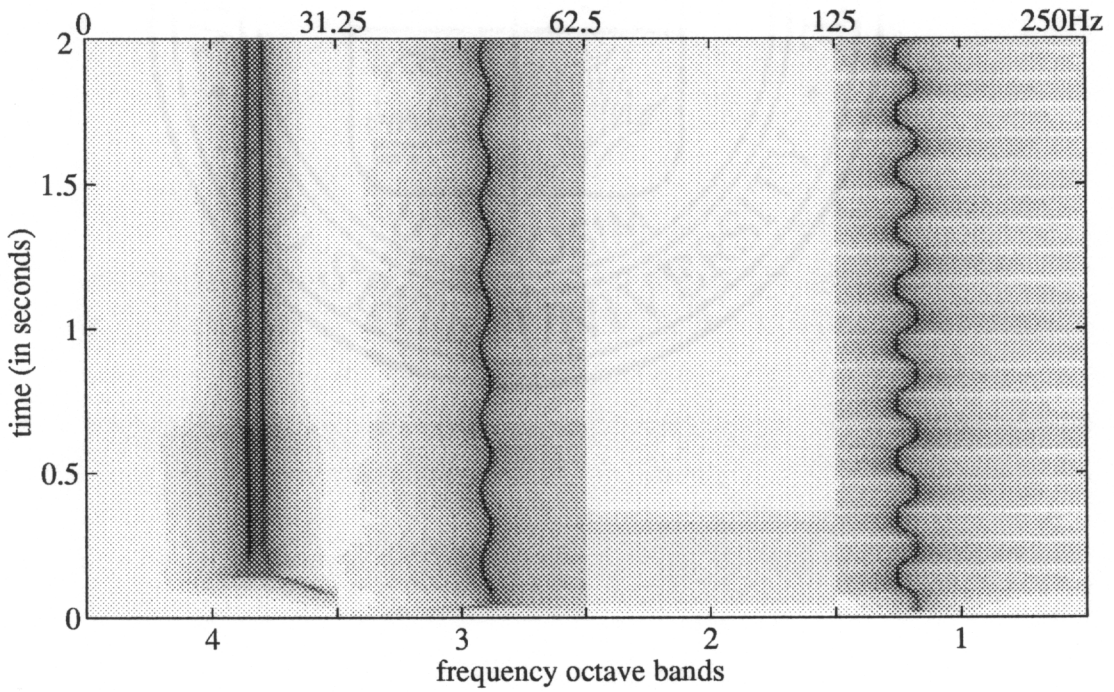
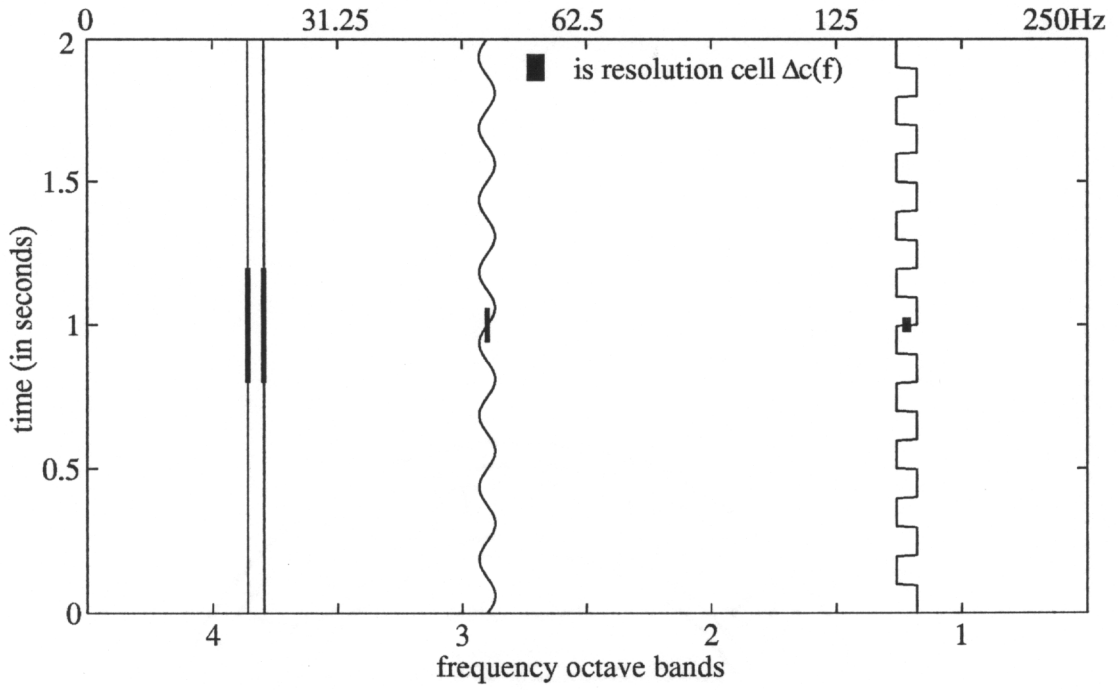
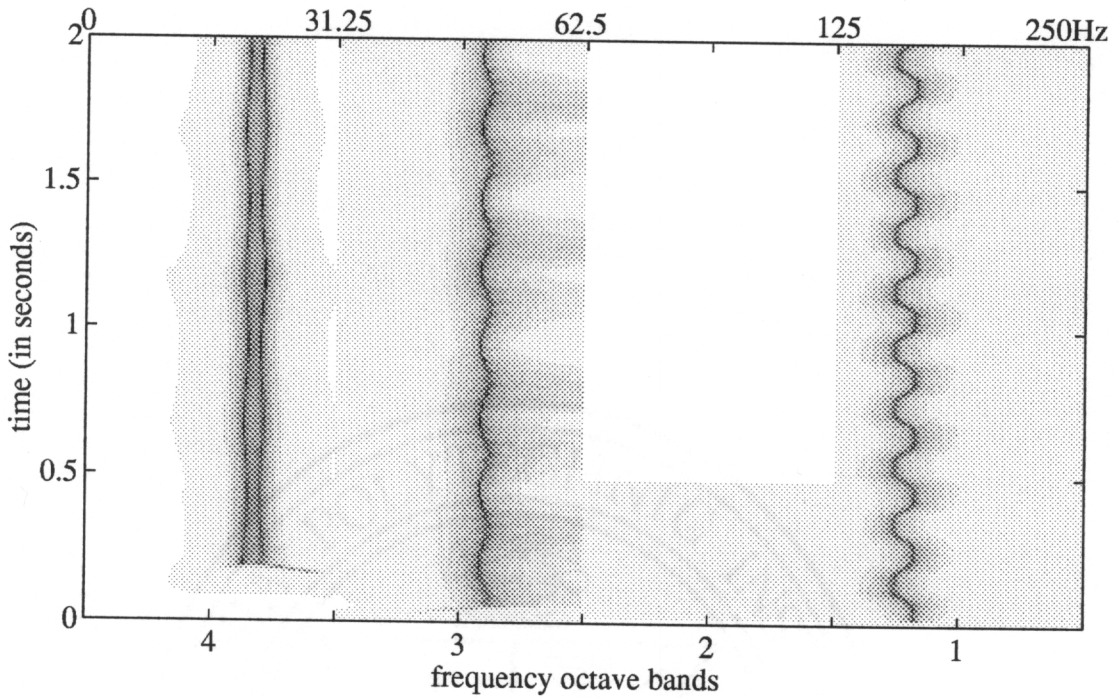


Figure 3.8 Testing signal (a, top), and the performance of MPSE-WRLS (b, bottom).



**Figure 3.8c** The performance of MPSE-Burg.

The fixed resolution AR spectral estimator used is the Burg algorithm. An 8th order AR model is chosen, and the sliding time windows used are 25, 50, 100, and 150 data points long respectively. For the longer window lengths,  $N_w \geq 50$ , we observed that the frequency estimates were some average of the constituent frequencies, with the time variations of the signal having been smoothed out. For the short window,  $N_w=25$ , all the frequency estimates show the switching pattern of  $f_3$ . In all 4 cases, the frequency resolutions of the estimator are too low to distinguish frequency components  $f_1$  and  $f_2$ , and none of the results retrieve the detail of  $f_4$ . Figure 3.9b gives an example where  $N_w$  equals 50.



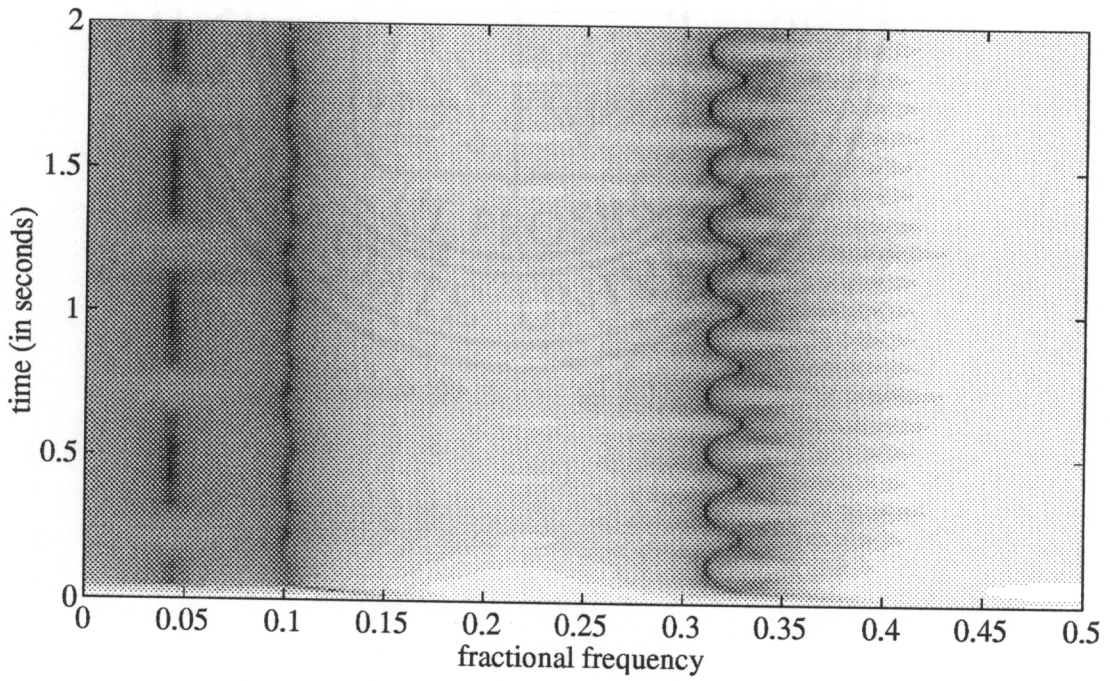
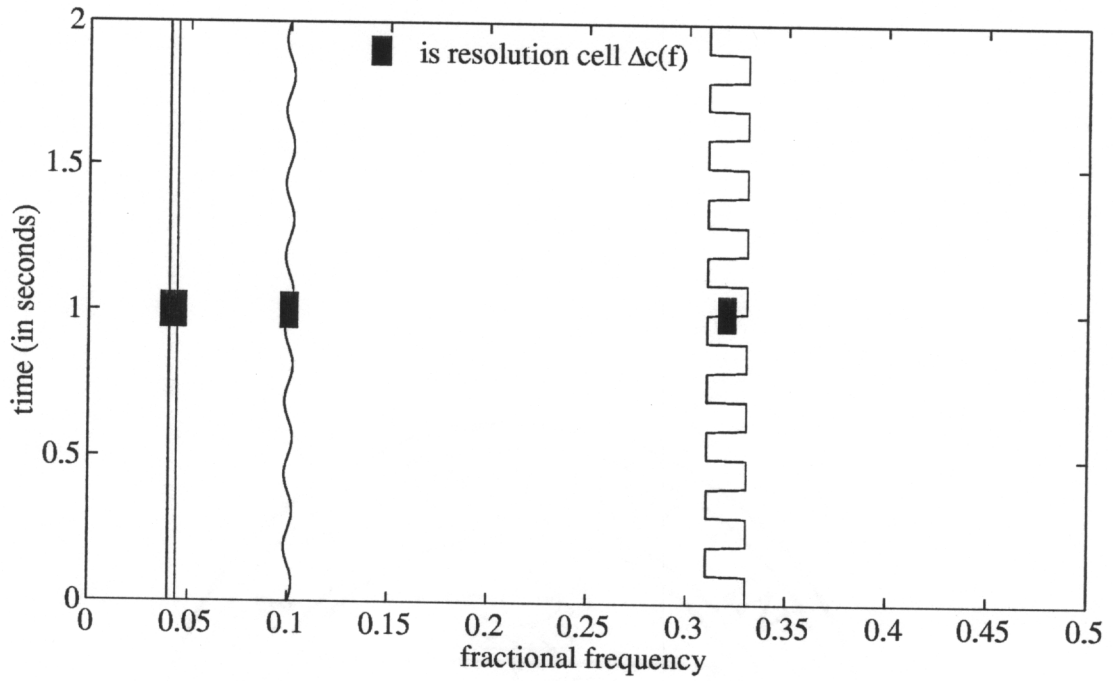


Figure 3.9 The testing signal (a, top), and the performance of AR-PSE (b, bottom).

In the second experiment the same test signal, represented by (3.2), is used. The performance of the wavelet transform will be compared with that of MPSE. To get the best resolution, the mother wavelet used is the Morlet wavelet, which is a Gaussian windowed complex tone. The parameter of Morlet's mother wavelet is the number of significant cycles  $N_c$  [113]. The number of samples in the daughter wavelet,  $N_w(f)$ , at a particular frequency  $f$ , can be defined as a function of  $N_c$

$$N_w(f) = \frac{N_c f_s}{f} \quad (3.17)$$

The resolutions for Morlet's mother wavelet are the same as for the Gaussian windowed Fourier transform, which were defined in Section 3.5 as:

$$\overline{\Delta f}_{N_w} = \frac{2.45}{N_w} \quad (\text{in cycles/sample}) \quad (3.18a)$$

$$\Delta n_{N_w} = 0.2045 N_w \quad (\text{in samples}) \quad (3.19b)$$

The condition for detecting signal components  $f_1$  and  $f_2$  is that the frequency resolution between  $f_1$  and  $f_2$  has to be smaller than the difference of  $f_1$  and  $f_2$ , i.e.

$$\max\{\overline{\Delta f}_{20}, \overline{\Delta f}_{22}\} < \frac{2}{f_s} = 4 \times 10^{-3} \quad (\text{in cycles/sample}) \quad (3.20a)$$

Using the inequality  $\overline{\Delta f}_{22} > \overline{\Delta f}_{20}$  in (3.5a), substituting in (3.4a), the  $f_1, f_2$  resolution condition becomes  $N_w(22) > 612$  samples, which via (3.3) can be represented as:

$$N_c \geq 27 \quad (\text{in mother wavelet cycles}) \quad (3.20b)$$

Some examples of frequency resolutions at 22 Hz, for different numbers of mother wavelet cycles  $N_c$ , are shown in Table 3.2. Note that for  $N_c$  equal to 25 the above frequency resolution condition is not satisfied, while for  $N_c$  equal to 35, 45, or 125 it is.

The signal component with  $f_3$  is associated with a time-varying spectrum. To detect  $f_3$  both the time and frequency resolutions have to satisfy certain requirements. The conditions for frequency and time resolution are, respectively

$$\max\{\overline{\Delta f}_{155}, \overline{\Delta f}_{165}\} < \frac{10}{f_s} = 20 \times 10^{-3} \quad (\text{in cycles/sample}) \quad (3.21a)$$

$$\max\{\Delta n_{155}, \Delta n_{165}\} \ll 0.1 \times f_s = 50 \quad (\text{in samples}) \quad (3.21b)$$

which, analogous to the  $f_1, f_2$  component frequency resolution, becomes  $N_w(165) > 122$  for the frequency resolution requirement, while the time resolution requirement leads to  $N_w(155) \ll 244$ . These respective requirements can be expressed equivalently as:

$$N_c \geq 41 \quad (\text{in mother wavelet cycles}) \quad (3.21c)$$

$$N_c \ll 75 \quad (\text{in mother wavelet cycles}) \quad (3.21d)$$

Note in Table 3.2 that the areas where the time and the frequency resolutions at 155 and 165 Hz are satisfied, barely overlap for  $N_c$  equal 45.

The signal component  $f_4$  varies continuously from 49 to 51 Hz. To retrieve the information of the variation, the frequency resolution must be much smaller than  $51 - 49 = 2$  Hz, so that

$$\max\{\overline{\Delta f}_{49}, \overline{\Delta f}_{51}\} \ll \frac{2}{f_s} = 4 \times 10^{-3} \quad (\text{in cycles/sample}) \quad (3.22a)$$

Also, the time resolution must be much smaller than the period of the variation in  $f_4$ . Therefore,

$$\max\{\Delta n_{49}, \Delta n_{51}\} \ll f_s / 4 = 125 \quad (\text{in samples}) \quad (3.22b)$$

If both the frequency difference and the variation period are discretized to the minimal two resolution bins, the frequency resolution condition is  $N_w(51) > 1,225$  samples, and the time resolution condition is  $N_w(49) < 306$  samples. These conditions become, respectively,

$$N_c \geq 125 \quad (\text{in mother wavelet cycles}) \quad (3.22c)$$

$$N_c \leq 30 \quad (\text{in mother wavelet cycles}) \quad (3.22d)$$

As both the time and frequency resolutions must be satisfied, both (3.22c) and (3.22d) must hold. It is clear that the  $f_4$  component cannot be resolved satisfactorily, in both time and frequency, even at the above minimal requirements. No number of mother wavelet cycles,  $N_c$ , is satisfactory; there is no overlap in Table 3.2 for the areas where the frequency resolution is satisfied and where the time resolution is satisfied for all the signal components.

Figure 3.10a,b shows an example of the Morlet wavelet transform scalogram with  $N_c$  equal to 40, which gets the "best" resolutions in time-frequency space for the given signal. The window width of the wavelet transform, the scale, varies from 80 to 1200 data points. The equivalent frequency region is from 16.67 Hz to 250 Hz ( $T_s = 0.002$  s). Comparison with the MPSE results, in Figure 3.8, shows that both methods exhibit multiresolution tracking capability. However, MPSE shows better resolution, both in time and in frequency, than the Morlet wavelet transform. In Figure 3.10b, the  $f_3$  component exhibits poor time and frequency resolutions, while no detail of the  $f_4$  component is retrieved with the Morlet wavelet transform.

**Table 3.2 The computations of the resolutions for Morlet wavelets.**

$N_c$	frequency $f$ (in Hz)			$N_w(f)$ (in samples)			$\overline{\Delta f}_{N_w}$ $\times 10^{-3}$			$\Delta n_{N_w}$ (in samples)		
	22	155	49	568	81	255	4.3	30	9.6	116	16.6	52.1
25	165	51			76	245		32	10		15.5	50.1
	22	155	49	795	113	357	3.1	22	6.9	163	23.1	73.0
35	165	51			106	343		23	7.1		21.7	70.1
	22	155	49	1023	145	459	2.4	17	5.3	209	29.7	93.9
45	165	51			136	441		18	5.6		27.8	90.2
	22	155	49	2841	403	1276	0.86	6.1	1.9	581	82.4	261
125	165	51			379	1225		6.5	2.0		77.5	251

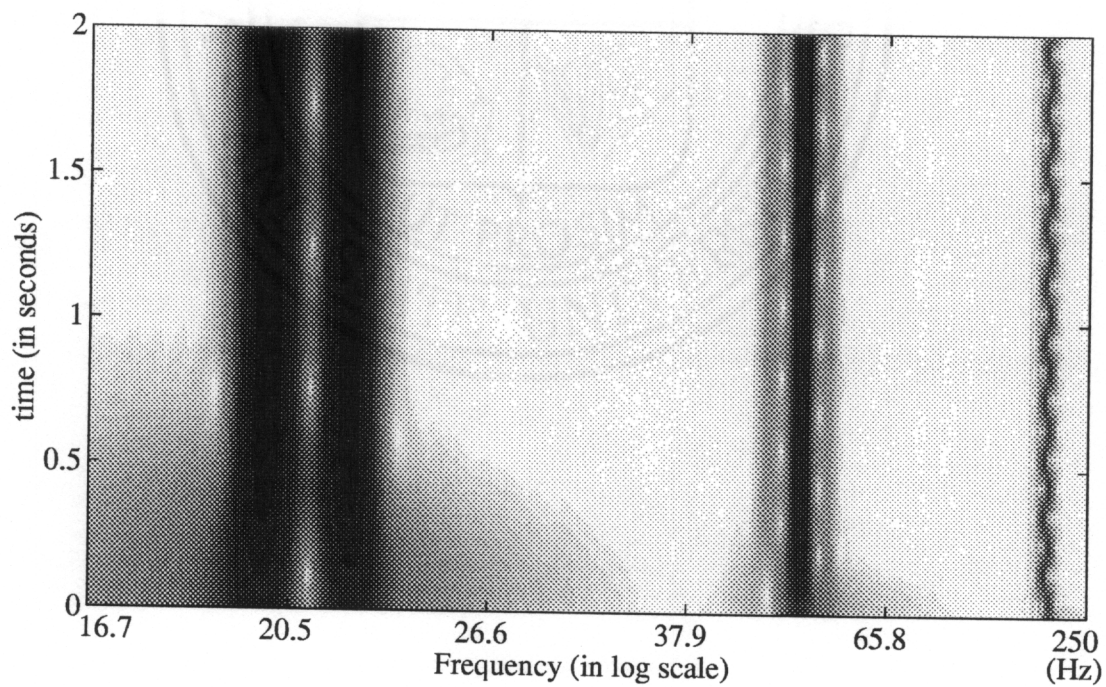
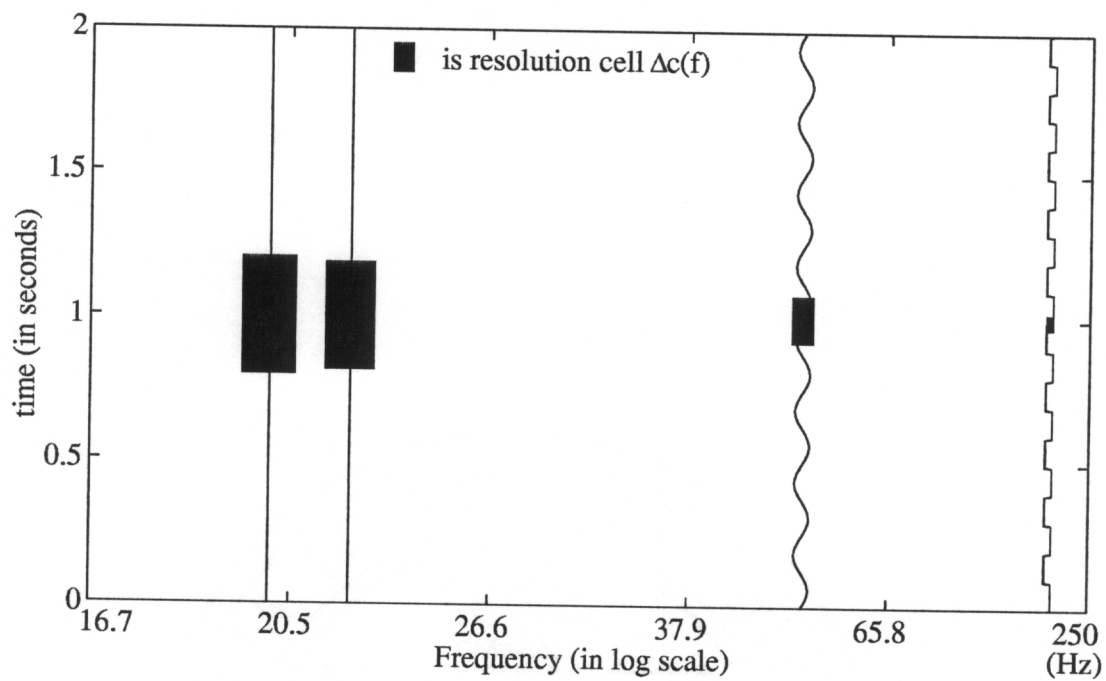


Figure 3.10 The testing signal (a, top), and the performance of the Morlet WT (b, bottom).

In addition, the computational requirements for the Morlet wavelet transform are much higher than for the MPSE method, because each Morlet wavelet decomposition is equivalent to a bank of FIR filter operations with the parameters of the Morlet wavelets. The final results of the wavelet transform required 1120 FIR filter operations with 80 to 1200 filter parameters, while the MPSE method uses only 4 adaptive WRLS filters to estimate the spectra. Just to give an indication: the MPSE computation for Figure 3.8b took 14.56 Mflops in Matlab, while those for the Morlet wavelet transform in Figure 3.10b took 9,483.6 Mflops.

The last experiment is designed to further explore the resolution characteristics of MPSE and the Fourier transform based methods. Here the MPSE is WRLS based and the Fourier transform based method used is still the Morlet wavelet transform since it reaches the lower bound defined by uncertainty principle. The testing signals are FM signals modulated by a sinusoid with different modulation depth and modulation frequency, i.e.

$$s(t) = \sin[\theta(t)] \quad (3.23)$$

where

$$\theta(t) = 2\pi 50t - \Delta m \frac{\cos(2\pi f_m t)}{f_m}$$

and its instantaneous frequency

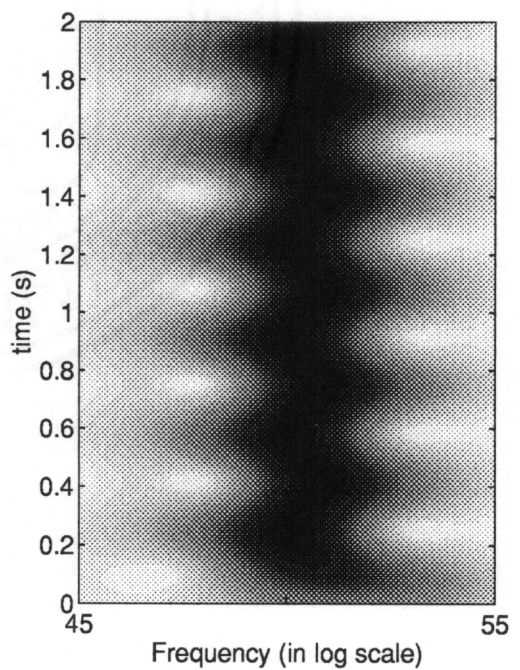
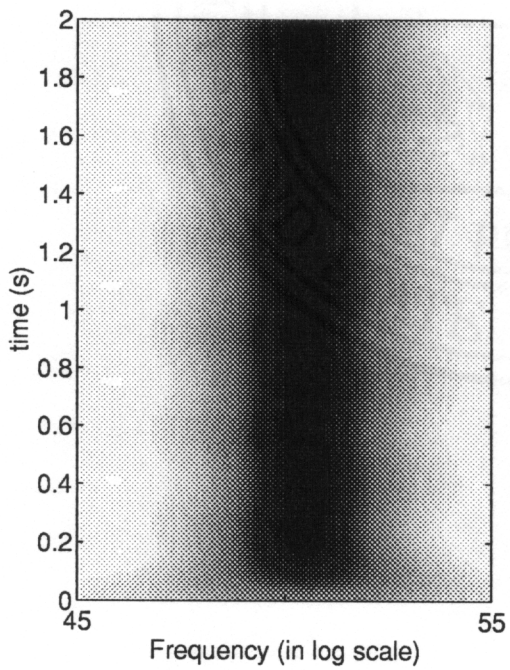
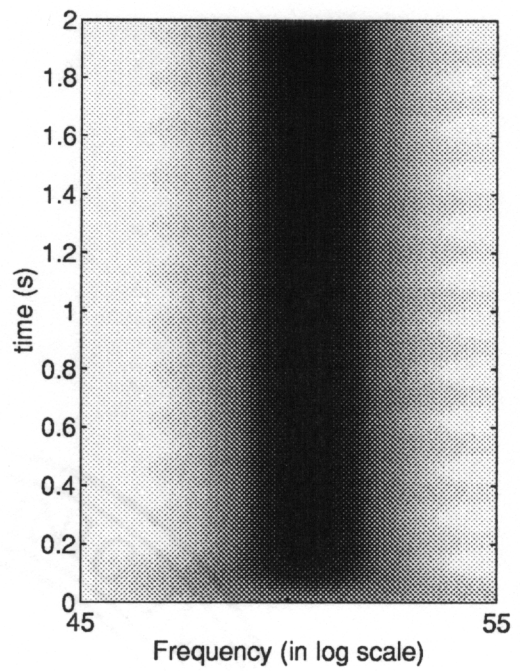
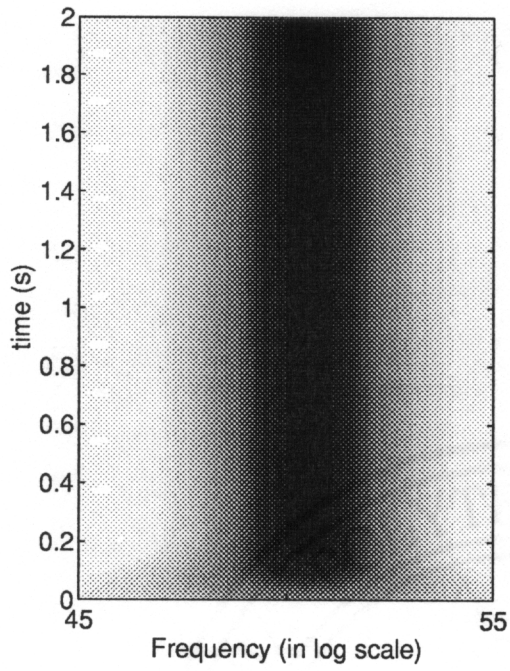
$$\begin{aligned} f(t) &= \frac{1}{2\pi} \frac{d\theta}{dt} \\ &= 50 + \Delta m \sin(2\pi f_m t) \end{aligned}$$

The parameter  $\Delta m$  is the modulation depth and  $f_m$  is the modulation frequency. These parameters are chosen to test the limits of the time and frequency resolutions of the processor. Zero-mean white noise is added to yield SNR equal to 40 dB. To detect the modulation signal, the estimator needs to satisfy required time and frequency resolutions; i.e. the time resolution of the estimator is better than the period of the modulation frequency, or  $\Delta t < \frac{1}{f_m}$ , and the frequency resolution is better than the modulation depth, or  $\Delta f < \Delta m$ . Table 3.3 provides 4 different sets of test signal parameters.

**Table 3.3 Required estimator resolutions for the test signals.**

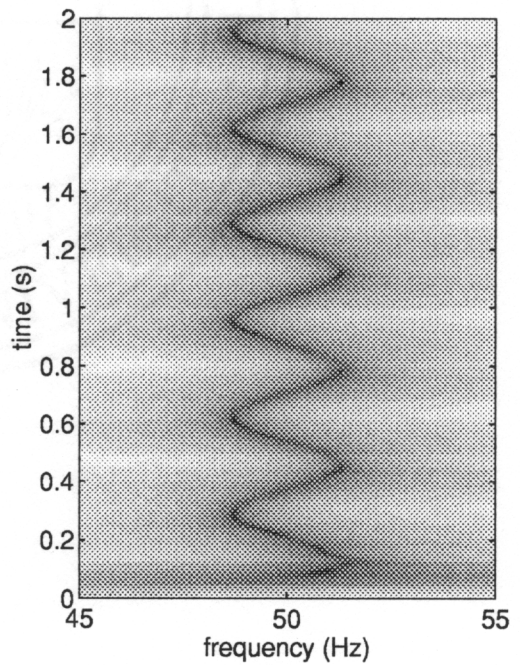
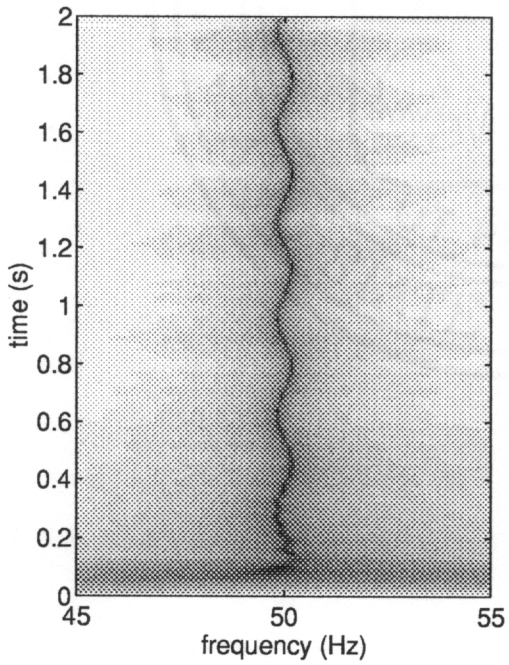
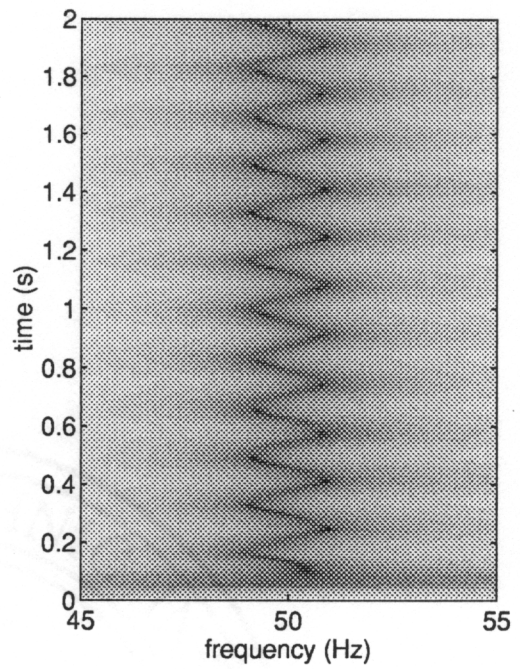
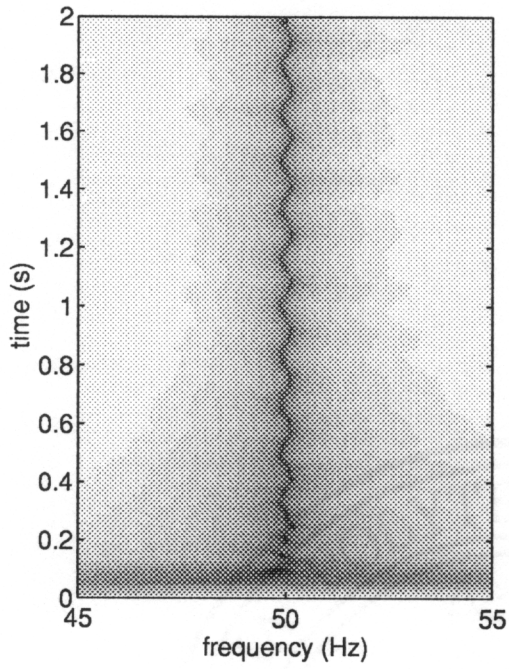
Parameter set	$\Delta m$	$\Delta f$	$f_m$	$\Delta t$	$\Delta c = \Delta t \times \Delta f$
$s_1$	$\pm 0.2$ Hz	$< 0.2$ Hz	6 Hz	$\Delta t < 0.167$ s	$\Delta c < 0.033$
$s_2$	$\pm 1.5$ Hz	$< 0.2$ Hz	3 Hz	$\Delta t < 0.333$ s	$\Delta c < 0.066$
$s_3$	$\pm 0.2$ Hz	$< 1.5$ Hz	6 Hz	$\Delta t < 0.167$ s	$\Delta c < 0.250$
$s_4$	$\pm 1.5$ Hz	$< 1.5$ Hz	3 Hz	$\Delta t < 0.333$ s	$\Delta c < 0.500$

From Table 3.3 we see that to resolve the first three test signals the product of the required time and frequency resolutions must be all smaller than the resolution lower bound, while to resolve the test signal  $s_4$  the product of  $\Delta f$  and  $\Delta t$  equals 0.5. In estimating all four test signals, the parameters of the estimators, the WRLS-based MPSE and the Morlet wavelet transform, are all kept the same. For MPSE the entire frequency band is divided into 4 octave bands. The order for octave band 1, 2, 4 is 0 since they are ‘empty.’ The order for the third octave band is 2. The time window width used is 10 data points after decimation by a factor of 4. In the Morlet wavelet transform, the number of significant cycles  $N_c$  in the time window is 40, which is chosen in favor of better time resolution. Figure 11 provides the results of the Morlet wavelet transform, and Figure 12 provides the results of WRLS-MPSE. In estimating  $s_1$ , since the time and frequency resolution requirements can not be satisfied, the Morlet estimate (Figure 3.11a, top left) can only provide a broad band trace of the signal without any time variation information. In estimating  $s_2$  (Figure 3.11b, bottom left), although the requirement for the time resolution has been reduced, the estimate from the Morlet wavelet transform still can not provide any temporal information as the result of poor frequency resolution. Again, in estimating  $s_3$  (Figure 3.11c, top right), reducing the frequency resolution requirement does not help to recover the temporal information when the time resolution is too low. The time variation pattern does show in estimation for  $s_4$  after the requirements for both time and frequency resolution are reduced. Notice the thickness of the spectral trace in all 4 cases, which reflects the frequency resolution of the Morlet wavelet transform. For the given  $N_c$  the related frequency resolution is about 3 Hz, as computed from (3.17) and (3.18a), with  $N_c = 40$ .



**Figure 11a-d. Performance of the Morlet wavelet transform (arranged from top left, bottom left, top right to bottom right).**





**Figure 12a-d. Performance of the WRLS-MPSE (arranged from top left, bottom left, top right to bottom right).**

The performance of WRLS-MPSE is demonstrated in Figure 12. All four test signals are correctly estimated by MPSE. There are transients of about 0.2 seconds at the beginning of each estimation. Meanwhile, the spectral traces are much thinner than for the wavelet estimates. This experiment again shows that MPSE has better time and frequency resolution than Fourier transform based methods. MPSE can also outperform the lower bound on resolution given by the Heisenberg uncertainty principle. This gives MPSE great potential in processing nonstationary signals.

### **3.7 Heisenberg uncertainty principle lower bound and resolution issues**

Comparing the resolution of MPSE and the Fourier transform-based methods, one is bound to ask why MPSE can have a resolution cell size below the lower bound given by the Heisenberg uncertainty principle. Reviewing the performance of the AR-PSE, we note that the AR-PSE methods emphasize the poles of the spectrum in order to produce better frequency localization and resolution than the WFT. However, the estimated magnitude at those poles usually exhibits a larger variance than for the WFT. In other words, the AR-PSE method trades off its magnitude resolution with frequency and time resolution.

There are three major characteristics which define a signal component. They are “time,” “frequency,” and “energy,” or sometimes the absolute value of the magnitude associated with the signal component. The Heisenberg uncertainty principle defines a boundary which preserves a low energy or magnitude estimation error. The definition of frequency resolution given in this study certainly does not preserve a low magnitude estimation error. Characteristically, for many applications, such as in signal detection, the more important information in the signal is not its magnitude or energy but the time and frequency characteristics, i.e. when and at what frequency does the signal appear.

Meanwhile, in conjunction with other magnitude estimation methods, the MPSE approach can give good time and frequency localization and resolution, while some other magnitude estimator can give good energy information based on the known [66] time and frequency information provided by MPSE. In that case, MPSE can obtain good time and frequency resolution without causing significant loss of magnitude resolution.

An interesting question, however, is raised. That is, whether or not the lower bound of resolution given by the Heisenberg uncertainty principle can be exceeded. If this lower bound is an insurmountable barrier, the trade-off in the magnitude resolution with the time and frequency resolutions in MPSE can not be recovered by any conjoint method; the efforts of seeking a conjoint magnitude estimator will then certainly be futile. If the resolution can go beyond this lower bound, then the following, more important, questions have to be answered. How general is the Heisenberg uncertainty principle? What is the scope of its application? How can we get maximum information from given samples? It is extremely important to understand these questions for the future direction of this research project.

To answer these questions, we have to first understand the real meaning of ‘uncertainty.’ In signal estimation, ‘uncertainty’ reflects how much can not be known about an object. The more that is known, the less that has to be estimated, and the less uncertainty in estimation there is. The uncertainty principle gives a lower bound which is the maximum information that can be retrieved based on what has been given. This lower bound varies with the *a priori* knowledge of the object.

The nonparametric estimation methods, such as the Fourier transform-based methods, rely on minimum knowledge of the object; that is, these methods make minor assumptions about the signal, and there is no specific model involved in the estimation process. The only assumption required for suitable nonparametric estimation is the stationarity of the signal, i.e. the unknown signal is at least wide sense stationary, or it has reached its stable state. However, parametric estimation methods are usually based on certain signal models, for example the AR, MA, or ARMA models. Parametric methods take advantage of the knowledge of the signal model in the estimation process. The quality of the final estimation result is, thus, strongly dependent on how well the selected model fits the *unknown* signal. If the model structure and its order are properly selected, the estimation can match the characteristics of the signal perfectly, and the resolution of estimation will certainly be better than for the nonparametric methods. If the selected model structure and/or its order are not correct, this leads to poorer estimation, and results are biased or otherwise misleading. In that case, no information can be retrieved from the given samples, the resolution of the estimator is non-existent.

The Heisenberg uncertainty principle is derived on the basis of knowing nothing about the object, so that the lower bound given by the Heisenberg uncertainty principle reflects the resolution limit

only for nonparametric estimation methods, not for parametric methods. Therefore, it is possible for a parametric estimator to outperform the nonparametric methods and to get a finer resolution than that indicated by the Heisenberg uncertainty principle. The key in successfully estimating the signal is to take maximum advantage of *a priori* knowledge with a minimum risk. If no knowledge about the object is available during the estimation stage, no assumption should be made in the estimation process since any erroneous assumption will generate false results. On the other hand, when some knowledge about the object is available, but the estimator can not make use of it to improve its performance, such an estimator is not considered a good or efficient one (in the sense of information usage). Therefore, the Fourier transform-based methods are more robust (or more risk-free) than the parametric methods as they are based on minimal assumptions about the objects under estimation. Also, Fourier transform-based estimators are low in efficiency, while the parametric methods are usually efficient but not robust.

### 3.8 Summary

The advantages of MPSE over Fourier transform-based methods are its capability to trade off magnitude resolution with time-frequency resolution, as well as time resolution with frequency resolution. Moreover, MPSE can take advantage of *a priori* knowledge to further improve its resolutions.

The advantages of MPSE over conventional fixed resolution AR-PSE is that MPSE can reduce the order of the AR process to that adequate for each subband. The required data length for estimating the power spectral density can thus be reduced, as the rule of thumb for the minimum data length to estimate AR parameters is “at least 3 times the order of the process [66].” MPSE also improves SNR by splitting the frequency band into multi-octave bands, which in turn increases the frequency resolution as shown in the next chapter.

## 4. DESIGN OF TIME-VARYING OPTIMAL FILTERS VIA MPSE

The optimal linear filter commonly known as the Wiener filter is the best estimator based on the minimum mean-square error criterion. That is, given the spectral characteristics of signal and noise, it provides the optimal estimate of the signal in the sense of minimum mean-square error. Optimal filtering has played an important role in SNR enhancement, signal detection and tracking, spectral estimation and system identification. Its theory and techniques have been successfully applied in such diverse fields as communication, control, radar, sonar, seismology, image processing, and biomedical engineering.

In this chapter, we study the special case of nonstationary signal recovery via time-varying Wiener filtering. The signal considered is a multi-frequency component nonstationary process, which can be modeled as a time-varying multi-pole (high order)  $AR(p)$  process. Contaminated with white noise, the observed data can be modeled as a nonstationary  $ARMA(p, p)$  process [66, 114]. Such a data model can be found in many practical situations, for instance, speech, multi-target radar or sonar signals, mechanical vibrations, and some electro-biological signals. The time-varying pattern of the signal components can be different, i.e. some may change continuously with time such as AM and/or FM signals, and some may exhibit on-off patterns.

The design of a Wiener filter requires knowledge of the statistics of the data to be processed. When the design information does not match the actual conditions, severe deterioration of the filtering process may result. Conventional estimators for contaminated AR processes are fixed resolution based, and thus mostly suitable for the stationary situation in which time resolution is not required. In nonstationary applications, the estimator must not only locate the signal components in frequency but also in time. A new method is proposed for realizing the time-varying Wiener filter, based on the multiresolution parametric spectral estimator (MPSE) [115, 116].

This chapter is organized as follows. Section 4.1 briefly reviews the background of optimal filtering for recovery of white noise contaminated signals. Section 4.2 covers the realizations of time-varying optimal Wiener filters, such as the causal and noncausal IIR forms and the FIR realization, all corresponding to the estimated time-varying model parameters. Section 4.3 provides

the implementation of MPSE for time-varying optimal Wiener filtering. In Section 4.4 and 4.5 we report on two computer experiments designed to test the performance of the MPSE-based optimal Wiener filter and compare it with using a conventional fixed resolution AR estimation technique. The resulting SNR improvements of these suboptimal forms are then compared with the SNR improvement from the noncausal IIR realization based on perfect (rather than estimated) knowledge.

#### 4.1 Optimal Wiener filtering background

Consider a real AR( $p$ ) process  $s_n$ , represented as:

$$s_n = -\sum_{i=1}^p a_i s_{n-i} + x_n \quad (4.1)$$

where  $\{a_i: i=1,2,\dots,p\}$  are the AR parameters, and  $x_n$  is the normally distributed white noise excitation with zero mean and variance  $\sigma_x^2$ . The Power Spectral Density function (PSD) of the time series  $s_n$  is defined by:

$$P_s(z) = \frac{\sigma_x^2}{A(z)A^*(z^{-*})}; \quad (4.2)$$

$$A(z) = \sum_{i=0}^p a_i z^{-i}; \quad a_0 \equiv 1$$

Complex conjugation is denoted by  $*$ , and  $^{-}$  indicates taking the reciprocal. Since  $s_n$  is real, the AR parameters are also real, and the roots of  $A(z)$  are either real or occur in complex conjugate pairs. We assume that the signal process is stable, with the roots of  $A(z)$  located inside the unit circle.

The observed time series  $y_n$  can be written as:

$$y_n = s_n + e_n \quad (4.3)$$

where  $e_n$  is the observation noise, which is a normally distributed white process with zero mean and variance  $\sigma_e^2$ . Under the assumption that the processes  $e_n$  and  $x_n$  are independent, the PSD of  $y_n$  is defined by:

$$\begin{aligned} P_y(z) &= P_s(z) + \sigma_e^2 \\ &= \frac{\sigma_x^2 + \sigma_e^2 A(z)A^*(z^{-*})}{A(z)A^*(z^{-*})} \end{aligned} \quad (4.4)$$

Using the optimal Wiener filter [117] we can write

$$H_{opt}(z) = \frac{P_{ys}(z)}{P_y(z)} = \frac{P_s(z)}{P_y(z)} \quad (4.5)$$

where  $P_{ys}(z)$  is the cross-PSD between  $y_n$  and  $s_n$ . The second equality results from  $e_n$  and  $x_n$  being independent, so that  $P_{ys}(z)$  equals  $P_s(z)$ .

Several difficulties arise in applying optimal filtering defined in (4.5) to signal estimation. The major difficulty is that the design of a Wiener filter requires *a priori* knowledge of the statistics of the data to be processed, such as the spectral density or correlation functions. Since such statistical characteristics of signal and noise are usually unknown before they have been separated, it becomes impossible in many practical cases to design and implement the optimal Wiener filter. Moreover, the performance of the filter depends greatly on the accuracy of the information on which the design of the filter is based. The filter is optimum only when the statistical characteristics of the signal and noise match the information used for design. When this information is contaminated by estimation errors, the design is no longer optimal. The estimation error could bring about serious distortion in the estimated signal. Therefore, the key to successful optimal filter design is to properly estimate the spectral or statistical characteristics of the input data.

The theory of Wiener filtering is not restricted to stationary environments. If the signal or noise is nonstationary, a time-varying optimal filter can be derived according to the time-varying characteristics of the input data [117]. Unlike the time invariant optimal Wiener filter, when the

signal  $s_n$  is a nonstationary process, its AR parameters and its spectrum are also time-varying. The general form of the Wiener filter presented in (4.5) is then a function of time, i.e.

$$H_{opt}(z, n) = \frac{P_{ys}(z, n)}{P_y(z, n)} \quad (4.6)$$

It requires the continuous availability of the time-varying information or estimates of the time-varying cross-PSD  $P_{ys}(z, n)$  and the PSD of the observation  $P_y(z, n)$ . This makes the design of the time-varying Wiener filter difficult. However, the time-varying characteristics of the signal can be tracked by a spectral estimator with a proper time window. Within the window, the signal is considered to be pseudo-stationary. The time-varying characteristics of the signal are updated in each sample interval while the time window slides along the time axis. The window should be short enough to ensure pseudo-stationarity. In a high SNR situation, a short window suffices to estimate the AR parameters. In a low SNR situation, the window must be long enough to yield acceptable variance of the estimator. As always, a trade-off between estimation variance and nonstationarity inference is required.

Several techniques have been proposed to estimate the model parameters for a contaminated AR process, such as the noise compensation method [114], general ARMA estimators, high order AR modeling [66], or prefiltering of the data to reduce the observation noise [118]. The noise compensation approach requires estimation of the autocorrelation function matrix and is sensitive to errors in the estimated noise variance, which could result in ill-conditioned correlation matrix estimates. General ARMA estimators require the solution of highly nonlinear equations, which will seldom be practical in a time-varying situation. The high order AR estimator suffers from large estimation variance and spurious poles. The prefiltering method is also seldom practical in the nonstationary situation, as it requires the application of a time-varying filter to track the changes in the signal. Moreover, as the original signal is a high order AR process, statistically consistent estimation of the model parameters requires a relatively long time window for all of the above methods. Consequently, this implies poor time resolution or tracking capability.

A new approach to time-varying Wiener filtering is presented here. The process consists of a two-stage procedure. The first stage is to estimate the statistical characteristics of the nonstationary



data by using the Multiresolution Parametric Spectral Estimator (MPSE) [99, 100] in conjunction with a noise compensation method. The second stage is to design the Wiener filter based on the estimated MPSE parameters, for subsequent filtering of the contaminated data. These two stages can run concurrently, so that an adaptive Wiener filtering operation is established.

## 4.2 Optimal Wiener filter realizations

The theoretical form of the optimal Wiener filter requires the spectral information of the signal and the noise. The spectral information can be estimated by some statistical estimators such as the AutoRegressive Parametric Spectral Estimators (AR-PSE). When the model parameters of the signal and the noise have been estimated, the coefficients of the optimal filter can thus be determined correspondingly. However, the ideal optimal filter is a noncausal IIR filter which is physically unrealizable. To perform the optimal filtering some approximations have to be made to make the filter practically feasible. There are different approximate realizations of the optimal Wiener filter, such as the Noncausal IIR realization, the time delayed Causal IIR realization, and the time delayed FIR realization. These realizations will be developed in this section. The derivations are mainly based on references [117, 119]

### 4.2.1 Noncausal IIR approximation of the optimal Wiener filter

Substituting (4.2) and (4.4) into (4.5), the optimal Wiener filter can be written as:

$$H_{opt}(z) = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_e^2 A(z)A^*(z^{-*})} \quad (4.7)$$

Note from (4.4) that the denominator of (4.7) can be spectrally factored.

$$\sigma_x^2 + \sigma_e^2 A(z)A^*(z^{-*}) = B(z)B^*(z^{-*}) \quad (4.8)$$

Using (4.8), the optimal Wiener filter can be rewritten as:

$$\begin{aligned}
 H_{opt}(z) &= \frac{\sigma_x^2}{B(z)B^*(z^{-*})} \\
 &= \left[ \frac{\sigma_x}{B(z)} \right] \left[ \frac{\sigma_x}{B^*(z^{-*})} \right]
 \end{aligned}
 \tag{4.9}$$

Equation (4.9) is the ideal optimal Wiener filter for recovering a white noise contaminated AR( $p$ ) process. Since the filter is a noncausal IIR filter, it is not physically realizable. However, if a block of observations  $\{y_n : n = 0, 1, \dots, N - 1\}$  is available, and the block length  $N$  is much longer than the transient of  $\sigma_x/B(z)$ , the noncausal IIR filter in (4.9) can be approximated well by an off-line procedure.

The procedure consists of two steps. First the observations  $\{y_n : n = 0, 1, \dots, N - 1\}$  are filtered by the causal filter  $\sigma_x/B(z)$ , and the output denoted  $v_n$ . Next, the anticausal filter  $\sigma_x/B^*(z^{-*})$  is implemented by filtering  $\tilde{v}_n$ , the time-reverse of the sequence  $v_n$ , with  $\sigma_x/B(z)$ , creating  $\tilde{s}_n$ , and finally time reversing  $\tilde{s}_n$ . The latter creates  $\hat{s}_n$ , the optimal Wiener filter estimate. Figure 4.1 provides the system diagram for the noncausal IIR approximation of the optimal Wiener filter, where  $\Psi(\cdot)$  denotes the time-reversing operation. Approximation errors are caused by the transients at both ends of the filtered data. Using (4.8) the filter parameters of  $\sigma_x/B(z)$  can be computed from  $A(z)$ ,  $\sigma_x^2$ , and  $\sigma_e^2$ , or their estimates.

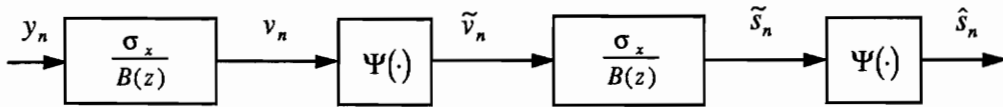


Figure 4.1 System diagram for the noncausal IIR optimal Wiener filter.

#### 4.2.2 Causal IIR approximation of the optimal Wiener filter

The noncausal IIR approximation of the Wiener filter can only be implemented off-line. To operate in real time, the anticausal part of the Wiener filter must be eliminated and/or replaced by

its causal approximation. Time delay is commonly used to reasonably compensate for approximation of the anticausal part of the filter.

From (4.4), (4.5), and (4.8), the optimal Wiener filter with a time delay  $D$ , i.e. using the observations  $\{y_i; i = 0, 1, \dots, n\}$  for estimating  $x_{n-D}$  rather than  $x_n$ , can be represented as:

$$H_{opt}(z; D) = \frac{P_s(z)}{B(z)B^*(z^{-*})} A(z)A^*(z^{-*})z^{-D} \quad (4.10)$$

$H_{opt}(z; D)$  in (4.10) can be factored as follows

$$H_{opt}(z; D) = \left[ \frac{A(z)}{B(z)} \right] \left[ \frac{P_s(z)}{B^*(z^{-*})} A^*(z^{-*})z^{-D} \right] \quad (4.11)$$

$$\underline{\underline{\Delta}} [H_{opt}^c(z)][H_{opt}^{nc}(z; D)]$$

where  $H_{opt}^c(z)$  is causal, and  $H_{opt}^{nc}(z; D)$  is noncausal. It has been proved that the causal IIR Wiener filter has the form [117]

$$H_{opt}^{IIR}(z; D) = H_{opt}^c(z)[H_{opt}^{nc}(z; D)]^+ \quad (4.12)$$

where  $[H_{opt}^{nc}(z; D)]^+$  represents the  $z$ -transform of the impulse response  $h(k)$  of  $H_{opt}^{nc}(z; D)$  restricted to its causal part.

Using (4.2) and (4.11),  $H_{opt}^{nc}(z; D)$  can be rewritten as:

$$H_{opt}^{nc}(z; D) = \frac{\sigma_x^2 z^{-D}}{A(z)B^*(z^{-*})} \quad (4.13)$$

$$= \left[ \frac{F(z)}{A(z)} + \frac{G(z)}{B^*(z^{-*})} \right] \sigma_x^2 z^{-D}$$

where  $F(z) = \sum_{i=0}^{p-1} f_i z^{-i}$  and  $G(z) = \sum_{i=-p}^{-1} g_i z^{-i}$ , and where  $F(z)$ ,  $G(z)$ ,  $A(z)$ , and  $B^*(z^{-*})$  are related

by the following equation

$$F(z)B^*(z^{-*}) + G(z)A(z) = 1 \quad (4.14)$$

The parameters of  $F(z)$  and  $G(z)$  can be determined by comparing the coefficients of similar powers of  $z$  on both sides of (4.14). That is,

$$\begin{pmatrix} b_{-p}^* & 0 & \cdots & 0 & a_0 & 0 & \cdots & 0 \\ b_{-p+1}^* & b_{-p}^* & \ddots & \vdots & a_1 & a_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 & \vdots & \vdots & \ddots & 0 \\ b_{-1}^* & b_{-2}^* & \cdots & b_{-p}^* & a_{p-1} & a_{p-2} & \cdots & a_0 \\ b_0^* & b_{-1}^* & \cdots & b_{-p+1}^* & a_p & a_{p-1} & \cdots & a_1 \\ 0 & \ddots & \ddots & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & b_0^* & b_{-1}^* & \vdots & \ddots & a_p & a_{p-1} \\ 0 & \cdots & 0 & b_0^* & 0 & \cdots & 0 & a_p \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{p-1} \\ g_{-p} \\ g_{-p+1} \\ \vdots \\ g_{-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (4.15)$$

If the time delay  $D = 0$ , we see from (4.13) that

$$[H_{opt}^{nc}(z; 0)]^+ = \frac{F(z)}{A(z)} \sigma_x^2 \quad (4.16)$$

and the causal IIR Wiener filter in (4.12) can be represented as

$$H_{opt}^{IIR}(z; 0) = \frac{F(z)}{B(z)} \sigma_x^2 \quad (4.17)$$

For a time delay  $D > 0$ , we find from (4.13)

$$[H_{opt}^{nc}(z; D)]^+ = \frac{F(z)}{A(z)} \sigma_x^2 z^{-D} + \left[ \frac{G(z)}{B^*(z^{-*})} z^{-D} \right]^+ \sigma_x^2 \quad (4.18)$$

Let  $W(z; D) = \sum_{i=-D}^{-1} \gamma_i z^{-i-D}$ , where  $\{\gamma_i; i = -\infty, \dots, -1\}$  is the noncausal impulse response of

$\frac{G(z)}{B^*(z^{-*})}$ . The causal IIR Wiener filter in (4.12) can then be represented as

$$H_{opt}^{IIR}(z; D) = \frac{F(z)z^{-D} + W(z; D)A(z)}{B(z)} \sigma_x^2 \quad (4.19)$$

### 4.2.3 FIR approximation of the optimal Wiener filter

The ideal Wiener filter can also be approximated by an FIR filter. Figure 4.2 provides the system diagram for an FIR realization of the Wiener filter. In the diagram, the estimate of the signal at time  $n - D$  is a linear combination of the observations  $\{y_i: i = n, \dots, n - L + 1\}$ , i.e.

$$\hat{s}_{n-D} = \sum_{k=0}^{L-1} h_k y_{n-k} \quad (4.20)$$

where  $L$  is the filter length. The estimation error is

$$e_{n-D} = s_{n-D} - \hat{s}_{n-D} \quad (4.21)$$

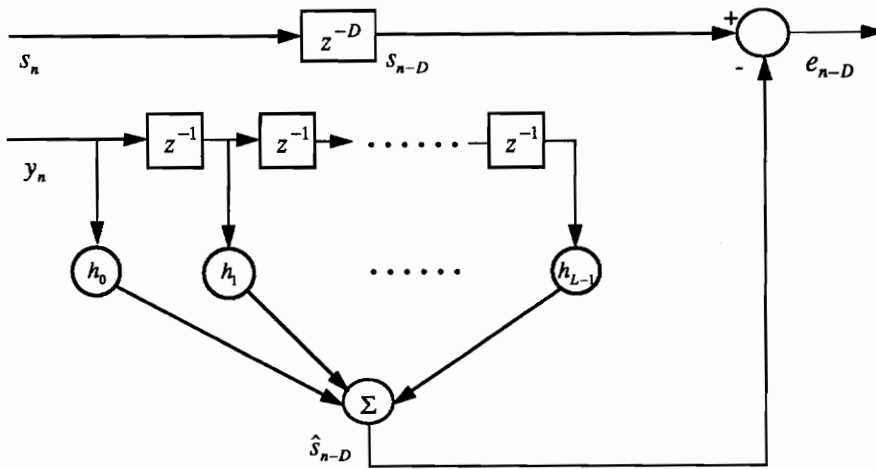


Figure 4.2 System diagram for the FIR approximation to the Wiener filter.

The weighting coefficients of the linear summation, the FIR Wiener filter parameters, can be determined using the least mean square criterion. Based on that criterion,  $e_{n-D}$  is statistically independent of the observations  $\{y_i: i = n, \dots, n - L + 1\}$ , i.e.,

$$\begin{aligned}
E\{y_{n-l}^* e_{n-D}\} &= E\{y_{n-l}^* s_{n-D}\} - E\left\{y_{n-l}^* \sum_{k=0}^{L-1} h_k y_{n-k}\right\} \\
&= 0 \quad \text{for } l = 0, \dots, L-1
\end{aligned} \tag{4.22a}$$

or

$$r_{ys}(l-D) = \sum_{k=0}^{L-1} h_k r_y(l-k); \quad \text{for } l = 0, \dots, L-1 \tag{4.22b}$$

Note that  $r_{ys}(k) = r_s(k)$ , and that (4.22b) can be written in matrix form

$$\begin{pmatrix} r_y(0) & r_y(-1) & \cdots & r_y(-L+1) \\ r_y(1) & r_y(0) & \cdots & r_y(-L+2) \\ \vdots & \vdots & \ddots & \vdots \\ r_y(L-1) & r_y(L-2) & \cdots & r_y(0) \end{pmatrix} \begin{pmatrix} h_0 \\ h_1 \\ \vdots \\ h_{L-1} \end{pmatrix} = \begin{pmatrix} r_s(-D) \\ r_s(-D+1) \\ \vdots \\ r_s(L-1-D) \end{pmatrix} \tag{4.23a}$$

or

$$R_y \underline{h} = \underline{r}_s \tag{4.23b}$$

where  $R_y$  is the autocorrelation matrix of  $y_n$ ,  $\underline{h} = [h_0 \ h_1 \ \dots \ h_{L-1}]^T$  is the FIR Wiener filter impulse response, and  $\underline{r}_s = [r_s(-D) \ r_s(-D+1) \ \dots \ r_s(L-1-D)]^T$  is part of the autocorrelation of  $s_n$ . The optimal Wiener filter parameters can be solved from (4.23). Since  $s_n$  is an AR( $p$ ) process, there is a one-to-one relation between the autocorrelation function of  $s_n$  and its AR parameters. The autocorrelation function of  $s_n$  can be computed according to the following relation.

$$r_s(k) = \begin{cases} -\sum_{i=1}^p a_i r_s(k-i) & \text{for } k \geq 1 \\ -\sum_{i=1}^p a_i r_s(-i) + \sigma_x^2 & \text{for } k = 0 \end{cases} \tag{4.24}$$

When the AR parameters and the variance of the excitation noise or their estimates are known, the autocorrelation function  $r_s(k)$  in (4.24) can be solved for by using the efficient Step-Down algorithm [66].

Since  $y_n$  is a white noise contaminated observation of  $s_n$ ,

$$r_y(k) = \begin{cases} r_s(k) & \text{for } k \neq 0 \\ r_s(0) + \sigma_e^2 & \text{for } k = 0 \end{cases} \quad (4.25)$$

The autocorrelation function matrix of  $y_n$  can be represented as:

$$R_y = R_s + \sigma_e^2 \mathbf{I} \quad (4.26)$$

where  $\mathbf{I}$  is the identity matrix. Using (4.23)-(4.26), the FIR Wiener filter parameters can be determined based on estimates of the AR parameters, the power of the excitation process and the power of the white contaminating noise.

### 4.3 Time-varying MPSE Wiener filtering

The MPSE parameters are the parameters of the noise-contaminated observation  $y_n$ . To improve the corrupted estimates, a noise compensation technique [114] can be used in conjunction with MPSE. The noise compensation method is based on the relation between the autocorrelation functions of  $y_n$  and  $s_n$ . That is, the estimated autocorrelation function of the signal can be determined using  $\hat{R}_s = \hat{R}_y - \hat{\sigma}_e^2 \mathbf{I}$ . The autocorrelation functions can be used either directly, to construct an FIR type Wiener filter, or indirectly, to determine the signal component parameters for constructing an IIR type Wiener filter or to estimate the time-varying spectrum of the signal component.

In situations where the SNR is extremely low, a two step estimation stage can help to reduce the noise level in the final results (Figure 4.3). The first step is to prefilter the noise-contaminated signal component in each non-empty octave band using an FIR type Wiener filter at the subband level. This estimation-step optimal filter is constructed from the autocorrelation functions estimated

by MPSE with noise compensation. The output from the first step,  $\hat{s}_n^{(i)}$ , an estimate of the signal component, is then used to re-estimate the AR parameters of the subband signal component. Since the SNR condition has been improved by the prefiltering step, the re-estimated signal parameters are less corrupted by noise than without the prefiltering step. However, extra distortion can be caused by prefiltering, which usually produces over-smoothed estimates. The choice of a one or two step estimation stage is based on the trade-off between further noise reduction and extra distortion. Under high SNR conditions, a one step estimation stage yields better results than a two step estimation stage. The experimental results presented in the next section were produced by two stage optimal filtering when SNR is about 0 dB and with the one stage approach when SNR exceeds 20 dB. The two stage approach performs better than the one stage realization under low SNR conditions while the one stage approach produces better results than the two stage approach under high SNR conditions.

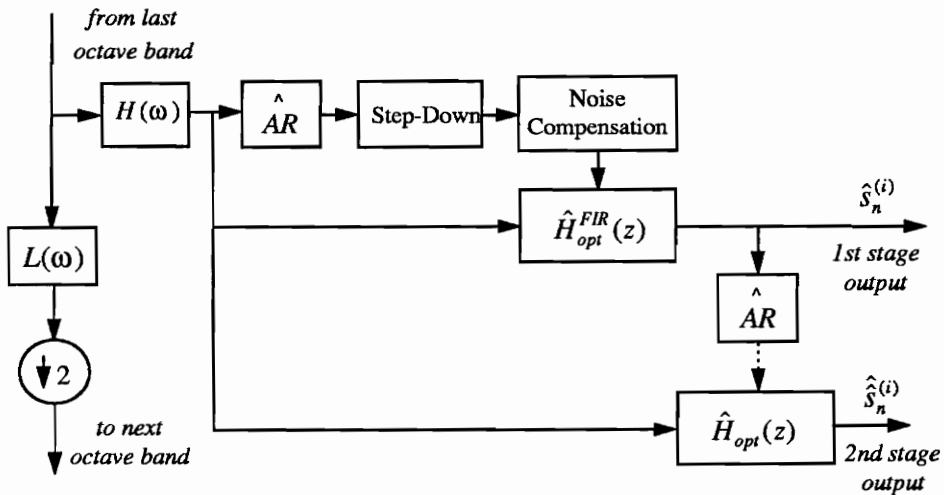


Figure 4.3 Structure for the MPSE procedure.

If the time-varying characteristics of the signal components are of major interest, the optimal Wiener filtering operation can be performed independently in each octave band. The filter



parameters can be determined according to the estimation process in each octave band. The final result is a set of decomposed signal components or their spectral estimates from a bank of lower order optimal filters. Performing the optimal filtering in each subband can provide 'optimal' conditions for the different frequency components. Lower order filters are also more stable, with shorter transients than higher order filters.

If the waveform of the signal  $s_n$  is of major interest, the optimal filtering has to be performed over the entire frequency band to prevent waveform distortion caused by different group delays. After the signal has been decomposed into the multi-frequency components in each octave band, the AR polynomial  $A_j(z)$ , for the  $j$ -th octave band can be represented as :

$$A_j(z) = \prod_{i=1}^{p_j} (1 - \rho_{ij} z^{-1}) \quad j = 1, \dots, M \quad (4.28)$$

where  $p_j$  is the order of the signal component in the  $j$ -th octave band, and  $M$  is the total number of octave bands.

The collection of all roots  $\rho_{ij}$  is exactly the set of roots for the overall AR polynomial  $A(z)$ , except that the root positions have been changed due to decimation. To reconstruct  $A(z)$ , the roots in each octave band need to be projected from its octave band to the entire frequency band. Writing the roots in (4.2) in polar form:

$$\rho_{ij} = \gamma_{ij} \exp(j\theta_{ij}) \quad (4.29)$$

the projection from the octave band to the entire frequency band is reflected in

$$\rho'_{ij} = \gamma_{ij} \exp(j \frac{\theta_{ij}}{2^{j-1}}) \quad (4.30)$$

$\hat{A}(z)$  can then be approximated as

$$\hat{A}(z) = \prod_{j=1}^M \prod_{i=1}^{p_j} (1 - \rho'_{ij} z^{-1}) \quad (4.31)$$

From (4.1) and (4.3) we see

$$\sum_{i=0}^p a_i y_{n-i} = x_n + \sum_{i=0}^p a_i e_{n-i} \quad (4.32)$$

Taking the variance on both sides of (4.32) yields

$$E\left(\left|\sum_{i=0}^p a_i y_{n-i}\right|^2\right) = \sigma_x^2 + \sigma_e^2 \sum_{i=0}^p |a_i|^2 \quad (4.33)$$

where we used the condition that the excitation process and the observation noise are zero mean white and independent processes. Finally, the excitation noise of the signal process can be approximated as

$$\hat{\sigma}_x^2 = \frac{1}{N} \left( \sum_{i=p}^N \left| \sum_{l=0}^p \hat{a}_l y_{i-l} \right|^2 \right) - \hat{\sigma}_e^2 \sum_{i=0}^p |\hat{a}_i|^2 \quad (4.34)$$

where  $N$  is the number of data points in the time window. After the estimates  $\hat{A}(z)$ ,  $\hat{\sigma}_e^2$  and  $\hat{\sigma}_x^2$  have been determined, the Wiener filter for the entire frequency band can be constructed.

Equations (4.31) and (4.34) provide approximations for constructing the second stage optimal Wiener filter for the entire frequency band, or global Wiener filter, based on the estimates in each subband, as obtained in the first stage. When the poles of the signal components are located close to the unit circle, the approximation is a good representation of the actual signal model; otherwise, some error in the estimation of excitation noise power may result. Fortunately, the performance of the global Wiener filter depends much more on correct pole positions than on correct noise powers. If the spectral estimator can track the changes of the signal components and provide correct pole positions over time, the signal components will pass the filter. A filter constructed from incorrect pole information can cause serious distortion, even filter out the actual signal. The gains and bandwidths centered at each pole are controlled largely by the excitation noise and the observation noise. With noise compensation, over-estimation of the observation noise power will produce over-smoothed results (narrower passband). It may also cause  $\hat{r}_y(0) - \hat{\sigma}_e^2 < 0$  (negative PSD), which

could cause serious distortion in the filtering. To prevent the negative PSD a threshold  $\delta > 0$  can be used, such that the constraint  $\hat{r}_y(0) - \hat{\sigma}_e^2 \geq \delta$  is valid, by reducing  $\hat{\sigma}_e^2$  as necessary.

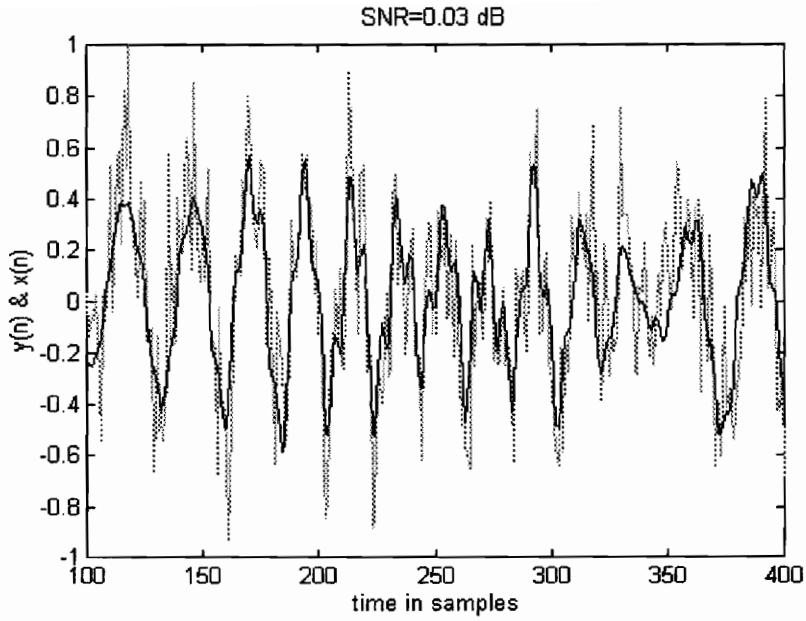
#### 4.4 Examples of optimal filtering of noise contaminated nonstationary signal

The experiment designed in this section provides examples of optimal filtering of noise contaminated nonstationary signals. In the experiment a time-varying AR(4) process in wideband noise is used, with SNR equal to 0 dB (Figure 4.4a). The poles of the AR(4) process are defined by:

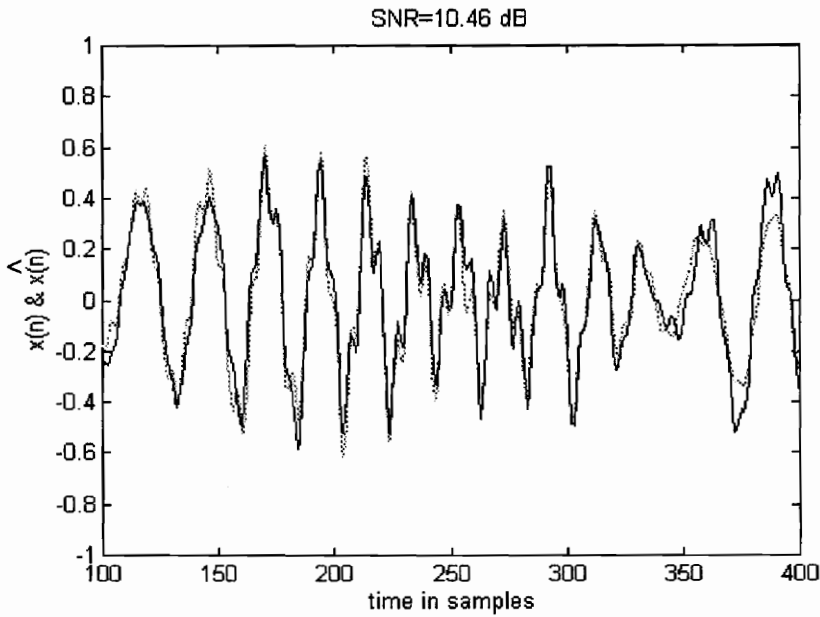
$$\begin{aligned} \rho_{1,2}(n) &= 0.99 \exp[\pm j2\pi(0.0625 + 0.0375 \cos(0.0112n))] \\ \rho_{3,4}(n) &= 0.99 \exp[\pm j2\pi(0.0313 - 0.0188 \cos(0.0112n))] \end{aligned} \quad \begin{array}{l} \text{(cycles/sample)} \\ \end{array} \quad (4.35)$$

The excitation noise  $\sigma_x^2$  of the AR process is a normally distributed process with zero mean and variance  $10^{-4}$ . The global Wiener filter realizations used are all noncausal IIR approximations. Given the true time-varying parameters, the ideal Wiener filter provides 10.46 dB of noise reduction (Figure 4.4b). The Wiener filter updated from a conventional fixed resolution windowed RLS algorithm [101], a special form of the Kalman filter, achieved 4.09 dB of noise reduction (Figure 4.4c), whereas the MPSE-based Wiener filter realized 8.73 dB of noise reduction (Figure 4.4d). The MPSE Wiener filter is constructed by projecting the estimated parameters for each sub-process from its own octave band to the global frequency band.

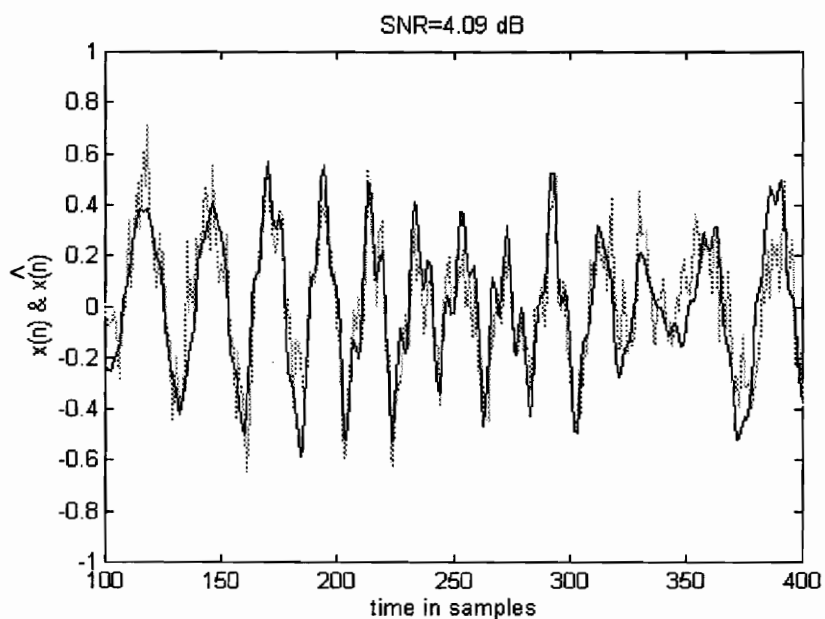
The same signals were also used to test the performance of different realizations of the time-varying Wiener filter, i.e. the non-causal IIR filter (NC-IIR), the causal IIR filter (C-IIR), and the FIR filter. The results are listed in Table 4.1. The experiments are grouped into two groups based on the condition of whether the time-varying AR parameters for designing the optimal filter are assumed to be known a priori or not. The measurement of performance is based on the mean square error  $P_e$  between the signal  $s_n$  and the optimal filter estimate  $\hat{s}_n$ , as well as the SNR of the optimal filter estimate. Here the mean square error  $P_e$  is estimated by:



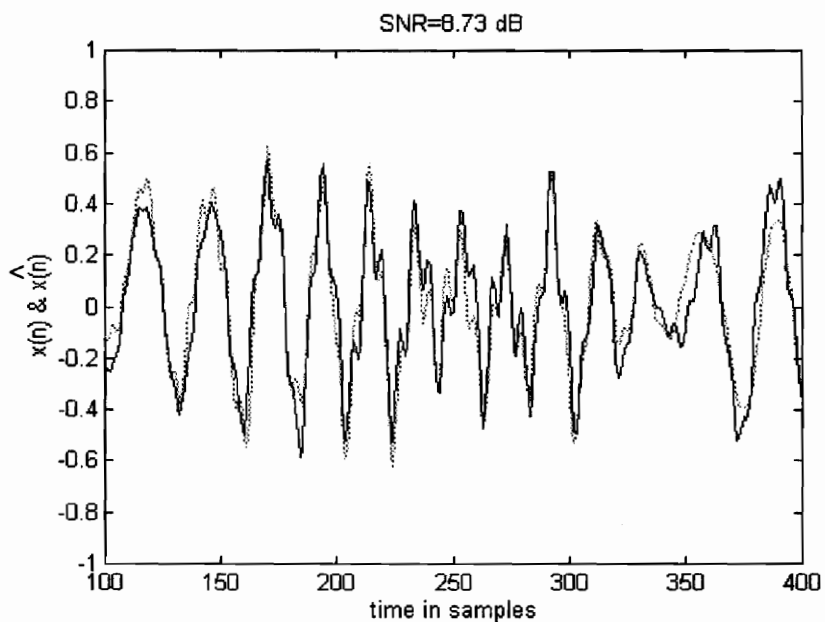
**Figure 4.4a** Signal  $x(n)$  (solid) and noise-contaminated observation  $y(n)$  (dotted).



**Figure 4.4b** Signal (solid) and estimate based on the Wiener filter associated with perfect knowledge (dotted).



**Figure 4.4c** Signal (solid) and estimate based on the Wiener filter associated with knowledge based on direct AR estimation (dotted).



**Figure 4.4d** The signal (solid), and the estimate (dotted) based on the Wiener filter associated with knowledge based on MPSE estimation.

$$\hat{P}_e = \frac{1}{N} \sum_{n=0}^{N-1} (s_n - \hat{s}_n)^2 ; \quad (4.36)$$

where  $N$  is the total number of samples in the estimate; the SNR is defined as:

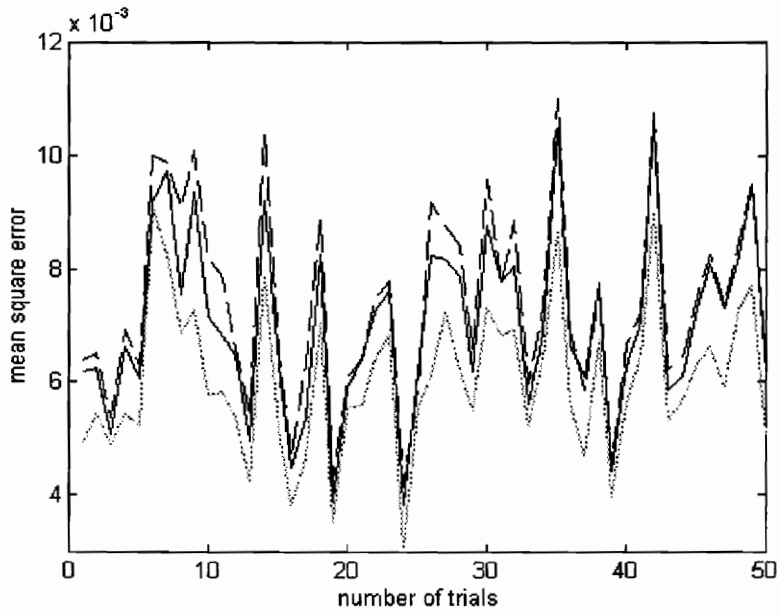
$$SNR = 10 \log_{10} \left( \frac{P_{s\hat{s}}}{\hat{P}_e} \right) dB \quad (4.37a)$$

where  $P_{s\hat{s}}$  is the cross covariance between the signal  $s_n$  and its estimate  $\hat{s}_n$ , which is estimated by:

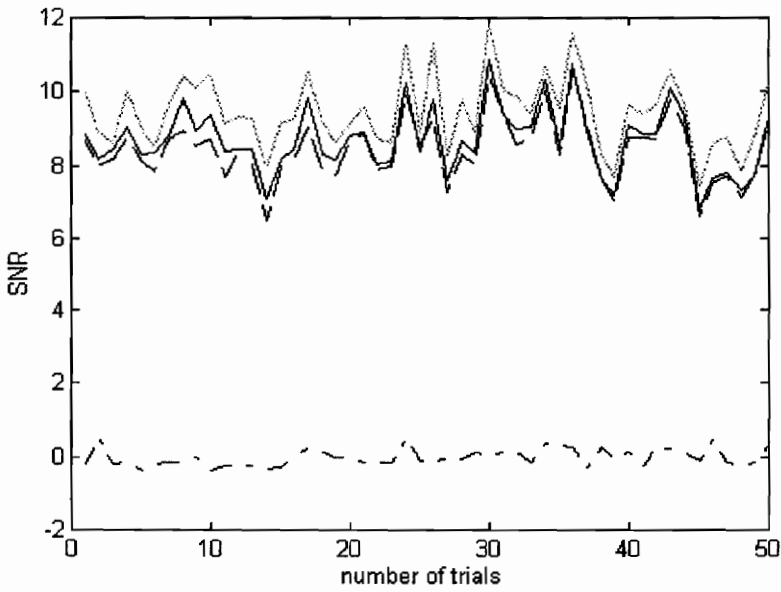
$$\hat{P}_{s\hat{s}} = \frac{1}{N} \sum_{n=0}^{N-1} s_n \hat{s}_n \quad (4.37b)$$

Since in this experiment the SNR of the observation is about zero dB, the SNR of the estimate is also the SNR improvement of the optimal Wiener filter. The total data length  $N$  used in this experiment is 1000, after 100 data points of transient having been removed from both ends of the estimated sequence. The time delay  $D$  used for the causal IIR and FIR Wiener filters is 20 samples. The length  $L$  of the FIR Wiener filter is 40 samples.

Table 4.1 shows that when the time-varying AR parameters are known a priori, the NC-IIR provides the best performance; the FIR performs in second place; while the C-IIR filter performance is the worst among all three forms. However, when the time-varying AR parameters are unknown and have to be estimated from the observations, the performance ranking is reversed. That is, the performance of the C-IIR filter is best; the FIR filter is next; and the NC-IIR filter is ranked last among three filter forms. To explain the performance ranking under different conditions, we executed the same test for 50 trials for all three optimal filter forms under the condition that the signal AR parameters were known a priori. Figure 4.5a, b provides the SNRs and the mean square errors for 50 trials of all three filter forms. This figure shows that the performance ranking of the optimal filters is consistent for each trial. Table 4.2 provides the mean values of the SNRs and their standard deviation. For comparison, the table also provides the performance ranking under time-invariant conditions by freezing the AR parameters at the values for  $n$  equal 0.



**Figure 4.5a** Mean square errors for the NC-IIR (dotted), C-IIR (dashed), and FIR Wiener filters.



**Figure 4.5b** SNRs for the NC-IIR (dotted), C-IIR (dashed), and FIR Wiener filters.

**Table 4.1 Performance of the optimal Wiener filter realizations.**

Filter forms \ Conditions	A priori known AR parameters	Estimated AR parameters
Non-causal <i>IIR</i>	$Pe = 4.7 \times 10^{-3}$ $SNR = 10.46dB$	$Pe = 7.1 \times 10^{-3}$ $SNR = 8.73dB$
Causal <i>IIR</i> with $D = 20$	$Pe = 5.9 \times 10^{-3}$ $SNR = 9.41dB$	$Pe = 7.0 \times 10^{-3}$ $SNR = 8.93dB$
<i>FIR</i> with $D = 20; L = 40$	$Pe = 5.4 \times 10^{-3}$ $SNR = 9.80dB$	$Pe = 7.1 \times 10^{-3}$ $SNR = 8.87dB$

**Table 4.2 SNRs of the optimal Wiener filters under time-varying and time-invariant conditions.**

Filter forms \ Conditions	Time varying $\overline{SNR} \pm \text{standard deviation}$	Time invariant $\overline{SNR} \pm \text{standard deviation}$
Non-causal <i>IIR</i>	$9.44 \pm 0.98 \text{ dB}$	$10.60 \pm 0.85 \text{ dB}$
Causal <i>IIR</i> with $D = 20$	$8.43 \pm 0.88 \text{ dB}$	$10.34 \pm 0.81 \text{ dB}$
<i>FIR</i> with $D = 20; L = 40$	$8.70 \pm 0.91 \text{ dB}$	$10.05 \pm 0.79 \text{ dB}$

As we expected, under time-invariant conditions the NC-IIR filter yields best performance since there is no approximation made in this filter form; the C-IIR filter produces the second best SNR, the reduction in the SNR being caused by the approximation of the anti-causal part of the NC-IIR with a finite time delay impulse response; the FIR provides the lowest SNR since both the causal and anticausal parts of the NC-IIR are approximated by delayed finite impulse responses. When the signal is nonstationary, the performance of the NC-IIR filter is still ranked first since its optimal form does not depend on the stationarity of the signal. However, its delayed approximate versions, the C-IIR and FIR depend on the stationarity of the signal. The longer the filter impulse responses are, the larger the influence of the nonstationarity. There exists an optimal filter length at which the improvement in SNR, obtained by increasing the filter length, is canceled by nonstationarity. Therefore in the given example, the FIR filter produced better performance in general than the C-IIR filter.

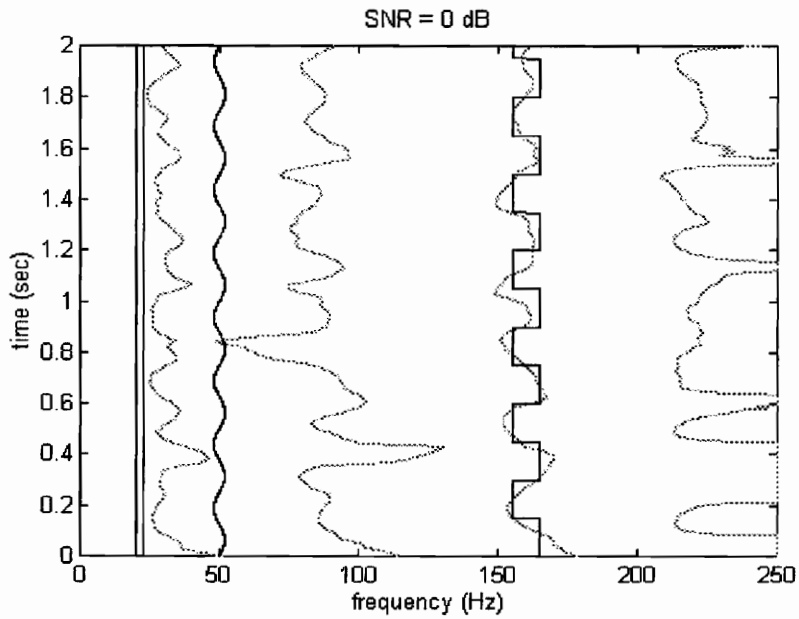


The standard deviations in SNR of the different optimal filters provide another characteristic of the optimal filter, i.e. the sensitivity of the filter performance to the match between the filter parameters and the signal parameters. The more optimal the filter is, the more sensitive it is to its parameters. Thus the NC-IIR filter is most sensitive to the match in parameters, and the causal IIR is the least sensitive among the three forms. When the signal parameters have to be estimated from noise contaminated observations, the sensitivity of the filters plays an important role in the performance. That is why the performance ranking among the three forms is reversed when the parameters of the optimal filters are obtained based on estimation.

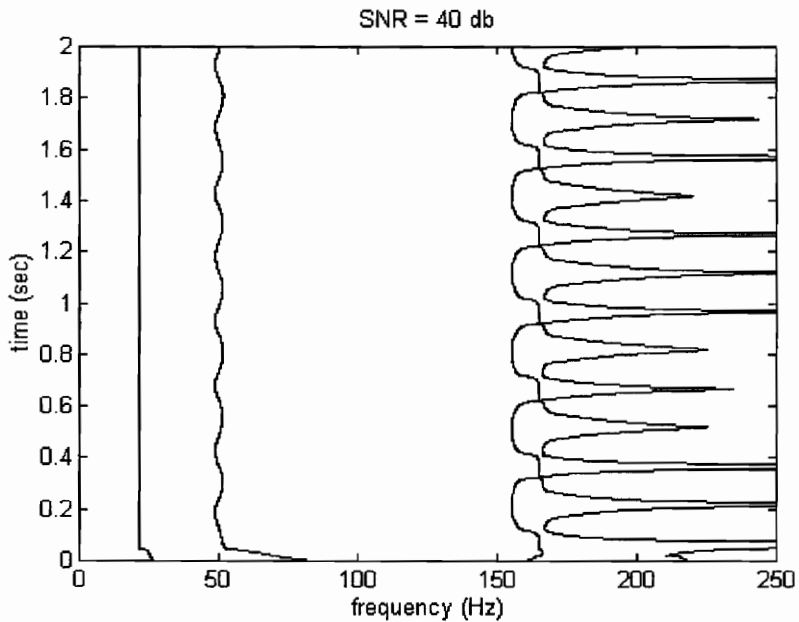
#### 4.5 Tracking noise contaminated nonstationary signals in the phase space

The second experiment is designed to test the performance in tracking of the time-varying spectral components of a noise contaminated signal. The test signal consists of four equal amplitude sinusoidal frequency components with differently time-varying patterns, which is an AR(8) process. We have  $f_1$  at 20 Hz,  $f_2$  at 22 Hz,  $f_3$  switching instantaneously between 155 and 165 Hz with a period of 0.3 second, and  $f_4$  changing sinusoidally between 48 and 52 Hz with a period of 0.25 second (See the solid lines in Figure 4.6a). Zero-mean white noise is added to achieve various SNR. The spectral estimates, based on the observed noisy signal, are then used to evaluate the performance of the associated Wiener filters. The Wiener filter updated from a conventional fixed resolution WRLS algorithm can not track the changes in the frequency components, even at high SNR (Figure 4.6a, b). The Wiener filter updated from MPSE shows a good match between the actual and estimated time-varying spectra (Figure 4.6c, d). The MPSE estimated spectrum follows the temporal changes for SNR as low as 0 dB (Figure 4.6c). The improved estimation process again translates to better performance for the MPSE-based optimal Wiener filter.

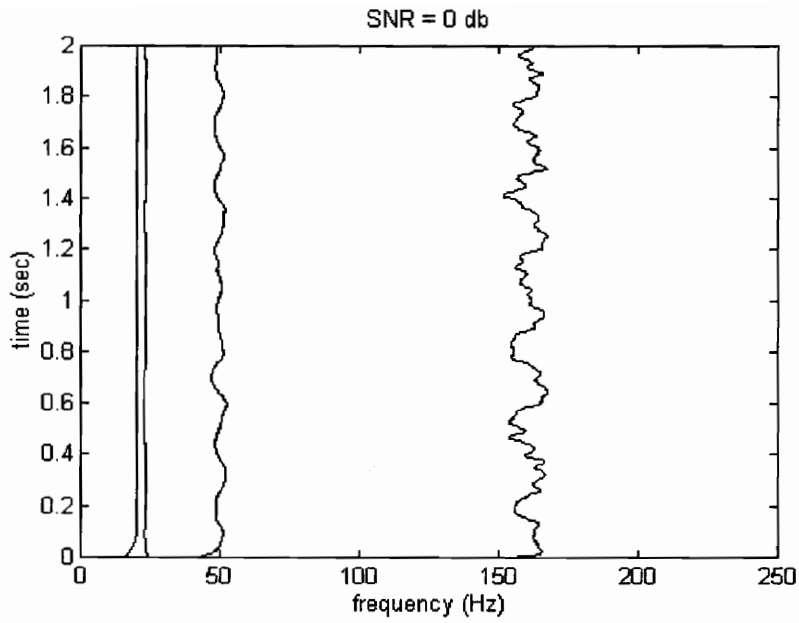
The given experiments show that the performance of the MPSE-based Wiener filter lies much closer to the ideally possible performance than for one based on conventional fixed resolution AR modeling. They also demonstrate that MPSE can outperform the conventional fixed resolution AR-PSE; not only in a high SNR environment, but also in a low SNR environment. That is, MPSE has better noise immunity than the conventional AR-PSE methods.



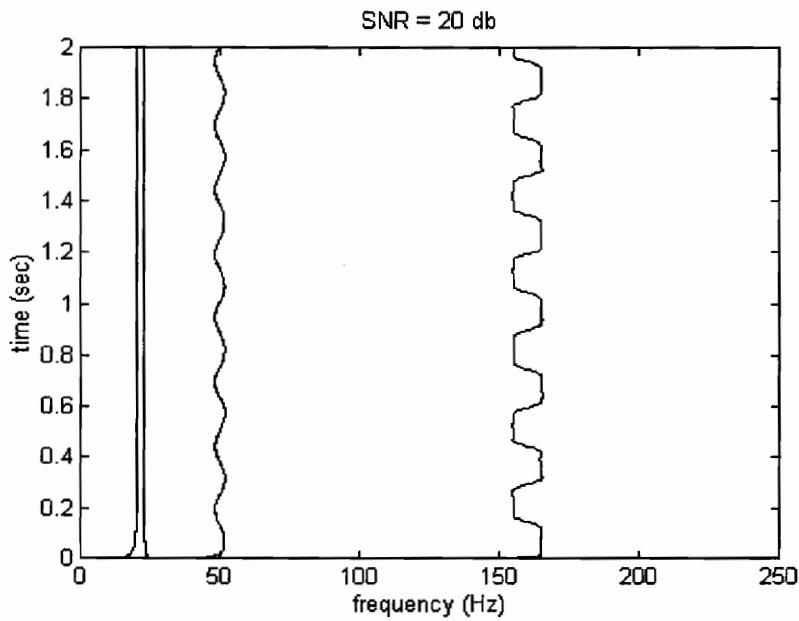
**Figure 4.6a** Spectral peaks of the signal (solid) and its estimate (dotted) using the fixed resolution WRLS based Wiener filter (SNR = 0 dB).



**Figure 4.6b** Spectral peaks of the signal estimated by the fixed resolution WRLS based Wiener (SNR = 40 dB).



**Figure 4.6c** Spectral peaks of the signal estimated by the MPSE-based Wiener filter (SNR = 0 dB).



**Figure 4.6d** Spectral peaks of the signal estimated by the MPSE-based Wiener filter (SNR = 20 dB).

## 4.6 Summary

The Wiener filter based on MPSE can zoom in a high frequency band, to catch fast "local" events, and zoom out in a low frequency band, to match the slow "global" changes. Compared with conventional fixed resolution parametric estimators, the MPSE method also offers better time resolution. The frequency band-splitting procedure used in MPSE can reduce the necessary model orders and improve signal-to-noise ratios, and thus reduce estimation errors. Three realizations of the optimal Wiener filter are given; for each the MPSE parameters are shown to relate directly to the Wiener filter coefficients, readily providing updates of the time-varying Wiener filter. Experiments show that the performance of the MPSE-based Wiener filter lies much closer to the ideally possible performance than for one based on the usual AR modeling.

The MPSE Wiener filtering can be performed either independently in each octave band, when recovering the signal components is of major interest, or in the entire frequency band, to prevent the waveform distortion caused by different group delays. Precautions need to be taken when using the noise compensation method. Over-estimation of the background noise level can cause over-smoothing of the signal, or even produce a negative PSD, which can bring about serious distortion of the waveform.

## 5. TIME-FREQUENCY ANALYSIS OF EEG SIGNALS

### 5.1 Introduction

The Electroencephalograph (EEG) of the brain reflects, over time, a dynamic interplay between excitatory and inhibitory processes. Specific frequency bands that can be associated with different states of arousal, ranging from coma to heightened attention, have been identified from the study of the Fourier spectrum of the EEG [120]. Recently more attention has been given to specific oscillatory activity generated either cortically or subcortically as indicators of ongoing reciprocal interactions between excitatory and inhibitory mechanisms either within neurons or by neuronal networks, within or across brain structures and regions. For example, lower frequencies ( $f < 4$  Hz) may involve the entire neocortex and are associated with lowered arousal within sleep or pathological states [121], while higher frequencies appear to originate from localized assemblies of neurons in specific structures and are associated with cognitive processes that change rapidly [122]. However, much less work has examined the fluctuating waxing and waning of the activity, in terms of both frequency and amplitude, over real time periods. Almost no work has examined the interrelationships between oscillatory activity between different frequency bands at the same recording site or across sites, let alone across time.

The EEG signals or brain waves are characterized as nonstationary stochastic signals. Unlike the electrocardiograph (ECG), there is no particular wave form that can be associated precisely with the EEG. Only the oscillation rhythm and its amplitude are considered to carry information. Thus, the estimation of the time-varying spectrum of the EEG is critical in studying brain waves and the associated brain functions. The oscillation rhythms of the EEG are commonly defined into different waves, such as  $\delta$ ,  $\theta$ ,  $\alpha$ ,  $\beta$ , and  $\gamma$  waves (see the second column of Table 5.1) or it can be defined into finer frequency bands (as listed in the third column of Table 5.1). Still, the frequency resolution corresponding to the bandwidth of each of the above rhythms is coarse.

Defining the EEG signal components according to their oscillation frequency may potentially hide the time-varying characteristics of the frequencies of these signal components; that is not only

the magnitude of the signal component is time-varying but also is its frequency. When the frequency of a brain wave varies continuously from one defined frequency band into another, e.g. from the  $\alpha$  frequency band into the  $\beta$  frequency band, one can not distinguish whether an old  $\alpha$  wave vanishes and a new  $\beta$  wave emerges or a single signal component varies its frequency from a low frequency to a high frequency. On the other hand, when monitoring the EEG in a certain frequency band with a bandpass filter, the change in intensity of the output of the filter can not distinguish the causes of the change, that is, the change in intensity may be caused by a change in the magnitude of the signal component, and/or caused by the change of the frequency of the signal component when the frequency shifts in and out of the filter passband.

**Table 5.1 Definitions of different EEG waves.**

rhythmic waves	( in Hz)	(in Hz)
$\delta$	0.5-3.5	0.5-3.5
$\theta$	3.5-7.5	low $\theta$ (3.5-5.5) & high $\theta$ (5.5-7.5)
$\alpha$	7.5-13	low $\alpha$ (7.5-9.5), middle $\alpha$ (9.5-11.5), high $\alpha$ (11.5-13.5)
$\beta$	13-38	low $\beta$ (13.5-20), high $\beta$ (20-38),
$\gamma$	38-42	38-42

Although EEG signals are nonstationary, their time-varying characteristics have most of the time been neglected in data analysis. Due to limited technology, most studies of scalp-recorded oscillatory activity in the time domain and in the frequency domain are two separate processes. The analysis in the time domain is mainly focused on statistical waveform measurements and waveform decomposition, such as in principal component analysis, by which the spectral information in the signal components is lost during data processing, which then frequently results in statistical inconsistency. The analysis in the frequency domain is mostly done by employing Fourier transform based methods. The result is a time-invariant spectrum which only provides an average,

over the processed window period, of the activities in the frequency band. There is no temporal information left in the Fourier spectrum of the brain wave. For instance, when a broad band spectrum is produced by Fourier analysis, one can not tell whether the broad band spectrum is generated by a narrowband time-varying frequency component or whether it is a sign of the coexistence of multi-frequency components.

Therefore, in the rapidly expanding field of research involving EEG signal processing, there is a need for new techniques to analyze the spectral and temporal aspects of dynamic brain activity in the time-frequency space, since it is a nonstationary process that has a time varying spectrum. The Multiresolution Parametric Spectral Estimation (MPSE) method is proposed as a new approach for time-frequency analysis of brain waves. The goal is to provide not only the temporal information (waveform) or spectral information (Fourier spectrum), but the combined information in the time-frequency phase space.

## **5.2 Time-frequency analysis of EEGs**

Time-frequency analysis of signals provides a decomposition into frequency components as functions of time. This allows one to see whether a particular frequency component exists, as well as its time-varying patterns. Such time-dependent behavior can then be correlated with timed events, such as evoked potentials, and thus produce new avenues for analysis and interpretation of neurological signals. Another feature of accurate time-frequency analysis is that it can show the simultaneous - in time - occurrence of different frequency components, especially when some are weak and others are strong. Again this provides additional information for analysis and interpretation. In addition, the time-varying patterns of each spectral component in both magnitude and frequency, together with their geometric distribution on the scalp, are important for identifying specific brain activities and for distinguishing noise from brain waves.

Most recent emphasis in time-frequency analysis methods has concentrated on the wavelet transform, a Fourier transform based method. While the wavelet transform produces a time-frequency analysis of a signal, there are limitations to the accuracy with which time and frequency can be determined. As studied in Chapters 1 and 3, the product of the time and frequency resolution is limited by the Heisenberg uncertainty principle; this is the well-known fundamental

limit of frequency resolution for any Fourier transform based method using a finite data sample. The frequency resolution is inherent in the procedure because a finite window of data is analyzed for its frequency content as the window moves in time. As the window is short, time localization is more accurate, while frequency localization is less accurate. As the window is made longer, time localization worsens and frequency resolution gets better; the latter assumes however that the behavior in frequency is stationary over the analysis window. Since the EEG signals are time-varying and some of its frequency components are short lasting, with various temporal patterns, both time and frequency resolution are essential for catching these phenomena. The resolution limit for Fourier transform based methods is not desirable in the processing of EEG signals.

In practice, due to computational complexity considerations, the popularly used wavelet transforms are discrete wavelet transforms whose basis functions are in the class of orthonormal bases of compactly supported wavelets [55]. With the multiplier  $a = 2$ , the daughter wavelets of the discrete wavelet transform act as FIR octave bandpass filters (for details see Chapter 6). These filters continuously estimate the signal energy within each octave band, such that the signal is decomposed into multi-octave band components. Since the shape of the signal components is determined by the associated daughter wavelets in the corresponding octave bands, the discrete wavelet transform requires the nonzero signal components to be located at the centers of the octave bands with the same symmetric shape as the spectrum of the daughter wavelets. Otherwise, the decomposition can suffer from serious information loss or distortion in later reconstruction, The requirement of octave distributed spectra is hardly acceptable in processing EEG signal.

In modern spectral estimation methods, the data are used to develop a model for the signal under analysis, and the spectral characteristics ascribed to the data are in fact the spectral characteristics of the model. Under the assumption of stationarity the modeled spectral characteristics are not subject to the finite analysis window, and the frequency resolution can exceed the resolution lower bound associated with Fourier analysis. This is not to say that there are no limitations on the frequency resolution, for now the accuracy of the spectral characterization depends on the model parameters, which in turn depends on the model choice and how well one can estimate its parameters from the available data. The most popular of the modern spectral estimation methods are based on the autoregressive (AR) model; this is undoubtedly due to the fact that efficient algorithms exist for AR parameter estimation. Under high signal-to-noise ratios, the AR modeling



procedures can outperform Fourier-based methods in terms of frequency resolution, especially for short data windows [65, 66]. The problems associated with AR-PSE are that the EEG signals are usually multifrequency components, which results in a high order AR model being needed. To estimate the high order AR parameters, a large time window is required to produce statistically consistent results, which yields low time resolution. The other problem of AR-PSE is its sensitivity to signal-to-noise ratio. These problems have limited its use in processing EEGs.

To use the superior frequency estimation characteristics of parametric estimation techniques, and also have the flexibility of changing the resolution scale in time-frequency space, we propose to process EEGs with our MPSE. As demonstrated in Chapters 3 and 4, MPSE has excellent frequency and time tracking capabilities, and outperforms the Morlet wavelet transform. It also shows better noise immunity than the fixed resolution parametric spectral estimation methods.

There are two different kinds of EEG records. One is the *running EEG* which is collected continuously from the subject under a design condition, such as sleeping, counting, or sitting relaxed. The other is the *evoked potential*. During data collection, a stimulus is repeatedly presented to the subject, which could be visual, auditory, or tactile. Finally, an average brain wave response to the stimuli is composed by summing all the trial responses such that the noise and brain waves activities which are uncorrelated with the stimuli cancel each other out. The results are the averaged brain responses to the stimuli. The summation operation enhances the signal to noise ratio by about  $\sqrt{\text{total - number - of - trials}}$  times.

The difference between the two EEG records, from a signal processing point of view, is that the running EEG consists usually of relatively long data records (30 seconds or more, or more than 6,000 samples for each channel). As the brain is a multi-tasking processor, brain waves are a superposition of many different components associated with different brain activities. The nonstationarity of the running EEG is stronger than for the evoked potential. The record length for evoked potentials is usually short (about 1 to 2 seconds, or less than 500 samples for each channel). After averaging, the signal-to-noise ratio and nonstationarity are improved. Meanwhile, the high frequency components in the evoked potential have also been reduced significantly by the averaging process. The signal energy is thus concentrated in the lower frequency bands.

The high resolution and tracking capability of MPSE is essential for analyzing evoked potentials as they last less than a second. It requires high time and frequency resolutions to follow the changes of the evoked brain activities from before stimulus to after stimulus. As mentioned earlier, the conventional Fourier transform suffers from fixed frequency and time resolutions, while discrete time wavelet transforms require the spectrum of the signal to satisfy octave distribution. Both the Fourier transform and the discrete wavelet transform have poor resolution as well. By contrast, MPSE has been shown to overcome these major difficulties and to have excellent frequency and time tracking capabilities in computer simulations [99, 100].

### **5.3 Experimental results**

To demonstrate the performance of MPSE in EEG data analysis, two different kinds of EEG records were used<sup>1</sup>. The first EEG signal is a three second running EEG from a subject with high sustained attentional abilities, obtained while listening closely to an information auditory passage with eyes closed. The second kind of EEG data are evoked potentials collected from a half second before the stimulus until one and a half seconds thereafter. The stimulus is an electrical 150 Volt pulse of 0.5 msec duration, which is applied to one of the subject's fingers. During this recording, the subject was under hypnosis and instructed to either pay attention to the stimuli or to not pay attention to them. The EEG data were recorded using a 19 electrode placement system (Figure 5.1) [123]. All channels were simultaneously recorded at a sampling rate of 256 samples/second for the running EEG and 200 samples/second for the evoked potentials. All EEG signals were prefiltered by a bandpass filter (built into the amplifiers) from 0.3 to 50 Hz. In the experiment, the Windowed Recursive Least Squares (WRLS) algorithm was used for the AR spectrum estimation, and the sliding window width used was 20 samples after decimation. The orders of the AR estimators were 2, 4, and 6 for octave bands 2 to 4 respectively.

---

The EEG data were provided by Dr. Helen Crawford from the Psychology department at Virginia Tech, and by Dr. Karl Pribram from the Center for Brain Research and Information Sciences at Radford University.

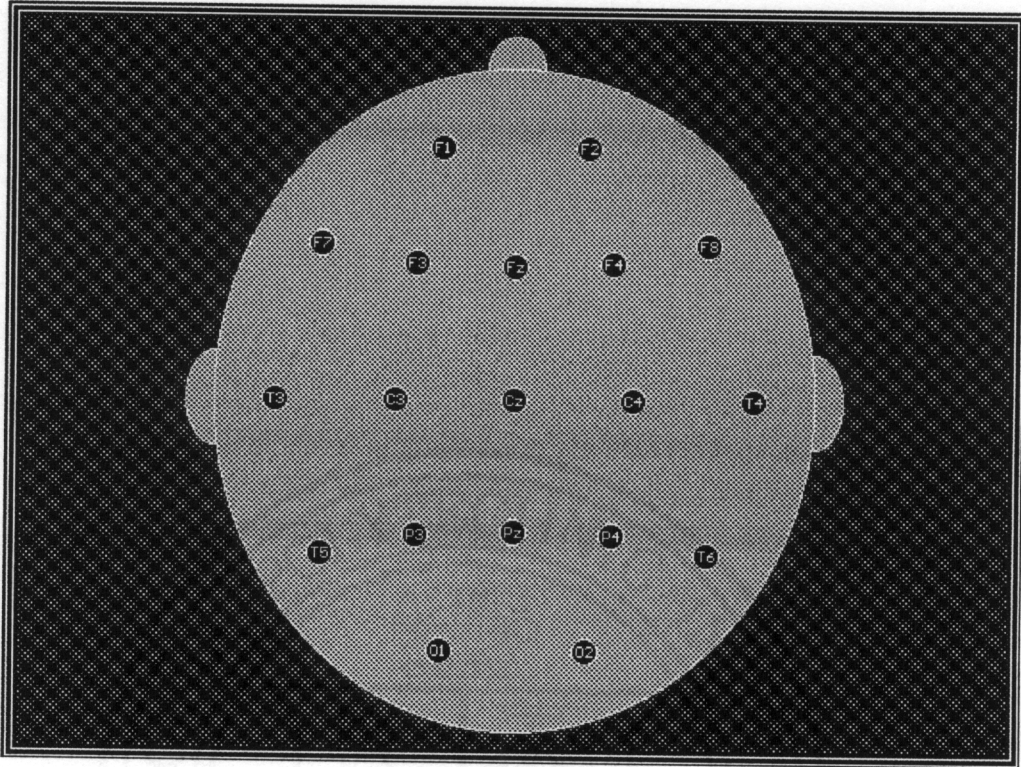


Figure 5.1 Nineteen electrode EEG system.

### 5.3.1 Examples of time-frequency analysis of running EEG

For processing the running EEG, MPSE was designed to split the frequency band into four octave bands. Using asymmetric designs of the band-splitting filters, the first octave band is from 56.32 to 128 Hz; the second octave band is from 28.16 to 56.32 Hz; the third octave band is from 14.08 to 28.16 Hz; and the last one is from 0 to 14.08 Hz. Decimation was applied in each band-splitting operation except the last one. That is, the sampling rate in the first octave band is 256 samples/second and 128 samples/second in the second octave band. The third and fourth octave bands shared the same sub-sampling rate of 64 samples/second. Since the frequency components of EEG signals are usually below 50 Hz (Table 5.1), the first octave band is considered an 'empty signal band' and its output was not used in this study. Figure 5.2 gives the original three second EEG records. For later comparison, a rectangular windowed Fourier transform was applied to the data. The FFT-spectrums in Figure 5.3 show frequency components averaged over the three second interval (768 samples). No time history of these components can be provided in such FFT results.

The observer can not tell whether these frequency components exist simultaneously or if they change from one frequency to another, or if there is any on/off pattern. The MPSE results (Figure 5.4) however, not only show the existing frequencies, but also provide the time history of each frequency component. *“Due to the non-overlapping of frequencies one can see clearly the presence of several dominant individual frequency bands that are associated with traditionally identified rhythmic oscillations within the 0.3 to 56.32 Hz range. Differences in the anterior and posterior regions of the brain support the view that there are two major attentional regions that have differential EEG patterns [124-126].”* [127].

To further explore the performance of the MPSE based time-frequency analysis with the spectrogram, the Morlet wavelet transform based scalogram and the conventional fixed resolution AR-PSE, we used the EEG data in channels F4 and P3 as examples. The Fourier transform based spectrogram was computed by using a 128 sample FFT with a Hamming sliding window (sliding one sample at a time). The Morlet wavelet transform was designed with 10 significant cycles in its Gaussian window. The fixed resolution AR-PSE is computed by an 8th order WRLS algorithm with a sliding window of 50 samples. Since the signal powers in these two channels are mainly located below 30 Hz, we only provide the spectral analysis results from 0.3 to 28.2Hz, i.e. the spectrum in the third and fourth octave bands.

In the first example (Channel F4), a broad band frequency component between 14 to 24 Hz (high  $\beta$ ) in channel F4 is observed in the FFT-spectrum (Figure 5.5c), while the time-varying MPSE-spectrum shows that this broad-band FFT-spectrum is formed by the time average of a single time-varying frequency component (Figure 5.5b). The narrow-band frequency component starts at  $t = 1$  second at  $f = 20$  Hz, increases to 23Hz, and then continuously declines to  $f = 17$  Hz at  $t = 2.2$  seconds. The amplitude of this frequency component also varies during that period of time. The time-varying pattern of this frequency component can be qualitatively identified from the plot of raw data (Figure 5.5a). The second frequency component is at  $f = 11.5$ -12 Hz (low  $\beta$ ). This frequency component clearly shows on-off patterns. Since the on-time of the component is short, the FFT-spectrum does not show a high spectral peak at the corresponding frequency band. However, there is a high spectral peak near 1 to 2 Hz in the FFT-spectrum, while the MPSE analysis result shows that this high spectral peak is contributed by a low-energy long-lasting broad-band signal component from  $t = 0$  to 2.2 seconds. Comparing the MPSE-spectrum with the raw

EEG data and the FFT-spectrum, the time-varying spectral pattern of these three frequency components can not be recognized by using either the waveform analysis in the time domain or the static spectrum analysis in the frequency domain.

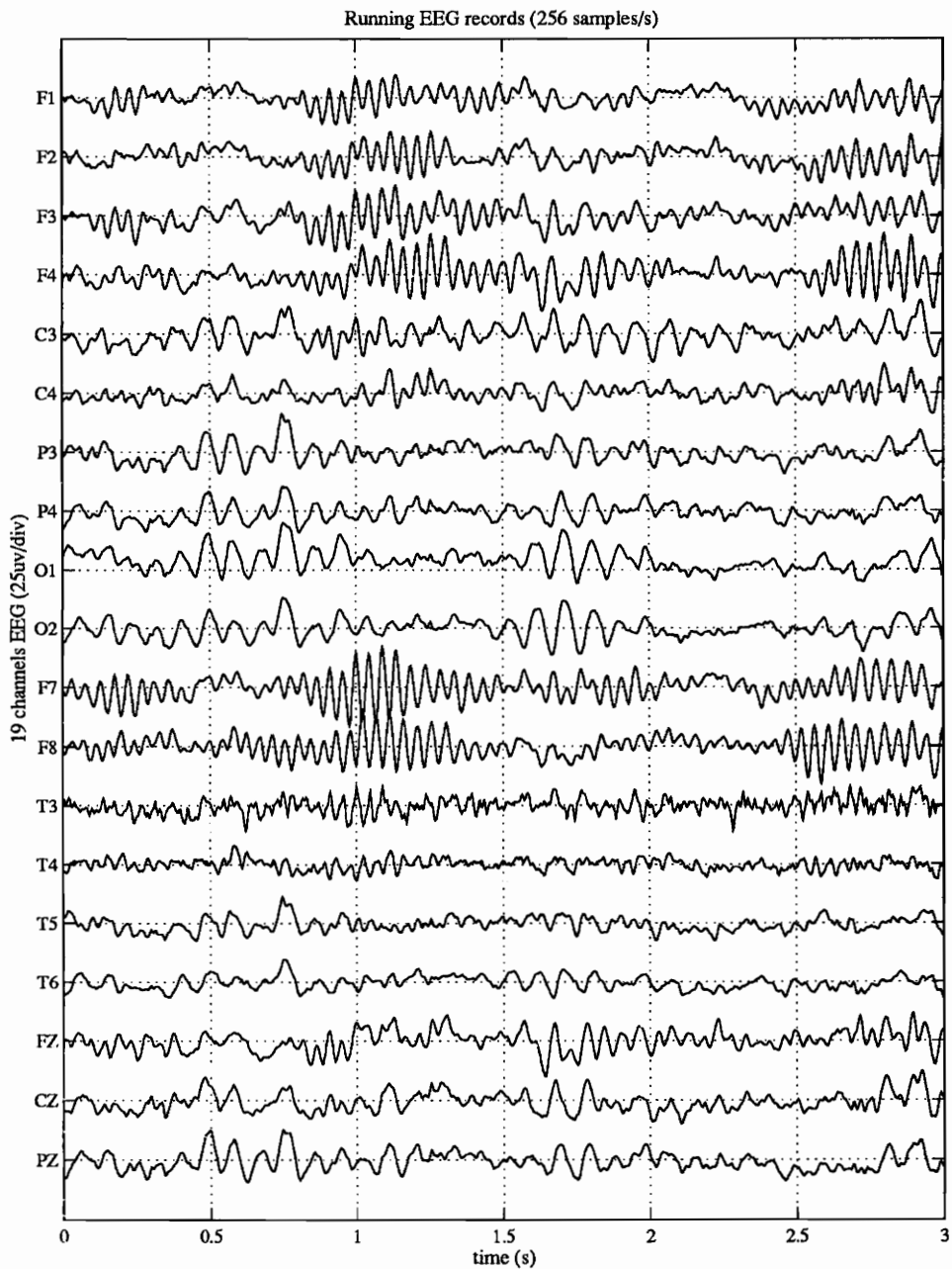


Figure 5.2 Three second raw EEG records.

The second example analyzes the EEG in Channel P3 (Figure 5.6). The MPSE-spectrum (Figure 5.6b) shows that there is a frequency component at  $f = 11$  Hz which changes across time, with its amplitude waxing and waning. There are on-off patterns which have also been observed in the low frequency band (0.5 to 5Hz). Although, the FFT-spectrum (Figure 5.6c) shows no significant spectral peak in the frequency band from 14 to 24 Hz, the MPSE-spectrum shows that there is a low power signal component located in that frequency region. Moreover, the on-off pattern matches that in the first spectral component in Channel F4, while its time-varying pattern has some phase changes. This may suggest that the high  $\beta$  component in P3 is the same EEG signal as in F4. The weakness in power and the changes in phase may have been caused by propagation from F4 to P3. If this hypothesis can be proved in future research, the distribution of individual signal components can be used to localize the signal generator, which in turn helps the neural scientist to understand the brain's function and brain organization.

Figure 5.7a,b show the Fourier transform based spectrogram and the scalogram of the Morlet wavelet transform for F4. The first signal component located in 14 to 24 Hz is barely seen, while the other components have been averaged out due to the low resolution. Both the spectrogram and the scalogram show interference patterns in their results. Since the time-window of the Fourier transform is 128 samples (half second), the spectrogram is 2.5 seconds when the time window slides from the beginning of the 3 second data to the end. In the wavelet transform, since the window width is inversely proportional to frequency, we could only analyze to a frequency as low as 5 Hz to generate 2.5 seconds of scalogram. The long transient in the wavelet based scalogram makes it impossible for the wavelet to analyze short lasting events or short data records. The MPSE-spectrum, however, has much shorter transients than the Morlet wavelet transform. The Fourier transform based spectrogram (Figure 5.7c) and the Morlet wavelet based scalogram (Figure 5.7d) for P3, again show poor time and frequency resolutions.

The fixed resolution AR-PSE for F4 is provided in Figure 5.8a. A strong interference (cross-talk) between the frequency components is obviously visible in the result. Increasing the window width of the AR-PSE, the cross-talk can be reduced at the cost of over-smoothing in time, which means a poor time resolution. For P3, the AR-PSE (in Figure 5.8b) shows very strong interference between the frequency components, to the point that they are no longer distinguishable.

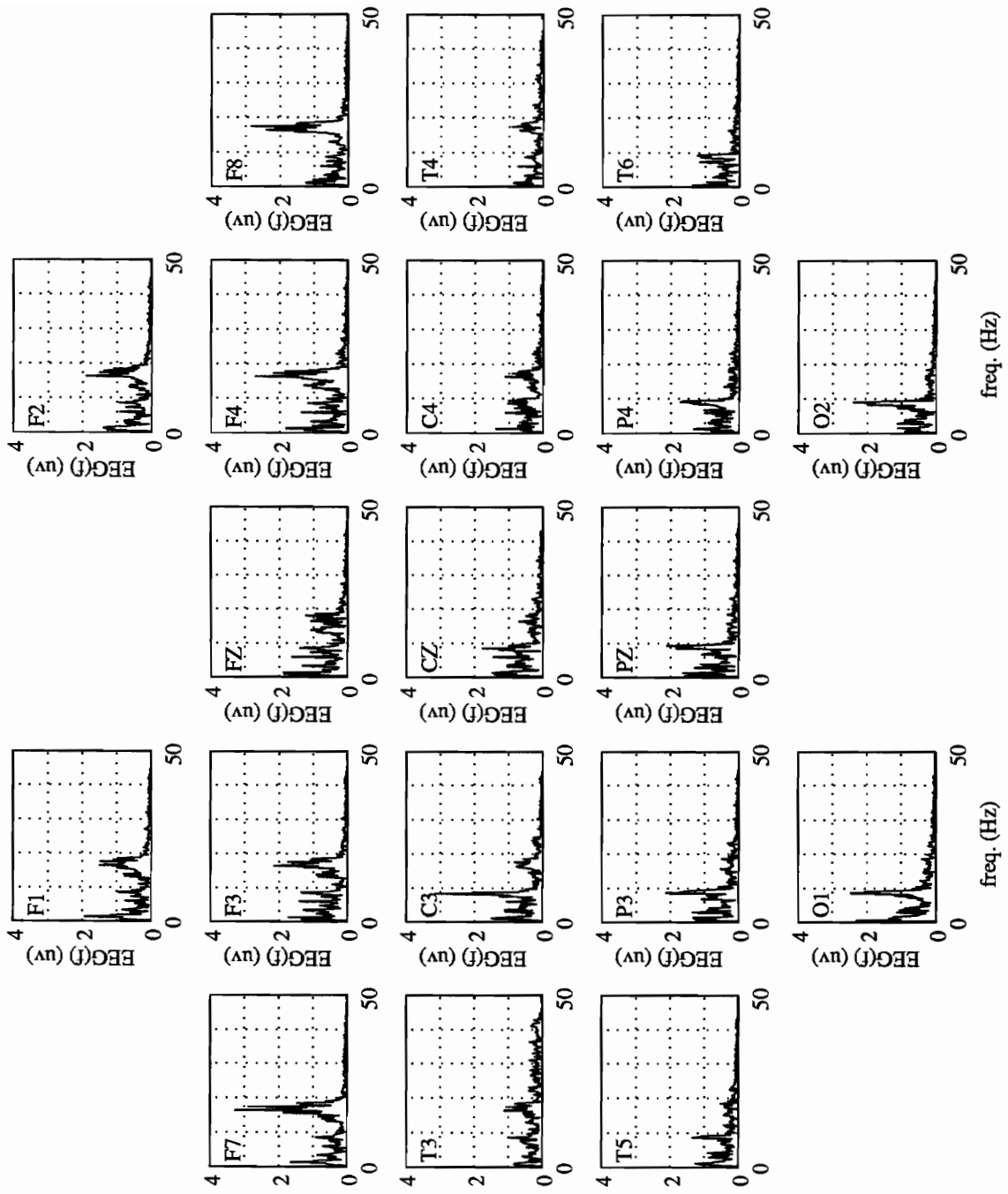


Figure 5.3 EEG Fourier Spectra.



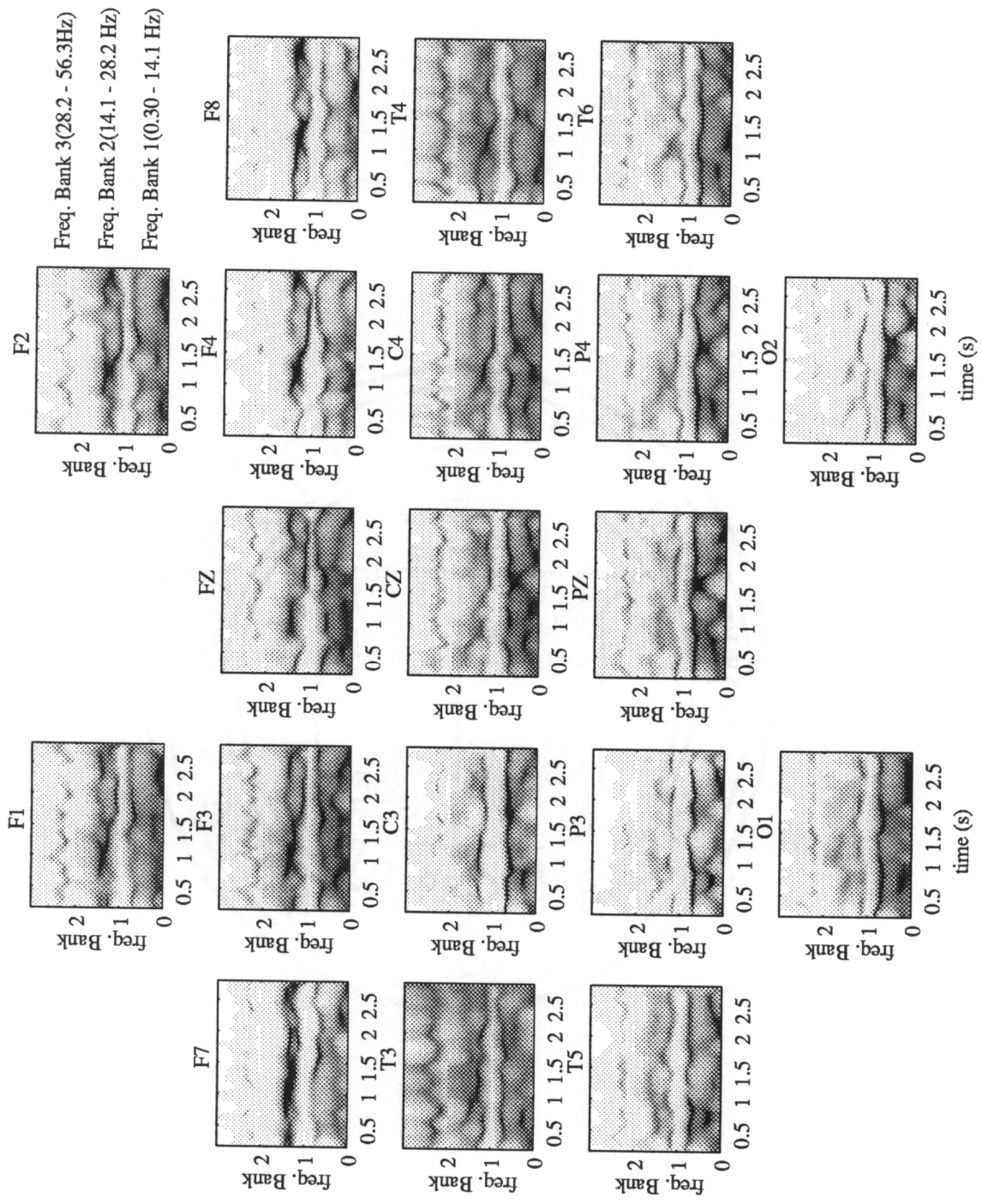
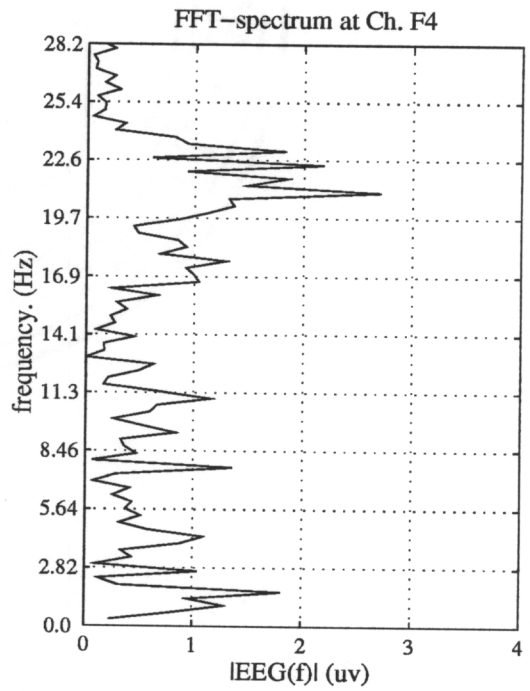
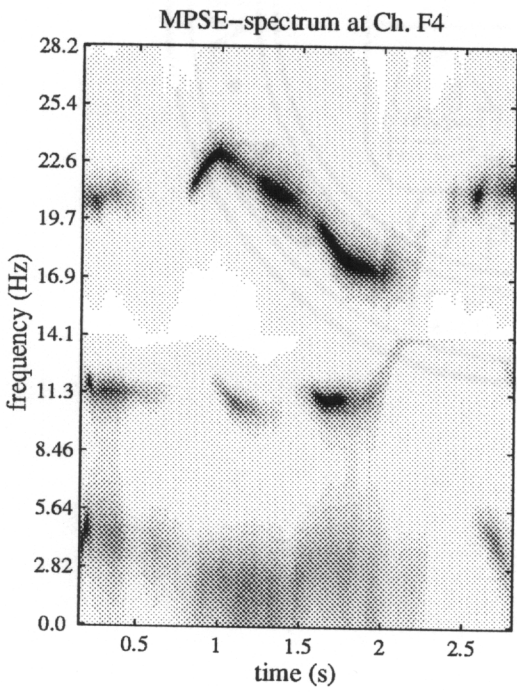
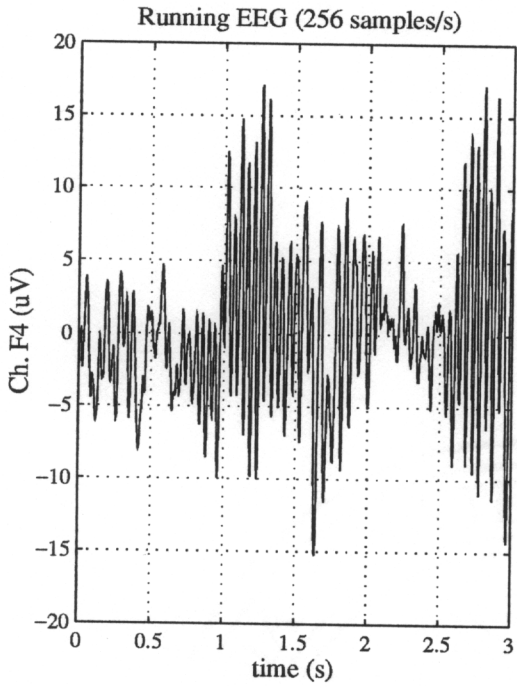


Figure 5.4 MPSE-based time-varying spectra.





**Figure 5.5** Running EEG FFT-spectrum and MPSE-spectrum for channel F4.

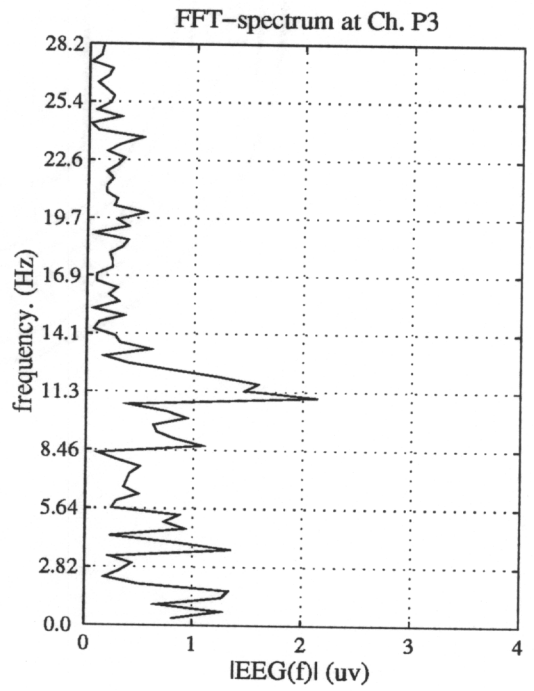
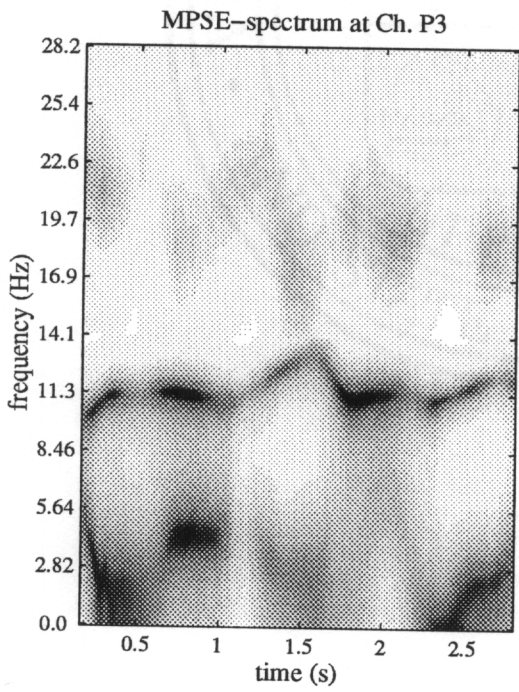
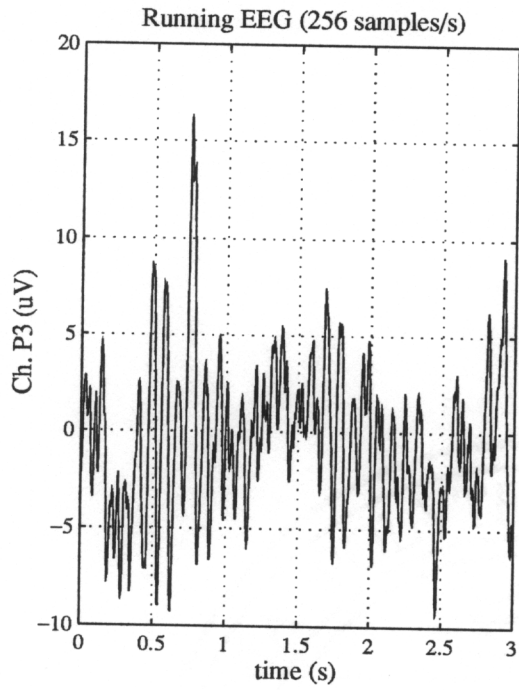
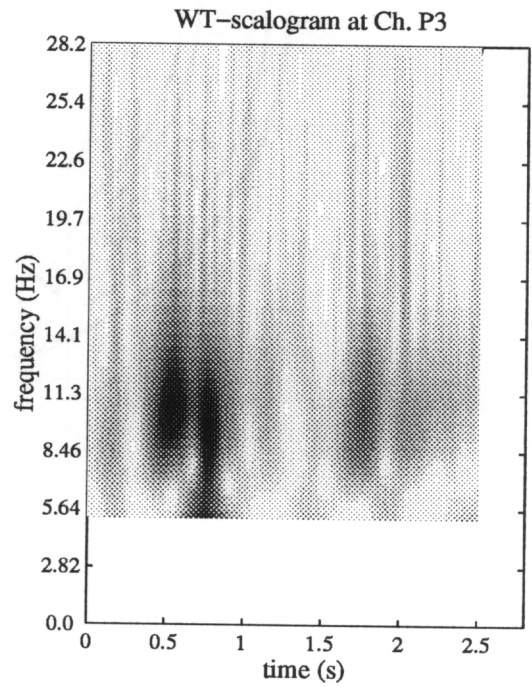
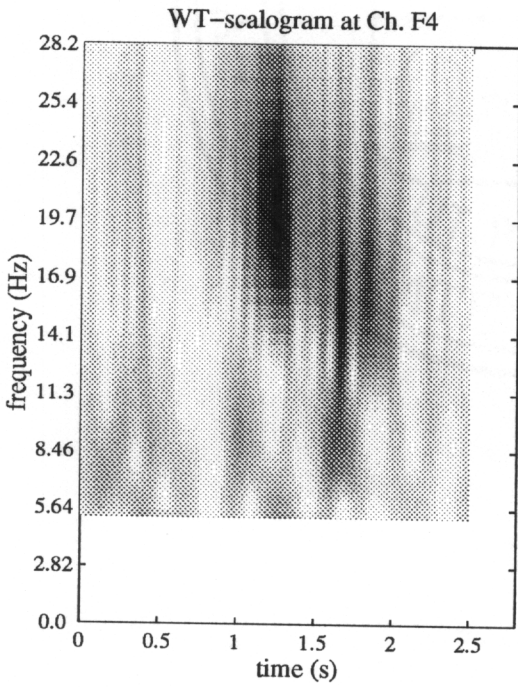
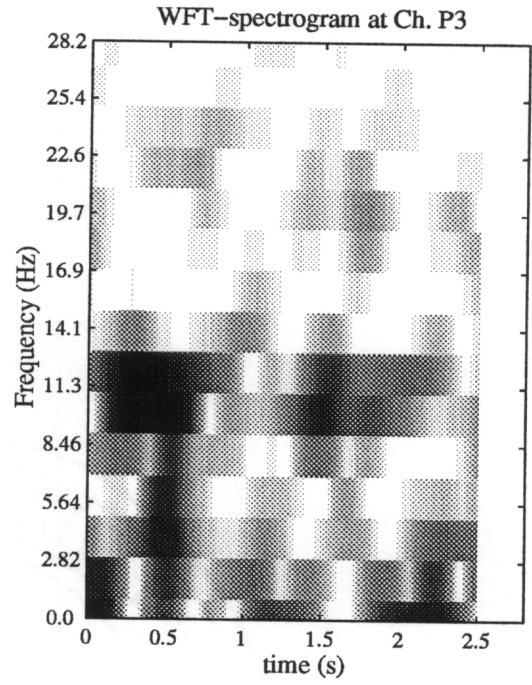
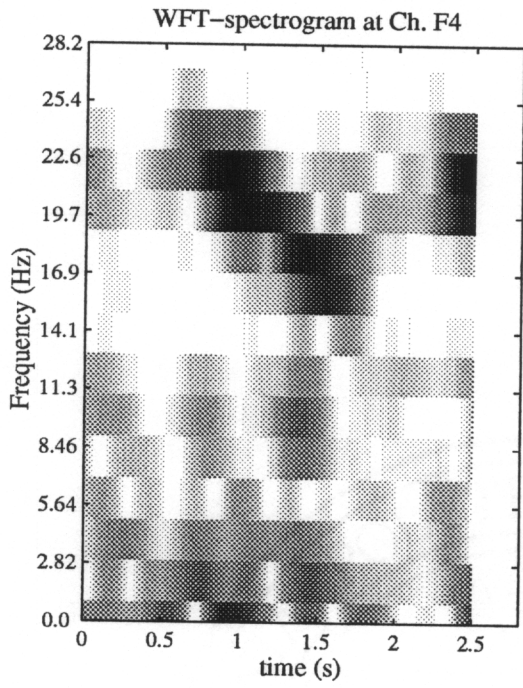


Figure 5.6 Running EEG FFT-spectrum and MPSE-spectrum for channel P3.



**Figure 5.7 Spectrograms (top row) and scalograms (bottom row) for channels F4 and P3.**

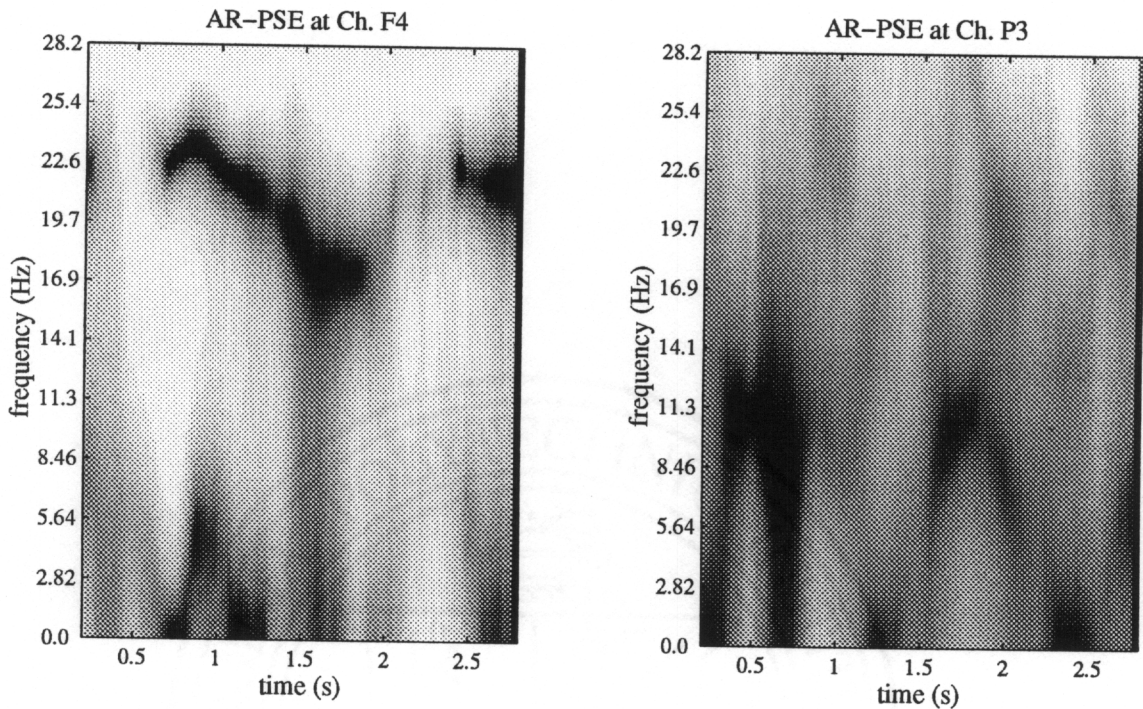


Figure 5.8a,b Fixed resolution AR-PSE for Channels F4 and P3.

### 5.3.2 Examples of time-frequency analysis of evoked potentials

The raw evoked potential data are provided in Figure 5.9. The MPSE method splits the entire frequency band (0.3 to 100 Hz) into 4 octave bands. The first octave band from 50 to 100 Hz was empty, and thus not used in processing. The MPSE results are shown in Figure 5.10. Again, the MPSE method demonstrates the dynamic nature of the EEG signal components: frequency changes across time, amplitude waxing and waning, differential topographical spatial distributions, and possible covariances between their presence or absence with one another. To quote my mentors [127]: *“Unlike [128], we observed a continuing of the “40-Hz” oscillation after the stimulus for a longer length of time than they reported, but with a change in frequency.” “There is a very visible “40 Hz” oscillation, particularly in T3, T4, F7, and F8, that fluctuates for a much longer*

*period than the 100 msec noted by Tiitinen for an auditory ERP [128]. Around 13 Hz there are two oscillations that are differentially distributed across the scalp and show frequency shifts pre- and post-stimulus. Finally, there is a strong theta oscillation that appears immediately after the stimulus and lasts only several hundred msec [128]."*

High resolution is not the only advantage for MPSE. Frequency band-splitting in MPSE improves the signal-to-noise ratio in each octave band. In analyzing the EEG signal, one of the difficulties is to classify or separate muscle activity (Electromyogram or EMG) from the EEG, especially in the higher frequency bands (above 30 Hz). To distinguish the EMG from the  $\beta$  or  $\gamma$  waves of the EEG is difficult because they occupy the same frequency band. The time-varying MPSE spectrum offers a new way to distinguish brain waves from EMG activity. The EMG is usually broadly distributed across a subject's scalp, while brain waves are relatively localized. The time-varying pattern of the high frequency components in all the channels can provide information on how broadly the same pattern is distributed across the scalp. In general, if the patterns are similar in all the channels, they are likely EMG, otherwise they are more likely to be brain wave activity.

The basic patterns of the time-varying spectrum can be summarized as:

- drifting pattern (e.g. channel F4 in Figure 5.5b);
- on/off pattern (e.g. channel F4 in Figure 5.5b and channel FZ in Figure 5.4);
- merging or bifurcating pattern (e.g. F2 in Figure 5.4);
- amplitude waxing and waning pattern (e.g. P3 in Figure 5.6b).

These studies provide preliminary verification of the MPSE method in terms of:

- (1) its superiority to other methods, such as the wavelet transform, and
- (2) its ability to provide further dynamic information about brain wave activity not available from traditional Fourier transform based methods.

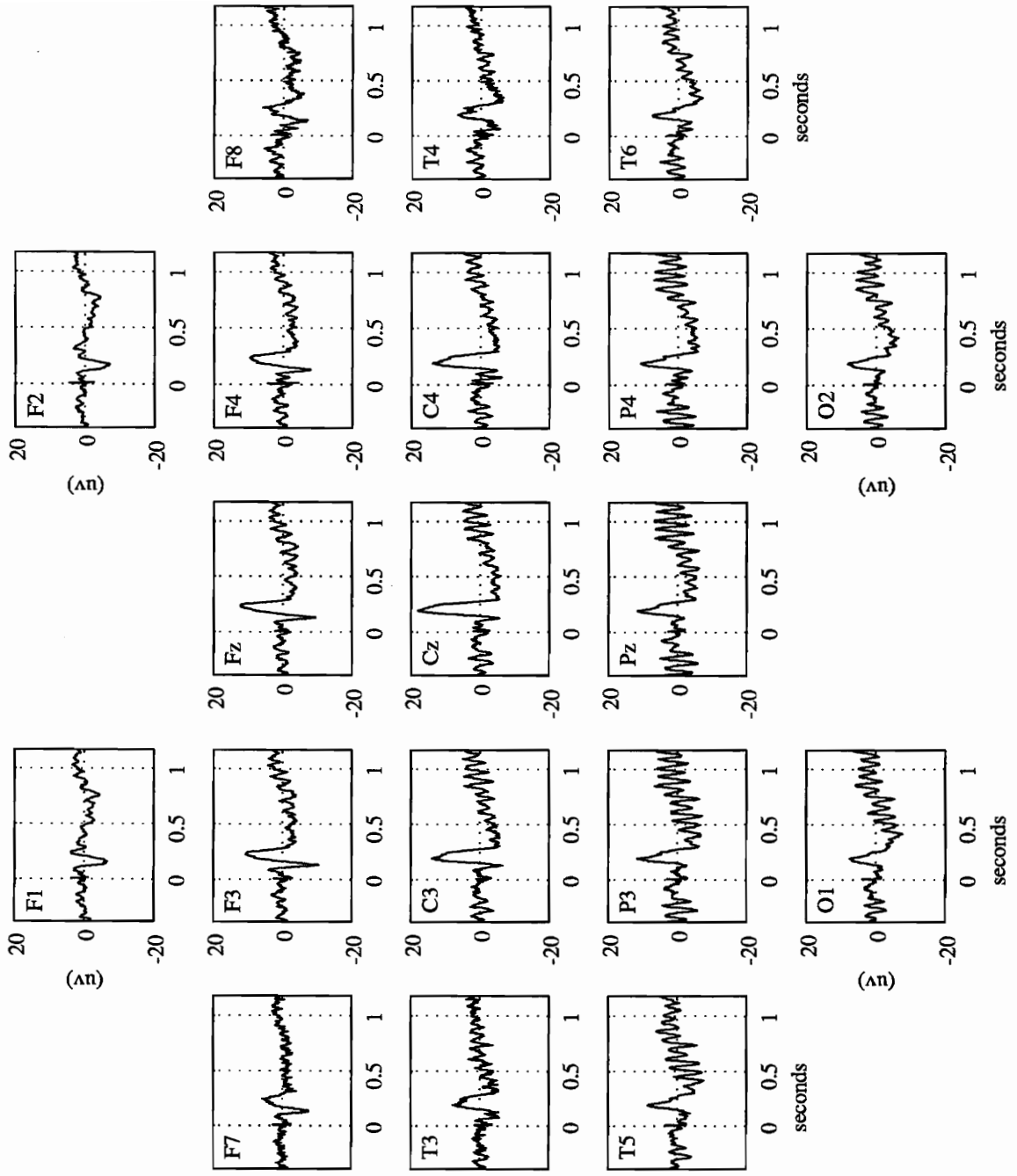


Figure 5.9 Example of Evoked Potentials.

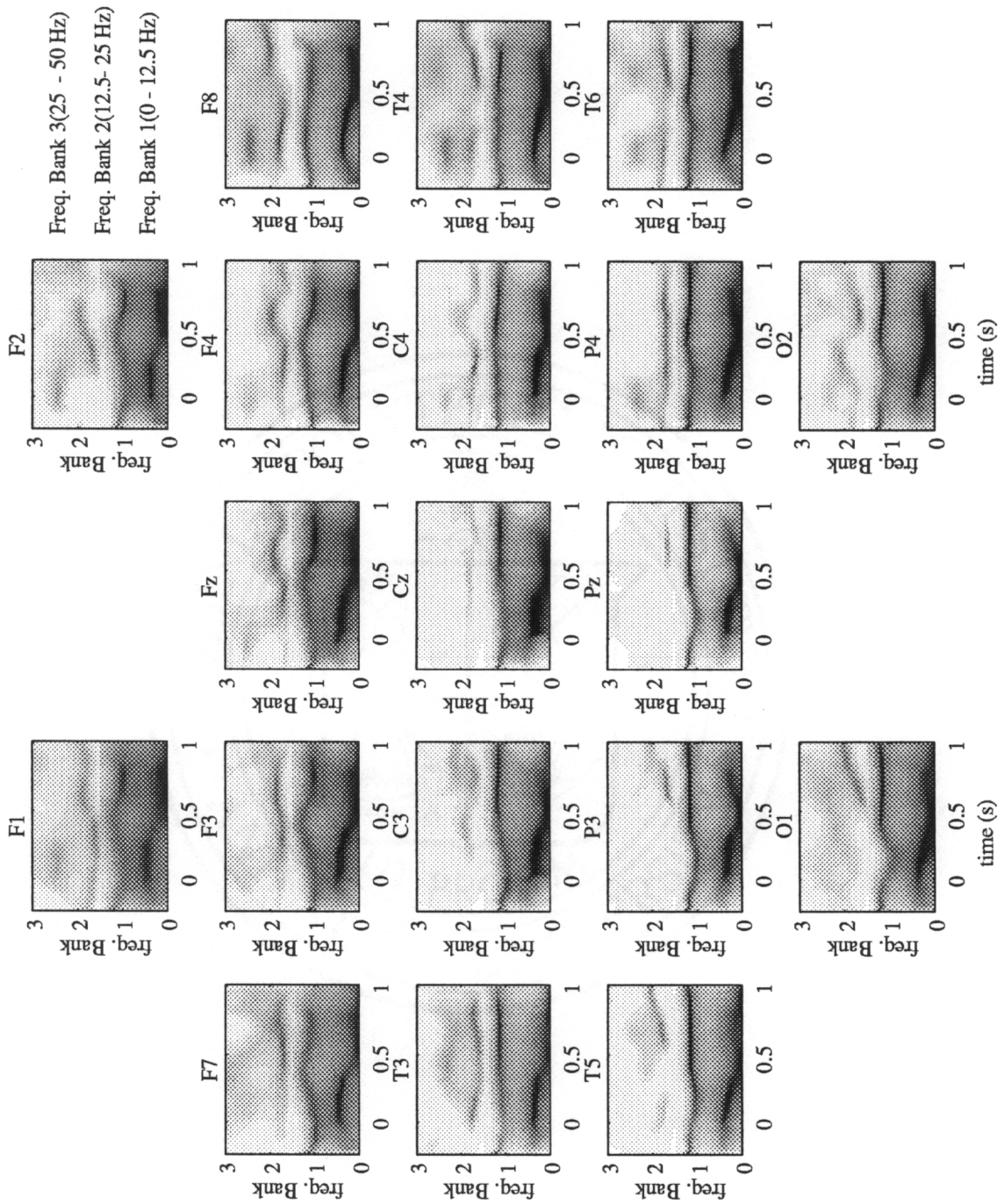


Figure 5.10 MPSE Results for Evoked Potentials.



## 5.4 Summary

The advantages of MPSE over Fourier transform spectral analysis, wavelet transform analysis and fixed resolution AR-PSE in the processing of EEG signals can be summarized as follows:

- Compared with the conventional method of EEG signal processing (waveform analysis, principal component decomposition, and static FFT based spectral analysis), MPSE demonstrated its superior performance in tracking and retrieving of the signal components in time-frequency space, such that the temporal and spectral information can be studied and visualized simultaneously.
- Compared with the Fourier transform based spectrogram, MPSE demonstrated higher time and frequency resolutions, and multiresolution capability.
- Compared with the wavelet transform, i.e. the scalogram, MPSE demonstrated better time and frequency resolutions, and a shorter transient period.
- Compared with the fixed resolution AR-PSE, MPSE overcomes the interference between the frequency components and the flexibility of choosing different window widths in different frequency bands.

Although the results provided in this chapter reflect an early stage of the research in processing and interpretation of EEG signals, the pilot data have already shown the great potential of MPSE in the processing of EEGs. More studies need to be performed in order to relate the time-varying patterns of the EEG signal to experimental conditions and the behavior of the subject. We expect MPSE to be a promising tool for future brain research.



## 6. OPTIMAL WAVELET DESIGN VIA MPSE

For the last decade, wavelet transforms have attracted a lot of interest. Theory and algorithms have been developed rapidly. Unlike the conventional Fourier transform, which suffers from fixed frequency and time resolutions, the wavelet transform is capable of trading off resolution between frequency and time domains within different frequency bands. This unique feature of the wavelet transform generated an explosion of applications in various areas of signal processing, such as in geophysics, astronomy, communications, radar, sonar, speech and image processing.

Among the various wavelet transforms, the most popular form used are the orthonormal discrete wavelet transforms (ODWTs), since an ODWT is computationally efficient and can be designed as a lossless process in the decomposition/composition operation. However, there are several problems related to ODWT: as a time-frequency analyzer, the time-frequency resolution and localization of an ODWT is poor; from a spectral analysis point of view, an ODWT produces a distorted spectrum with energy leakage, aliasing, and magnitude distortion; to produce a reasonable time-varying spectrum, the signal is required to be octave-distributed with a spectral shape matching the spectra of the wavelets. To overcome these problems, a new approach for the design of the optimal wavelet is proposed in this chapter. Using the multiresolution parametric spectral estimator, the new method continuously tracks the time-varying signal to adapt the optimal wavelets, which then yields high resolution, localization, and fidelity for the time-frequency decomposition [130]. The optimal wavelets also act as matched filters, so that they can greatly reduce broadband background noise in the decomposition process [115, 116].

### 6.1 Wavelet transform background

The general form of the wavelet transform can be represented as the inner product of the signal  $y(t)$  and the *daughter wavelets*  $h_{\tau,\nu}(t)$ , i.e.

$$W(\tau,\nu)=\langle y(t),h_{\tau,\nu}(t)\rangle=\int_{-\infty}^{\infty}y(t)h_{\tau,\nu}^*(t)dt \quad (6.1)$$

where  $\nu$  is the time scale and  $\tau$  is the center of  $h_{\tau,\nu}(t)$ , or the time localization of the daughter wavelets. From a functional analysis point of view, the inner product of (6.1) provides a similarity measure between the signal and a set of daughter wavelets. From a signal processing point of view, the wavelet prototype, the *mother wavelet*  $h(t)$  is the impulse response of a bandpass filter centered at the frequency  $\overline{f_0}$  with an effective bandwidth of  $\Delta f$ , defined as [35]

$$\Delta f = \sqrt{2\pi \frac{\int_{-\infty}^{\infty} (f - \overline{f_0})^2 |H(f)|^2 df}{\int_{-\infty}^{\infty} |H(f)|^2 df}} \quad (6.2)$$

Controlled by the time scale  $\nu$  and the time localization  $\tau$ , the daughter wavelets  $h_{\tau,\nu}(t)$  result from translation and scaling of the mother wavelet  $h(t)$  and can be represented as

$$h_{\tau,\nu}(t) = \frac{1}{\sqrt{\nu}} h\left(\frac{t-\tau}{\nu}\right) \quad (6.3)$$

Using the time-scaling and time-shifting properties of the Fourier transform, the frequency responses  $H(f)$ , of the mother wavelet, and  $H_{\tau,\nu}(f)$ , of the daughter wavelet, are related by

$$H_{\tau,\nu}(f) = \sqrt{\nu} H(\nu f) e^{-j2\pi f \tau / \nu} \quad (6.4a)$$

Equation (6.3) shows that the daughter wavelets  $h_{\tau,\nu}(t)$  are also the impulse responses of bandpass filters, now centered at

$$\overline{f_{\tau,\nu}} = \overline{f_0} / \nu \quad (6.4b)$$

The effective bandwidths of the daughter wavelet bandpass filters can be represented as

$$\Delta f_{\tau,\nu} = \sqrt{2\pi \frac{\int_{-\infty}^{\infty} (f - \overline{f_{\tau,\nu}})^2 |H_{\tau,\nu}(f)|^2 df}{\int_{-\infty}^{\infty} |H_{\tau,\nu}(f)|^2 df}} \quad (6.5)$$

Substituting (6.4a) and (6.4b) into (6.5), we have

$$\Delta f_{\tau,\nu} = \sqrt{\frac{2\pi \int_{-\infty}^{\infty} \frac{1}{\nu^2} (\nu f - \bar{f}_0)^2 |H(\nu f)|^2 df}{\int_{-\infty}^{\infty} |H(\nu f)|^2 df}}$$

Using the substitution  $f' = \nu f$ , the above equation can be recognized as

$$\Delta f_{\tau,\nu} = \frac{1}{\nu} \sqrt{\frac{2\pi \int_{-\infty}^{\infty} (f' - \bar{f}_0)^2 |H(f')|^2 df'}{\int_{-\infty}^{\infty} |H(f')|^2 df'}}$$

or

$$\Delta f_{\tau,\nu} = \frac{1}{\nu} \Delta f \tag{6.6}$$

The main feature distinguishing the WFT and WT in the time domain is that the width of the time window is constant in the WFT, so that the number of oscillations of the basis functions increases with frequency, while the number of oscillations of the WT bases is constant over frequency with a time-varying window width. Substituting (6.4b) into (6.6) and defining the ratio of the effective bandwidth to the center frequency of the bandpass filter as the Q factor of the daughter wavelets, yields

$$Q = \frac{\Delta f_{\tau,\nu}}{f_{\tau,\nu}} = \frac{\Delta f}{f_0} = \text{constant} \tag{6.7}$$

Therefore, the scale  $\nu$  keeps the Q factor of the daughter wavelets constant. Unlike the conventional Fourier transform, which suffers from fixed frequency and time resolutions, the WT is capable of trading off resolution between frequency and time domains within different frequency bands. That is, it can zoom in a high frequency band to catch fast "local" events and zoom out in a low frequency band to observe slow "global" changes.

The WT provided in (6.1) can not be computed directly, since it is in continuous form. Like the continuous Fourier transform, the continuous wavelet transform is hardly used in practical applications unless the mathematical function for the signal is known and (6.1) can be solved analytically. Therefore, the continuous wavelet transform is of theoretical interest but not practically important. To be able to perform the WT in a digital computer, a discrete form of the wavelet transform (DWT) has to be implemented. The general DWT can be derived by discretizing the time scale  $\upsilon$  and the time localization  $\tau$  into a grid, i.e.

$$\begin{aligned} \upsilon &= \upsilon_0^n; & \upsilon_0 &> 1 \\ \tau &= m\tau_0\upsilon_0^n; & \tau_0 &> 0 \end{aligned} \tag{6.8}$$

where  $\upsilon_0$  and  $\tau_0$  are constants that determine the fineness of the grid, and the indices  $m$  and  $n$  are integers. When  $m$  and  $n$  increase linearly,  $\upsilon$  and  $\tau$  will change exponentially. The general form of the DWT is defined by

$$c_{m,n} = \langle y(t), h_{m,n}(t) \rangle = \int_{-\infty}^{\infty} y(t) h_{m,n}^*(t) dt \tag{6.9}$$

where the wavelets  $h_{m,n}(t) = \upsilon_0^{-n/2} h(\upsilon_0^{-n}t - m\tau_0)$ .

## 6.2 Wavelet transform issues

In the literature, we find two different kinds of DWTs. According to the linear dependency of the daughter wavelets, the DWT can be characterized as the *frames of wavelet transform* (frame decomposition) or the *orthonormal wavelet transform* (orthonormal decomposition) [10]. The frame DWT decomposition is a direct approximation of the continuous WT. The basis functions, the daughter wavelets, of this kind are usually linearly dependent, and the set of wavelets is redundant in terms of spanning the phase space. The redundancy of the linearly dependent wavelets depends on the choice of  $\upsilon_0$ . When  $\upsilon_0$  is chosen to be barely larger than 1,  $\upsilon$  and  $\tau$  are sampled in a fine grid. The DWT in (6.9) is then close to its continuous version in (6.1). Although these wavelets span the entire phase space, they are highly redundant; in other words, the phase space is

oversampled. When  $\nu_0$  is chosen to be larger, such as  $\nu_0 = 2$ , for which the decomposition is done octave by octave,  $\nu$  and  $\tau$  are sampled in a sparse grid. As a result, the redundancy in the wavelet set will be greatly reduced, but the wavelet set may not span the entire phase space. The decomposition, and later composition, can suffer from information loss. Only some special basis functions which satisfy additional conditions can guarantee to span the entire phase space. The tighter the wavelet frame is, the tighter the restrictions on the wavelet functions. The theory of wavelet frame [10] provides a general framework to choose proper  $\nu_0$  and  $\tau_0$  according to the choice of mother wavelet, and produces balance between wavelet redundancy and precision of reconstruction. When the wavelet functions are orthonormal, and span the entire phase space, the wavelet decomposition is referred to as an orthonormal wavelet decomposition, which has been proven to be equivalent to the Quadrature Mirror Filter banks (QMF) or subband-signal coding techniques.

The advantage of redundancy in the linearly dependent DWT is that the decomposition yields high resolution and good localization in the phase space. The linearly dependent basis functions are closely located in the phase space (good localization), each of them acting as a short time constrained and sharply tuned bandpass filter (high resolution). Taking the Morlet wavelet (a Gaussian windowed complex sinusoid [129]) for example, the product of the time resolution and the frequency resolution, the resolution cell size in the phase space, reaches the lower bound of the Heisenberg uncertainty principle which has been proven to be the best resolution that can be produced by any Fourier transform based method [35]. Another advantage from the good tuning feature of linearly dependent wavelets is better noise immunity. The wavelet redundancy also reduces the sensitivity to computational errors during the process.

The disadvantage of the highly redundant linearly dependent DWT is that the method is inefficient in terms of computation and storage. For instance, assume an unknown *chirp* signal decomposed by the Morlet wavelet. Although there is only one signal frequency component at any given moment, the transform has to decompose the signal from high frequency to low frequency with small time-frequency increments in order to produce a time-varying spectrum with high resolution and good localization. Since  $\nu_0$  is not an integer, it is hard to apply decimation during the process. If the signal frequency is changing continuously from 100 Hz to 16 Hz, with a sampling frequency of 500 Hz, the scalogram, an equivalent time-varying spectrum of the signal, is

obtained by a bank of 273 FIR filters with filter lengths from 40 to 600 samples, where  $\nu_0 = 1.01$  and  $\tau_0 = 1$ . The total computational effort for a MATLAB subroutine is about 500 Mflops for processing 3 seconds worth of samples. The cost in computation will be compared in Section 6.5 to that for our optimal wavelet decomposition. The computational inefficiency is the main difficulty associated with real-time implementation of linearly dependent wavelets.

In contradistinction to the linearly dependent wavelets, the orthonormal wavelets form an orthonormal basis set in the phase space, and there is no redundancy in orthonormal wavelets. Since  $\nu_0$  is selected as integer, such as  $\nu_0 = 2$ , decimation can be applied to the coarse signal level in each decomposition step. Therefore, the transform is highly efficient in both computation and storage, which is the major advantage of orthonormal wavelets.

There are several disadvantages associated with the orthonormal wavelet decomposition. First, as a time-frequency analyzer, the orthonormal decomposition is poor in time-frequency resolution and localization. Due to the sparse sample grid in the phase space, the decomposition produces a poor time-frequency localization. The frequency localization is equal to the width of the octave band. The decomposition can not tell where the frequency component is located within the octave band, how many spectral peaks exist or what the spectral shape of the signal components is. The additional constraint of orthonormal wavelets also precludes the free choice of highly concentrated envelope functions, such as the Gaussian function, and thus produces poor time and frequency resolutions. Another problem of orthonormal time-frequency analysis is that the decomposition result, the scalogram, is hard to interpret and compare with the decomposition results using different wavelet decompositions. The octave band-splitting filters, the daughter wavelets, are neither intended to eliminate energy leakage from the transition and stop bands, nor to reduce distortion in the passband. The QMF decimation will also cause unavoidable aliasing during the decomposition [54]. The decomposed signal component is thus a mixture of signal frequency powers from the octave band occupied by the wavelet and from its neighboring octave bands. It also has different degrees of distortion caused by the mismatch of the signal spectral shape and the octave bandwidth of the daughter wavelet, especially when the signal component has multiple-spectral peaks or a broadband background with a sharp peak at the edge of the octave band. From the spectral analysis point of view, the scalogram is a seriously distorted spectrum, although the energy leakage, aliasing, and passband distortion will cancel each other out in the composition

process. To yield a good time-varying spectrum, the signal requires to be octave distributed with a spectral shape matching the spectra of the wavelet [57]. In applications, such a requirement is not practical. The final problem with orthonormal decomposition is that the lossless composition is sensitive to the existence of noise or interference, the accumulation of computational error, and the intentional loss of small signal details, such as eliminating the 'empty' octave bands for data compression.

### 6.3 Adaptive wavelet transforms

The problems associated with DWTs mentioned in the last section are mainly due to the fact that the wavelet design is independent of the signal; in other words the wavelets used are not adaptable. As a Fourier transform based method, the wavelet transform is a nonparametric method. That is, the basis functions, or wavelets are observation (signal) independent. They are designed to span the entire possible signal space, rather than what the signal really occupies; even if only one signal frequency component exists at any given moment. Meanwhile there are many different wavelet functions, which can be characterized as either frame or orthonormal decompositions, and none is better in absolute terms than the others. The performance of the wavelets depends on the selection of the mother wavelet. Since the signal is nonstationary, the selected wavelet may perform well in one time period but poorly elsewhere. Therefore, a good strategy in wavelet design is to track the signal and adapt the wavelet to follow the changes. Instead of spanning the entire signal space, the wavelets are allowed to concentrate and span just the subspace occupied by the signal.

The idea of adaptive wavelet design was first proposed by Szu and his colleague [131]. To estimate the characteristics of the signal, a Radial Basis Function neural network (RBF) is used within each octave band. In the neural network, each radial basis function, the neuron, acts as a highly tuned bandpass filter. Only when the frequency of a signal component matches the resonance frequency of the neuron, will the neuron produce maximum output. This frequency selective property of the RBF is also referred to as the *cocktail effect*. After the properties of the signal components have been estimated, the center frequency of a pre-chosen wavelet will be set to the frequency of the signal component with proper time scale and translation. Thus, the center frequency of the daughter wavelet follows the changes of the signal in each octave band. When

there is more than one signal component in a single octave band, a *super wavelet* is constructed by linearly combining the single wavelets centered at each of the signal frequency components. The super wavelet will enhance all the signal components in the octave band. The spectral shape of the super wavelet can contain multiple peaks, and can thus closely resemble the signal spectrum in each octave band. It has been proven that a linear combination of admissible wavelets will still be admissible [132]. This proof established the mathematical foundation for constructing the super wavelet and wavelet adaptation.

Adaptive wavelet design is a two step process. The first step is to estimate the characteristics of the signal. In the second step the statistics of the signal are used to derive the optimal wavelet which best fits the signal. Some optimization criterion will be introduced in the second step. These two steps can run concurrently, so that an adaptive wavelet transform operation is established. To take advantage of decimation, the wavelets need to be organized octavely. The wavelet designed for each octave band is no longer a simple dilation and translation of the mother wavelet. Instead, it is a linear combination of a set of simple wavelets called *elemental wavelets*. Here we adopt the name, *super wavelet* [131], for the compound daughter wavelet of the octave band. The mother wavelet provides the prototype for the elemental wavelets. Potentially good elemental wavelet prototypes are the linearly dependent wavelets, such as the Morlet wavelet, or the Gaussian windowed real sinusoidal wavelet. These linearly dependent wavelets can sample the time-frequency space on a fine grid, so that the super wavelet can match the signal in every octave band.

Two problems remain with the existing adaptive wavelet design. First the RBF network plays a role as a bank of bandpass filters. Each of these can be thought of equivalently as a Gaussian enveloped wavelet function. The process of estimating the signal characteristics is accomplished by applying a wavelet transform in each octave band over an extra fine grid. The time and frequency resolution and localization will largely depend on how fine the grid is or how many RBF neurons the network has. The choice of the fineness of the RBF grid is signal independent. To guarantee good resolution and localization, the RBF network has to be configured fine enough even for a single signal frequency component in the octave band. Therefore, the method itself is computationally inefficient. The second problem is that the selection of the signal frequency is not enough to produce an optimal wavelet. There are other parameters, such as how to linearly combine wavelets to construct the super wavelet, i.e. what is a good set of weighting coefficients;



and also, what is the optimal bandwidth for each wavelet making up the super wavelet. These optimization problems have not been addressed in the existing techniques. In the next section, we propose a new approach to optimal design of the super wavelet by using Multiresolution Parametric Spectral Estimation (MPSE). This method not only provides a systematic way of estimating the signal components, it also generates an optimization procedure for using the signal component information to design an optimal super wavelet. Two different error measures are introduced in the process. The results can be controlled by weighting these error functions. The resulting method shows strong noise immunity as well.

#### 6.4 Optimal wavelet design

The key to optimal wavelet design is proper estimation of the signal components within the octave band. The estimation process provides information about the central frequencies of the elemental wavelets, produces the proper bandwidth for each elemental wavelet, and shows how to linearly combine them to form a super wavelet for the given octave band. In this section, we establish a systematic process to adaptively design the super wavelet for each octave band based on the MPSE spectral information. Two different error functions will be introduced for the optimization of the wavelet design. The discussion will focus on a single octave band, but the method is suitable for all other octave bands. For simplicity, we use the term *subband-signal* to refer to the signal components located in the octave band where the super wavelet is built.

One of the advantages of the parametric decomposition is that there is a direct relation between the signal power spectrum and the model parameters [66], i.e.

$$P_x(z) = \frac{\sigma_u^2 B(z)B^*(z^{-1})}{A(z)A^*(z^{-1})} \quad (6.10)$$

where

$$\begin{aligned} B(z) &= \sum_{k=0}^q b_k z^{-k} \\ A(z) &= \sum_{k=0}^p a_k z^{-k} \quad ; \quad a_0 \equiv 1 \end{aligned} \quad (6.11)$$

For a stable process all the poles of (6.10), the roots of  $A(z)$ , are inside the unit circle of the  $z$ -plane. These poles provide the peaks of the signal spectrum and the roots of  $B(z)$  provide the zeros of the signal spectrum. The magnitude of the discrete Fourier transform of the signal can be computed by taking the square root of (6.10) and letting  $z = e^{j\omega}$ , i.e.

$$|X(\omega)| = \frac{\sigma_u |B(z)|}{|A(z)|} \Big|_{z=e^{j\omega}} = \frac{\sigma_u \left| \sum_{k=0}^q b_k e^{-j\omega k} \right|}{\left| \sum_{k=0}^p a_k e^{-j\omega k} \right|} \quad (6.12)$$

If the signal is a real process, the poles are either real or occur in complex conjugate pairs located at the frequency of each signal component. The closer the pole to the unit circle of the  $z$ -plane, the sharper the spectral peak is. Thus, the AR parameters can fully determine the shape and the frequencies of spectral peaks, or the spectra, of signal components. When the signal is zero mean (no D.C. component), the number of signal components is equal to one half of the order  $p$  of the AR model. That is,  $p$  model parameters can represent a real signal process with  $p/2$  frequency components.

Assume that the subband-signal is a real zero mean process and has  $p/2$  ( $p$  is an even integer) components. After the AR parameters have been estimated, the magnitude of the Fourier transform of the subband-signal can be represented as:

$$|X(\omega)| = \frac{\sigma_u}{|A(\omega)|} \quad (6.13)$$

where the fractional frequency  $\omega \in [-\pi, \pi]$ . From (6.11), with  $z = e^{j\omega}$ , we find

$$|A(\omega)| = \sqrt{\left[ 1 + \sum_{k=1}^p a_k \cos(k\omega) \right]^2 + \left[ \sum_{k=1}^p a_k \sin(k\omega) \right]^2} \quad (6.14)$$

Since the signal is real, its Fourier transform is an even function, i.e.  $X(\omega) = X(-\omega)$ ; we can limit our discussion to the positive frequency region. The reason that we only consider the magnitude of

the subband-signal spectrum is that the optimal wavelet filter will be designed as a linear phase FIR filter. There is no phase distortion in that case.

After the subband-signal spectrum has been estimated, the next step is to build a super wavelet which best matches the signal in the octave band. The elemental wavelet used for constructing a super wavelet is a Gaussian windowed sinusoid whose spectrum can be represented as:

$$H_k(\omega) = e^{-\frac{(\omega-\omega_k)^2}{\tau_k}} + e^{-\frac{(\omega+\omega_k)^2}{\tau_k}} \quad k = 1, \dots, \frac{P}{2} \quad (6.15)$$

where  $\omega_k$  is the center frequency of the elemental wavelet and  $\tau_k$  is the decay rate of the spectrum, which controls the bandwidth of the elemental wavelet. As indicated, the number of elemental wavelets in a super wavelet is equal to one half the AR order. That is, each elemental wavelet will construct one frequency component. Thus, the center frequency of the elemental wavelets will be at the pole frequencies of the subband-signal spectrum. The spectrum of the super wavelet can be represented as:

$$H_{\text{sup}}(\omega) = \sum_{k=1}^{P/2} \gamma_k H_k(\omega) \quad (6.16)$$

where  $\gamma_k$  is a gain factor determined by the magnitude of the subband-signal spectrum at the center frequency of the elemental wavelet, i.e.

$$\gamma_k = |X(\omega_k)| \quad k = 1, \dots, \frac{P}{2} \quad (6.17)$$

Thus far, the elemental wavelet is designed to match the frequency and the magnitude of the peaks of the subband-signal spectrum. The only remaining parameter is the decay rate or bandwidth of the elemental wavelet, which will be determined by an optimization process.

To perform the optimization, we introduce two error functions. First define the error as the distortion due to the bandpass (super wavelet) filter. This distortion is related to the flatness of the passband and to the bandwidth. If all the signal frequency components pass the filter with the same gain and the same group delay, then the spectral shape of the subband-signal is preserved. Thus,

the difference between the original subband-signal spectrum, as estimated by the AR estimator, and the spectrum at the output of the wavelet filter will provide a measurement of the spectral distortion caused by the super wavelet decomposition. Thus the spectral error is defined as:

$$E_1 = \int_0^{\pi} (|X(\omega)| - |H_{\text{sup}}(\omega)| |X(\omega)|) d\omega \quad (6.18a)$$

Clearly the spectral error will be minimized if the spectrum of the super wavelet is an all pass filter. The second error function is defined as the difference between the subband-signal spectrum and the spectrum of the super wavelet, i.e.

$$E_2 = \int_0^{\pi} (|X(\omega)| - |H_{\text{sup}}(\omega)|) d\omega \quad (6.18b)$$

When the spectrum of the super wavelet matches the spectrum of the subband-signal, the second error function is minimized. This second, or matching error function is chosen to prevent the spectrum of the super wavelet from being overly flat. The matched filter for white noise, implemented by the super wavelet, can help to eliminate background noise and greatly improve the noise immunity of the optimal wavelet decomposition [101]. Therefore, each error function helps to improve one aspect of the super wavelet. Our general error function is constructed as the convex combination of the two error functions,

$$E = \lambda_1 E_1 + \lambda_2 E_2 \quad (6.19)$$

where the weighting coefficients  $\lambda_1, \lambda_2$  are non-negative and sum to one. Choosing different weighting coefficients, one can put different priorities on the error functions.

With the error function in (6.19), the bandwidth of the elemental wavelets can be optimized using the Least Mean Square (LMS) algorithm. The decay rate,  $\tau_k$ , of the  $k$ -th elemental wavelet can be updated recursively by the following equation

$$\tau_k(n+1) = \tau_k(n) + \frac{2\mu E}{\gamma_k^2} \quad k = 1, \dots, \frac{p}{2} \quad (6.20)$$

A fast convergence rate has been observed during the optimization.

Equation (6.16) is the frequency domain representation of the super wavelet. The time domain representation of the super wavelet can be obtained by using the inverse Fourier transform of the frequency representation. However, since the Fourier transform is a linear operator, and the inverse Fourier transform of a Gaussian is also a Gaussian, the time domain representation of the super wavelet can be derived analytically. The Fourier transform pair of a Gaussian windowed sinusoid is:

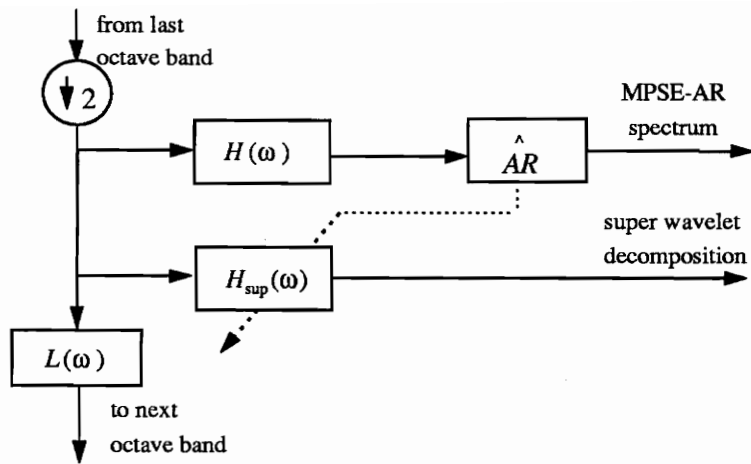
$$H(\omega) = e^{-\frac{(\omega-\omega_0)^2}{\tau}} + e^{-\frac{(\omega+\omega_0)^2}{\tau}} \Leftrightarrow h(n) = \frac{\sqrt{\tau}}{2\pi} e^{-\frac{\tau n^2}{4}} \cos(\omega_0 n) \quad (6.21)$$

The time representation of the super wavelet in (6.16) is thus:

$$h_{\text{sup}}(n) = \sum_{k=1}^{p/2} \gamma_k \frac{\sqrt{\tau_k}}{2\pi} e^{-\frac{\tau_k n^2}{4}} \cos(\omega_k n) \quad (6.22)$$

Since the super wavelet in (6.22) is an even function, i.e.  $h_{\text{sup}}(n) = h_{\text{sup}}(-n)$ , it is a linear phase filter. As  $h_{\text{sup}}(n)$  has infinitely long tails, it needs to be truncated to a finite time interval with a proper filter length. After truncation the super wavelet also needs to be delayed by one half of its total length to get a causal linear phase filter.

The above process provides a systematic way of designing the optimal super wavelet for any of the octave bands. Figure 6.1 shows a system diagram for the optimal wavelet design process. Similar to the orthonormal wavelet decomposition, the optimal wavelet decomposition consists of a highpass filter and a lowpass filter for band splitting. The only difference between the two wavelet decompositions is that the highpass filter in the optimal wavelet decomposition is adaptable. This adaptation is performed based on an MPSE-AR spectral estimator which provides information about the signal distribution in the octave band.

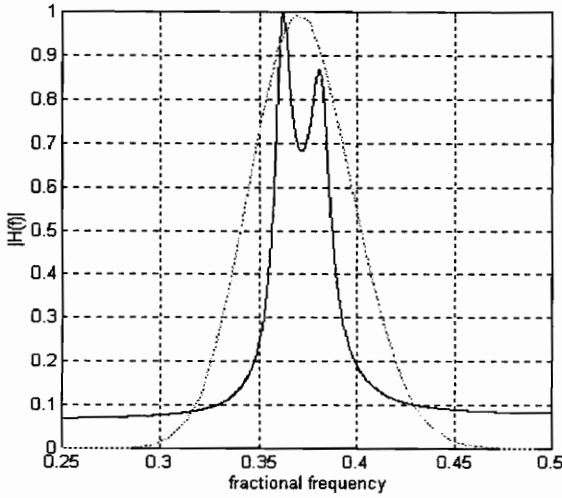


**Figure 6.1** The system diagram for Optimal wavelet decomposition.

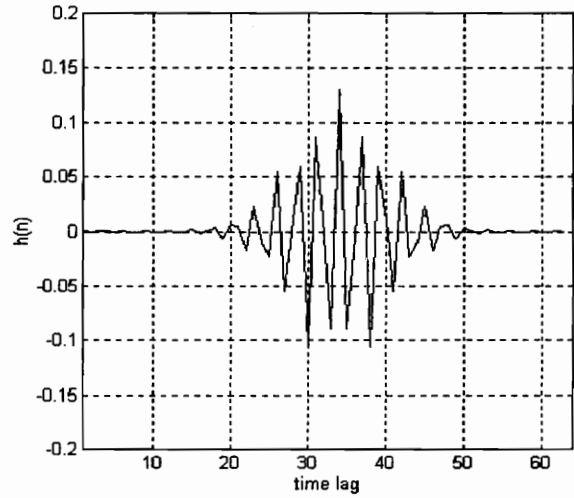
## 6.5 Experiments

Figures 6.2 and 6.3 provide two examples of the optimal wavelet design process. In Figure 6.2, the subband signal is a fourth order AR process with two positive frequency poles, i.e. the signal contains two frequency components, where one is narrower in bandwidth and with larger power than the other (Figure 6.2a solid line). White noise is added to this signal to simulate an observation noise with SNR of 20 dB. Using the optimal wavelet design procedure above, a super wavelet is constructed by using two elemental wavelets located at each spectral peak of the spectrum of the subband-signal. The weighting coefficients  $\lambda_1, \lambda_2$  are chosen to have the same value of 0.5. The LMS learning curve in Figure 6.2d shows how fast the super wavelet converged to its optimal form. The spectrum of the super wavelet is presented as the dotted line in Figure 6.2a; its time representation is given in Figure 6.2b. In Figure 6.2c, the dotted line indicates the ideal spectrum of the subband-signal in the octave band (noise free), the solid line is the spectrum of the observation. Finally, the dashed line in Figure 6.2c indicates the spectrum of the output from the super wavelet highpass filter. The result shows that the optimal wavelet decomposition can preserve signal information with minimum spectral distortion and maximum broadband noise elimination. Figure 6.3 provides another example of optimal wavelet design. The only difference between these two examples is that the subband-signal now contains three frequency components

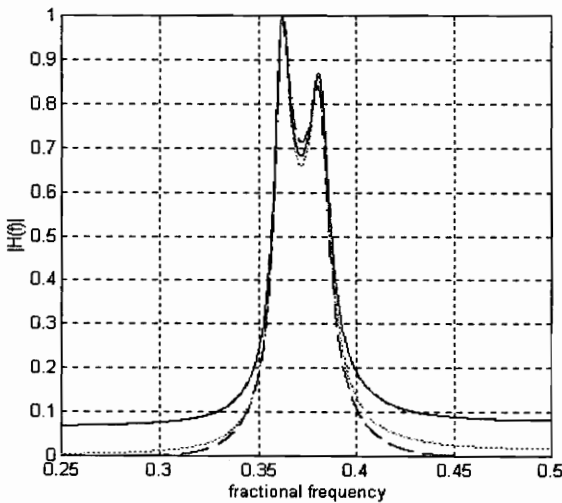
(Figure 6.3a solid line). The spectrum of the super wavelet is presented as the dashed line in Figure 6.3a; its time representation is given in Figure 6.3b. The super wavelet, in Figure 6.3c, shows good noise immunity as well.



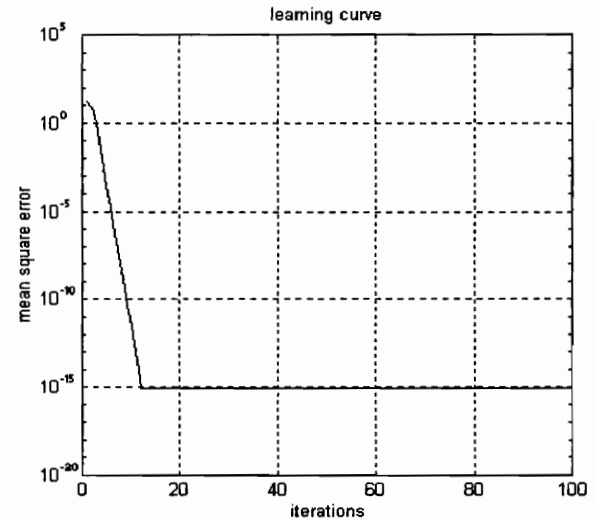
**Figure 6.2a Super wavelet spectrum (dotted line) and signal spectrum (solid line).**



**Figure 6.2b Super wavelet time representation.**



**Figure 6.2c Sub-band signal spectrum (solid line), noise-free sub-band signal spectrum (dotted line), and super wavelet filter output spectrum (dashed line).**



**Figure 6.2d 4-th order super wavelet LMS learning curve.**

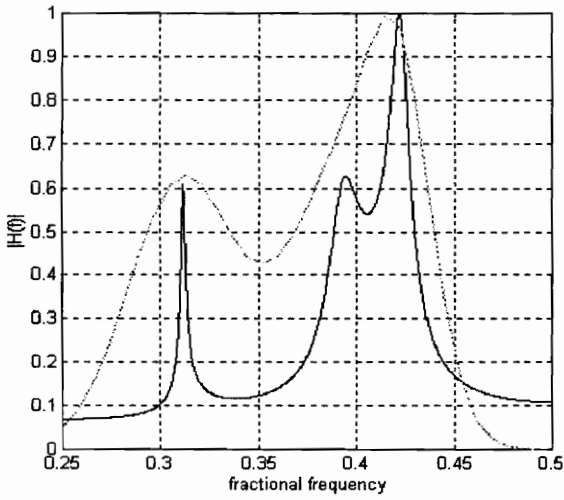


Figure 6.3a Super wavelet spectrum (dotted line) and signal spectrum (solid line).

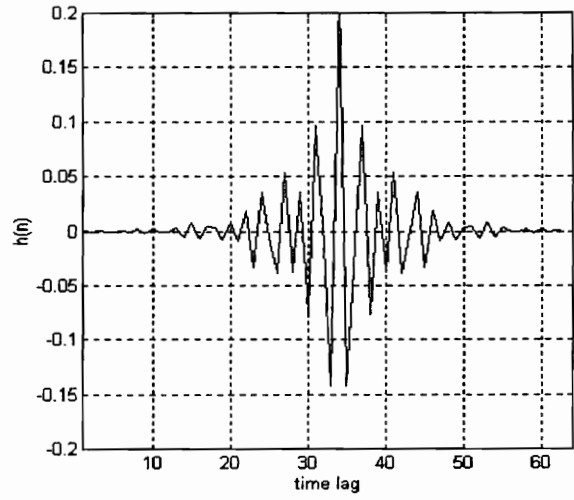


Figure 6.3b Super wavelet time representation.

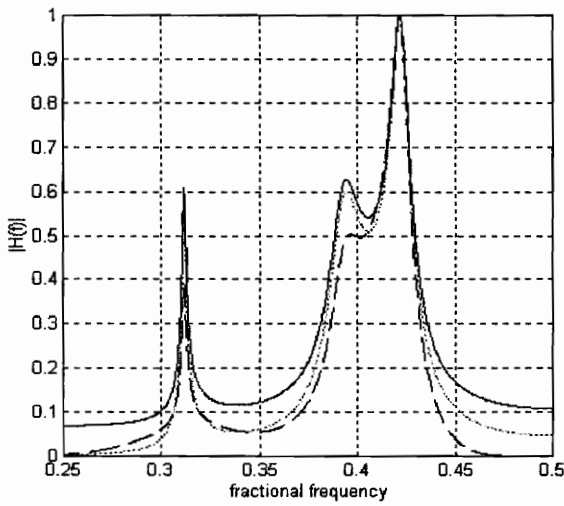


Figure 6.3c Sub-band signal spectrum (solid line), noise-free sub-band signal spectrum (dotted line), and super wavelet filter output spectrum (dashed line).

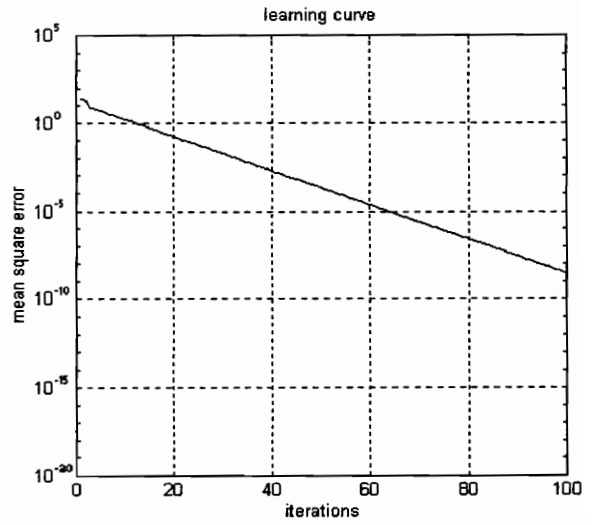


Figure 6.3d 6-th order super wavelet LMS learning curve.



Taking the same chirp signal as in Section 6.2 as an example, since there is only one signal frequency component at any given moment, the optimal wavelet design will produce only one super wavelet whose center frequency varies with the chirp signal. The total computation in a MATLAB program is about 4.6 Mflops for processing 3 seconds worth of samples.

## **6.6 Summary**

In this chapter we have proposed a systematic way of optimal wavelet design, such that the super wavelet can track the subband signal and adapt its spectrum to follow the changes. The resulting wavelet design is optimal in terms of better time-frequency resolution and localization, less spectral distortion during decomposition, and better noise immunity than for other wavelet transforms. It is also optimal in the efficiency of computation and storage. Since the super wavelet is designed octavely, decimation can be applied to the coarse subband signal in every decomposition step. As the central frequency of the super wavelet can track and match the changes in the signal, the number of octave bands generated by the decomposition depends on the order of the signal model, instead of on the lowest frequency in the signal component. Thus, our method produces fewer 'empty' octave bands in many applications, which can increase efficiency in terms of computation and storage.

# 7. RADIAL BASIS FUNCTION DECOMPOSITION AND ADAPTATION

## 7.1 Introduction

In the previous 5 chapters, we have developed the Multiresolution Parametric Spectral Estimator (MPSE) in theory, techniques and applications. The main focus of these chapters is on the localization of the characteristics of a time-varying signal in time-frequency space (phase space). Compared with the other linear decomposition methods, such as the Fourier transforms, the wavelet transforms, and the fixed resolution parametric spectral estimators, the advantage of the MPSE is its high resolution and flexibility of the multiresolution realization. In many applications, the time-varying characteristics of the signal or system actually reflect its nonlinear nature. As discussed in Section 1.4, a nonlinear system can be approximated as a time-varying linear system. If the linear technique used can track fast enough, the behavior of a nonlinear system can still be followed. Later, in Section 7.3, an illustrative example will be provided, where a linear dynamic model-based WRLS algorithm is used to track a Lorenz chaotic nonlinear system. However, the time-varying linear approximation depends on how strong the nonlinearity is, how fast the algorithm can follow, and how fast the measurements can be made. In many cases, a simple nonlinear problem can turn into a complex problem with the time-varying linearization approach, and very likely the attempt will fail. Since MPSE is a linear dynamic model-based method, it certainly has limitations in dealing with nonlinear processes.

In a practical situation, when the time-varying behavior of an unknown process is observed, one should be concerned with whether or not the time-varying observation is due to linear approximation. When the nonlinearity of the process under observation is taken into account, the process may be stationary or time-invariant instead of time-varying, or the time-varying rate can be significantly slowed down. With appropriate nonlinear techniques a nonlinear processing problem may be solved more easily than with a linear approximation approach.

The main difficulty in modeling, or analyzing, a nonlinear dynamic system is the lack of general theory and systematic methods. Unlike with linear systems, where the local properties are also valid globally, the properties of a nonlinear system usually remain local. There is no general theory

as a navigator, or general techniques, for solving nonlinear problems. In linear decomposition, the merit function for solving the decomposition coefficients has a unique minimum, while in nonlinear system decomposition the merit function may encounter multiple-minima. Although the problem of searching for the global minimum of a multi-minimum merit function is an ancient topic, so far no method can guarantee to find the global minimum, except in two very special cases. The first case is that the existence of a lower bound of the merit function can be proved either mathematically or physically, and one of the minima found is on that lower bound. This minimum is at least one of the global minima. The other case is that the total number of minimum points associated with the merit function has been found analytically, and all of them have been found in the search process. The smallest one among those minima is the global minimum. Unfortunately, for most practical problems, we know neither the mathematical form nor the existence of the lower bound of the merit function. Searching for a global minimum is still greatly dependent on one's fortune.

Since the nonlinear problems are case related, finding general decomposition basis functions is another challenge. Polynomials have been widely used in the past. There are many different ways of approach, either directly or indirectly. Using the Taylor series expansion as the basic model and estimating the expansion coefficients is a direct approach [133]. Popular indirect polynomial decomposition methods are the Group Method of Data Handling (GMDH) [134-136], as well as most of the neural network representations [137]. Searching for the global minimum, to estimate the model coefficients, is a major problem for polynomial-based decompositions. The other problem is that the basis functions,  $\{x^0, x, x^2, \dots, x^k\}$  extend from negative infinity to infinity. To construct the dynamic hyper-surface (Section 1.4) with infinitely extended basis functions, the problem has to be solved globally.

The Radial Basis Function (RBF) approach is a newly developed modeling method for nonlinear processes. As mentioned in Section 1.4, the RBF are locally supported so that the optimization problem can be extended from a small local region to the entire dynamically feasible region. It also provides great flexibility and potential for constructing high dimensional dynamic hypersurfaces with locally supported dome-shaped RBFs as the building blocks. However, since the RBFN is a newly developing technique, there are many problems associated with it. For example, there are more free parameters in the RBF than in a polynomial approximation and the judgment on which parameters should be constrained and which should remain unconstrained has not been rendered.

The goal for this chapter is to develop a tool for constructing an RBF representation for nonlinear processes and to explore the potential of the RBFN approach for nonlinear process modeling. In Section 7.2, the Equal-Distance Sampling Rule (EDSR) is proposed as a tool for constructing an RBFN. Several application examples can be found in Sections 7.3 and 7.4. Meanwhile, the experiments for the RBFN structure adaptation are provided in Section 7.5.

## 7.2 Radial basis function network construction

### 7.2.1 Solving for the model parameters of the radial basis function network

In the literature, the RBF decomposition is often referred to as a Radial Basis Function Network (RBFN), because the model representation and the network representation are equivalent. That is, both provide a functional linkage between one set of variables (the independent or input variables) and another set of variables (the dependent or output variables). In later discussion the term RBFN will be used to refer to either the radial basis function decomposition, or to the radial basis function network.

The RBFN has the following general form:

$$\begin{aligned}
 y &= F(\mathbf{x}) \\
 &= \sum_{i=1}^N a_i \phi_i(\|\mathbf{x} - \mathbf{c}_i\|)
 \end{aligned} \tag{7.1}$$

where  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$  is an input vector in  $\mathbb{R}^n$  space,  $y = F(\mathbf{x})$  is a nonlinear multivariate function which maps from  $\mathbb{R}^n$  to  $\mathbb{R}$ , with  $\{a_i: i = 1, 2, \dots, N\}$  as the set of model parameters, and  $\phi_i(\|\mathbf{x} - \mathbf{c}_i\|)$  is the  $i$ -th Radial Basis Function (RBF) centered at  $\mathbf{c}_i = [c_{1,i}, c_{2,i}, \dots, c_{n,i}]^T$ .

Many system identification or signal modeling problems can be treated as the reconstruction of  $F(\mathbf{x})$  from observations. After the basis functions have been selected, the modeling task is to find the associated set of model parameters  $\{a_i: i = 1, 2, \dots, N\}$ . For example, if  $M$  input-output pairs have been observed, these input-output pairs are linked by (7.1) as:

$$\begin{aligned}
y_1 &= F(\mathbf{x}_1) = \sum_{i=1}^N a_i \phi_i(r_{1,i}) \\
y_2 &= F(\mathbf{x}_2) = \sum_{i=1}^N a_i \phi_i(r_{2,i}) \\
&\vdots \\
y_M &= F(\mathbf{x}_M) = \sum_{i=1}^N a_i \phi_i(r_{M,i})
\end{aligned} \tag{7.2a}$$

where  $r_{j,i} = \|\mathbf{x}_j - \mathbf{c}_i\|$  is the norm distance from the  $j$ -th input vector  $\mathbf{x}_j$  to the  $i$ -th center  $\mathbf{c}_i$ .

Equation (7.2a) can also be written in matrix form:

$$\begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} \phi_{1,1} & \cdots & \phi_{1,N} \\ \vdots & \cdots & \vdots \\ \phi_{M,1} & \cdots & \phi_{M,N} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix} \tag{7.2b}$$

or

$$\mathbf{y} = \Phi(\mathbf{X})\mathbf{a} \tag{7.2c}$$

where  $\phi_{i,j} = \phi_i(r_{j,i})$ ,  $\mathbf{y} = [y_1, y_2, \dots, y_M]^T$  is a collection of outputs,  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M]^T$  is a collection of input vectors, and  $\mathbf{a} = [a_1, a_2, \dots, a_N]^T$  is the model parameter vector.

We distinguish three possible cases in solving (7.2) for the set of parameters. The first case is for  $M < N$ , when the number of equations is less than the number of unknowns. Equation (7.2) then has no unique solution, and more input-output samples are required. The second case is for  $M = N$ , when the number of equations and unknowns is equal, and the matrix  $\Phi(\mathbf{X})$  is not singular. Equation (7.2) now has a unique solution for the set of parameters  $\mathbf{a}$ . The result is a hypersurface, defined by (7.1), on which all the input-output samples lie. The last case is for  $M > N$ , when the number of equations is larger than the number of unknowns. If the unknown dynamic system is a deterministic system, there are  $M - N$  row-vectors in the matrix  $\Phi(\mathbf{X})$  that are linearly dependent and can be eliminated from  $\Phi(\mathbf{X})$ , and now (7.2) has a unique solution. If the system is a stochastic dynamic system, or the observations have been contaminated by background

noise, a least squares solution of (7.2) can be obtained by using the pseudo-inverse of the matrix  $\Phi(\mathbf{X})$ , i.e.

$$\hat{\mathbf{a}} = \Phi^\#(\mathbf{X})\mathbf{y} \quad (7.3)$$

where  $\Phi^\#(\mathbf{X})$  denotes the pseudo-inverse of the matrix  $\Phi(\mathbf{X})$ , and  $\hat{\mathbf{a}}$  denotes the least squares estimate of the true set of model parameters  $\mathbf{a}$ . The constructed hypersurface has the shortest average distance to all the input-output samples.

To determine the fit of the model to an unknown system, an error function  $\xi$  is evaluated by

$$\xi = \frac{1}{M} \sum_{k=1}^M \left\| y_k - \sum_{i=1}^N a_i \phi_i(\|\mathbf{x}_k - \mathbf{c}_i\|) \right\| \quad (7.4)$$

Usually, the data set used to build the model and the data set used to test the model validity are different sets, which helps avoid over-fitting of the sample data. This is very important for constructing the stochastic dynamic model.

### 7.2.2 Selection of center locations for the radial basis functions

In constructing the RBFN representation of a signal model or unknown system, the most important, and also difficult, aspect is the selection of the RBFs based on the given samples. There are many free parameters, in addition to the set of model parameters  $\mathbf{a}$ , for RBFs. For instance, if the mathematical form of the basis function  $\phi_i(\mathbf{x})$  is chosen as Gaussian (1.13b), the free parameters are the dilations of the Gaussian functions, and their center locations. If the dimension of the input space  $n > 1$ , the dilations along different directions can also be different. Moreover, the total number of basis functions used is another factor important to the RBFN.

In reported results, the dilations  $\sigma^2$  of the basis functions are usually the same for all the basis functions and all the directions. The choices are either arbitrary constants or determined by the reachable dynamic region of the system under identification [98]. The centers of the RBFs are selected from the input vectors; they are either the entire collection of input vector samples or a subset of it [98]. If all the input vectors in the training data set are chosen and the number of input

vectors in the set is large, it may no longer be practical to compute and store  $\Phi(\mathbf{X})$  which is of size  $M \times M$ . When many centers are located close to each other, the matrix  $\Phi(\mathbf{X})$  will be nearly singular, or ill conditioned. It then becomes difficult to solve (7.2) numerically. In modeling a stochastic system, closely located basis functions may also overfit the hypersurface.

A sub-optimal method to select the centers has been proposed by Chen, Cowan and Grant in their research on the identification of nonlinear systems [98]. Their procedure is called the Orthogonal Least Squares learning algorithm (OLS). The OLS algorithm starts with the selection of the first basis center from the entire input vector set with  $M$  members. Equation (7.2) is then solved by the least squares method with  $N = 1$ . The approximation error  $\xi$  of the one basis function model is also computed. This procedure will be repeated until all the members in the collection of input vector samples have been tested. The input vector sample producing the minimum approximation error is then chosen as the first 'optimal' center location for the RBF. The second center location for the RBF is then chosen from the remaining input vector set with  $M-1$  members. The OLS will be used to solve (7.2) for the parameters of the model with two basis functions ( $N=2$ ), and the model approximation error  $\xi$  is calculated. This procedure is repeated until all possible second basis functions have been tried from the remaining input vector set. The *optimal* second basis function is chosen according to the smallest approximation error. The subsequent basis functions are selected following the same procedure, adding new basis functions until the approximation error has reached an acceptable level. The solution is Suboptimal because the center locations are selected one after the other.

The problem of the OLS method is that it is computationally intensive and requires large storage space, especially when the number of input-output samples and/or the number of final basis functions is large .

A simple method is proposed here for selecting the centers for an RBF. The method is named the Equal-Distance Sampling Rule (EDSR), which is partly inspired by sampling theory in signal processing. When a continuous signal is sampled by a  $\delta$  sequence, the sample interval is usually fixed. Although, this sample rate may not be optimal for a particular part of the signal waveform, the general performance can be preserved if the Nyquist sampling rate is satisfied. It is rarely practical to adaptively change the sampling rate according to the signal waveform, although the

efficiency can be much higher. Thus, the EDSR is designed to sample the unknown hypersurface with  $n$ -dimensional Gaussian basis functions spaced at approximately ‘uniform’ distances.

In the real world, the dynamic range of the output and input of a system is always finite. The shape of the accessible region in the input space is usually irregular. For instance, the nonlinear Henon equation is a second order quadratic difference equation defined by [138]

$$\begin{aligned} y_n &= F(y_{n-1}, y_{n-2}) \\ &= c - ay_{n-1}^2 + by_{n-2} \end{aligned} \tag{7.5}$$

where  $a, b, c$  are constant parameters ( $c$  is also equivalent to a constant driving force of the dynamic system). The input space of the difference equation is composed of  $y_{n-1}$  and  $y_{n-2}$ , a two dimensional space. The dynamic surface  $F(\cdot, \cdot)$  is hyperbolic along  $y_{n-1}$  and linear along  $y_{n-2}$ . By selecting different parameter sets  $\{a, b, c\}$ , the above difference function can demonstrate periodic, unstable, or chaotic behaviors. When the parameters  $b$  and  $c$  are set at  $b=0.3$  and  $c=1$ , by continuously varying parameter  $a$  from 0.5 to 2.0, the Henon dynamic system exhibits different periodic and chaotic behavior. Plotting all possible statuses of the output of the system  $y_n$  against parameter  $a$ , we produce the so called bifurcation map of the Henon equation (Figure 7.1a). From the diagram, one can easily find the values of  $a$  which produce one possible status of  $y_n$ , two possible statuses of  $y_n$ , and  $k$  statuses ( $k=3,4,\dots$ ), as well as chaotic dynamics. When the system is in one status mode,  $y_n = y_{n-1} = y_{n-2}$ , there exists only one possible input vector  $[y_{n-1}, y_{n-2}]$  in the two dimensional input space. Thus the feasible input region is a single point. When the system is in the mode with  $k$  possible statuses, there are  $k$  possible input vector in the input space. When the system is in chaos, although  $y_n$  has numerous possible statuses, the input vectors still fill in a small arch-like belt area of irregular shape in the input space (see Figure 7.1b). Since the input dynamic ranges are  $-1.4 < y_{n-1} < 1.4$ , and  $-1.4 < y_{n-2} < 1.4$ , the input space can be sampled on an  $8 \times 8$  grid with a sampling distance of 0.4, and the total number of basis functions is 64 to cover the square area containing the dynamic region. Of these RBFs many are in the inaccessible dynamic area.



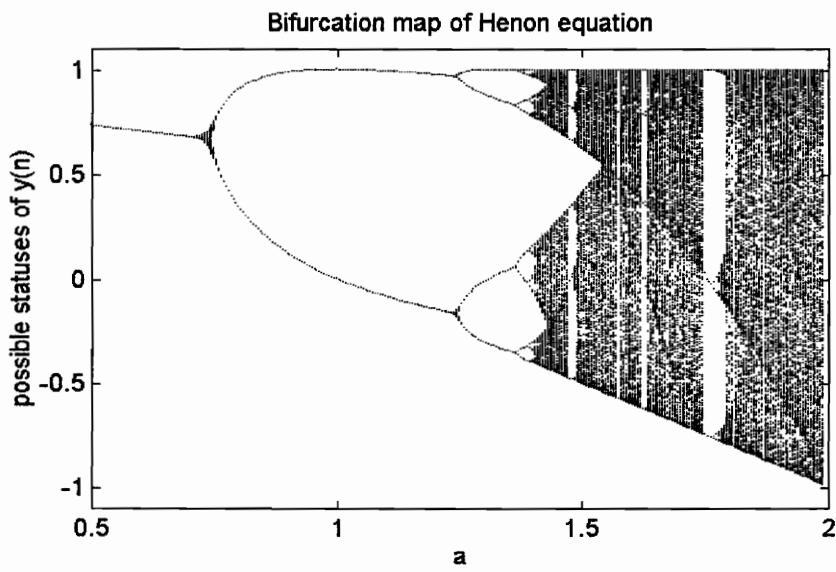


Figure 7.1a Bifurcation map of the Henon equation.

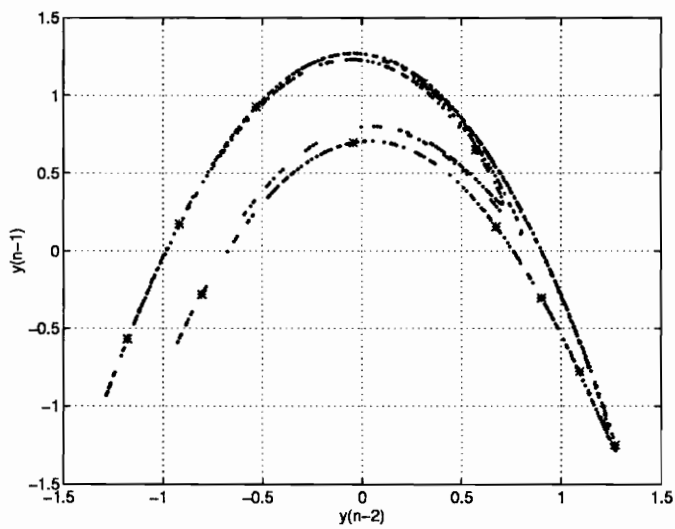


Figure 7.1b The input vectors '\*' and the selected centers for the RBF '\*'.

The input space in the Henon equation is low dimensional. If the input space were high dimensional, to sample the input space on a uniformly distributed grid in a hypercube, the total number of basis functions would be enormous, and most would fall in the dynamically inaccessible area. To cover the entire dynamic region without putting any basis function in the unreachable region, the EDSR is designed to choose the center locations of the basis functions from the available input vectors with an approximately equal distance. The EDSR starts with arbitrarily selecting an input vector as the first center of the RBF, and measuring the distances from that center to all other input vectors. A second center is selected from the input vector set according to the distance to the first center, which is closest to the selected sampling interval. Meanwhile, input vectors will be discarded from the set of center location candidates if they are close neighbors, i.e., if they fall radially inside the region of the sampling interval centered on the first two basis functions. The procedure is repeated until all members of the input vector set have either been selected as centers or discarded as being close neighbors. After the centers of the RBFs are chosen, the set of model parameters is solved for by using the pseudo-inverse approach in (7.3).

In the temporal domain, the sampling period can be determined by analysis of the temporal frequency of the signal. For the RBFN, the spatial frequency of the hypersurface can not be analyzed since it is supported by the input space with an irregular geometric shape. To determine the proper sampling distance, the EDSR partitions the input-output training data set into two separate sets. One set is for constructing the RBFN, i.e. for choosing the centers of the basis functions and for estimating the model parameters. The other set is for testing the validity of the model. The construction of the EDSR starts from an initial sampling distance  $\delta_0$ . The RBFN based on  $\delta_0$  is evaluated by using (7.4) to compute the approximation error  $\xi_0$  with the second data set. In the second iteration, the new sampling distance  $\delta_1$  is chosen from

$$\delta_1 = \delta_0 + \mu(\xi_{end} - \xi_0) \quad (7.6a)$$

where  $\mu$  is the step size, and  $\xi_{end}$  is the maximum acceptable approximation error. Repeating the first step, we obtain the new approximation error based on  $\delta_1$ . Starting from the third iteration, the new sampling distance is chosen from

$$\delta_{k+1} = \delta_k + \mu(\xi_k - \xi_{k-1}) \quad (7.6b)$$

The iteration continues until the error is lower than the desired value  $\xi_{end}$ , or the error function no longer converges, i.e.  $|\xi_k - \xi_{k-1}| < \text{threshold}$ .

The advantages of the EDSR are simplicity and effectiveness. There is no sophisticated computation, and there is no matrix operation involved, so that it can work easily for a large training set in a high dimensional input space. For the example of the Henon equation, with 0.4 sampling distance, the total number of centers required to cover the entire dynamic feasible input space is 11 (Figure 7.1b). Compared with 64, the efficiency of the EDSR is obvious. Another advantage of the EDSR is its flexibility. If any new sample becomes available later, and it is not within the sample radius of any of the selected centers, the new sample can be added as a new basis center. Therefore, the RBFN built by the EDSR is potentially adaptable during its operation.

The performance of the RBFN using the equal-distance sample rule has been tested extensively in modeling linear and nonlinear (chaotic) dynamic processes for different orders of system dynamics and SNR conditions. Part of the experimental results are presented in Sections 7.3 and 7.4. Meanwhile, the performance of the RBFN using the equal-distance sample rule has been compared with those using the OLS learning rule. When the number of basis functions for the EDSR and OLS was selected to be the same, both methods produced the same quality of results for the given experiments. It was thus demonstrated that the performance of an RBFN is not particularly sensitive to exactly which centers have been picked up if the distribution density of the RBFs is satisfied and the entire feasible dynamic range is covered.

### **7.3 RBF decomposition for time series prediction**

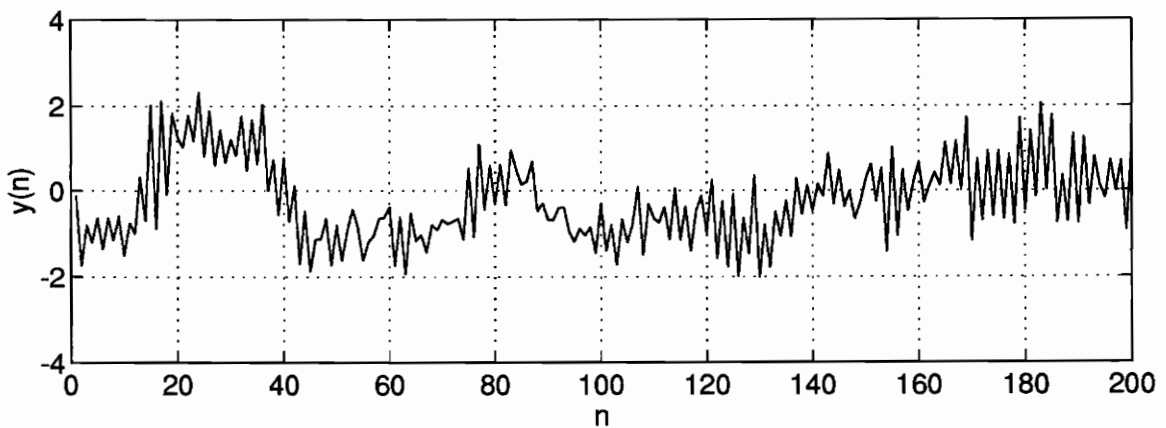
Three computer simulations were designed to demonstrate the performance of linear and nonlinear modeling methods in one step forward prediction of the linear or nonlinear (chaotic) processes. The linear method used was either the WRLS or Burg algorithm, which are based on the linear AR model. The nonlinear method was the RBF representation of an unknown dynamic function. The center locations of the RBF were chosen according to the EDSR given in Section 7.2. The weighting coefficients of the decomposition were estimated by using the least square algorithm. The computer simulations are compared on the basis of the residual of the forward prediction of an unknown dynamic time series.

First, a linear dynamic process was used. This experiment is designed to demonstrate that the RBF modeling could yield the same good results for a linear system as a linear model-based method. After that, two nonlinear (chaotic) processes were used in the computer simulations. One is a continuous dynamic process generated by the Lorenz equation, and the other is a nonlinear discrete process generated by the Henon equation. Both the linear and nonlinear modeling techniques have been tested in the computer simulations. These experiments are designed to explore the performance of the piecewise linear and nonlinear modeling techniques, and to test the EDSR for selecting the RBF centers as well.

The linear dynamic time series was generated by a second order AR process (Figure 7.2):

$$y_n = 0.1y_{n-1} + 0.8y_{n-2} + \eta_n \tag{7.7}$$

where  $\eta_n$  is Gaussian white noise with zero mean and variance  $\sigma^2 = 0.27$ . In the computer simulation, two sets of sample data were generated by (7.7) with 200 samples in each data set. The first set is used for constructing the model and the second set is used for testing the validity of the model.



**Figure 7.2 AR(2) linear time series.**

A second order AR model was used for the linear forward prediction. The model parameters were estimated by the Burg algorithm. The variance of the linear prediction error is  $\hat{\sigma}_{AR}^2 = 0.2611$ , which is close to ideal. Five basis functions were selected by the EDSR for the RBF decomposition model (Figure 7.3). These basis functions constructed a linear dynamic 'plane' in the input space (Figure 7.4). To compare the estimated dynamic plane with the ideal one, Figure 7.5 provides the cross sections of the estimated (the solid lines) and original dynamic plane (dashed lines), i.e.

$$\begin{cases} y_n = 0.1y_{n-1} & \text{the projection on } y_{n-2} = 0 \text{ plane} \\ y_n = 0.8y_{n-2} & \text{the projection on } y_{n-1} = 0 \text{ plane} \end{cases} \quad (7.8)$$

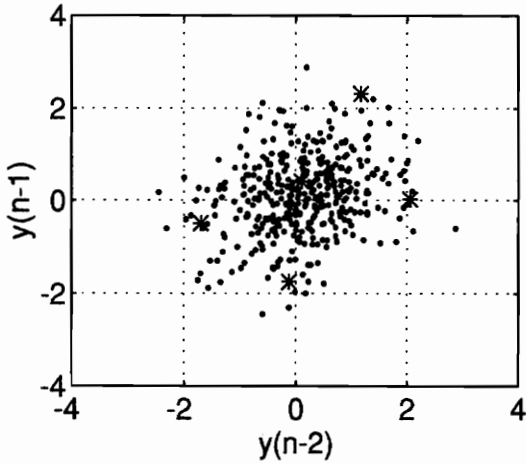
The diagram shows that the estimated dynamic plane is not perfectly 'flat', as it has the freedom not to be, but it is close to that. The variance of the forward prediction error for the RBF decomposition method is  $\hat{\sigma}_{RBF}^2 = 0.2612$ . Both the linear and the nonlinear estimator give almost the same performance in the computer simulation.

In the second computer experiment, the testing data were generated by the Lorenz equation which is defined by

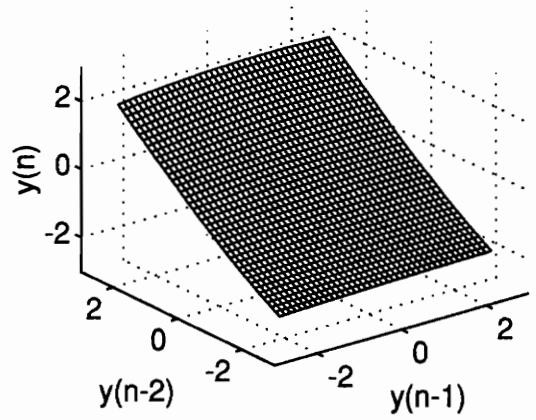
$$\begin{cases} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = -xz + \alpha x - y \\ \frac{dz}{dt} = xy - \beta z \end{cases} \quad (7.9)$$

where  $\alpha$ ,  $\beta$  and  $\sigma$  are constants. Chaotic solutions of the Lorenz equation result with  $\alpha = 28$ ;  $\beta = 8/3$ ; and  $\sigma = 10$  [139]. The Lorenz equation is a third order nonlinear differential equation whose phase space is three dimensional with variables  $x(t)$ ,  $y(t)$ , and  $z(t)$ . The Lorenz dynamic system has two attractors and a repeller (unstable equilibrium point). Figures 7.6a,b provide the 3 dimensional dynamic trajectory of the Lorenz equation and the one dimensional projection of the trajectory  $x(t)$ . Two dynamic basins are shown in Figure 7.6a, which are

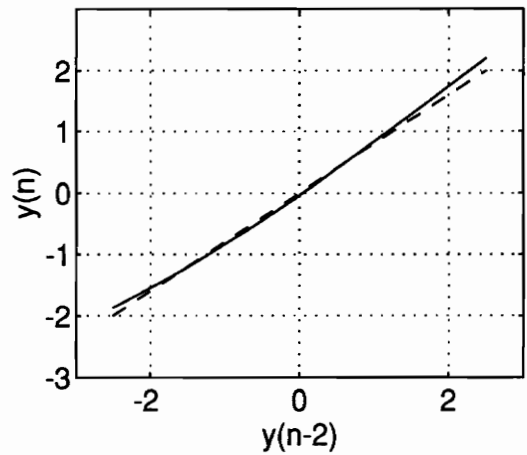
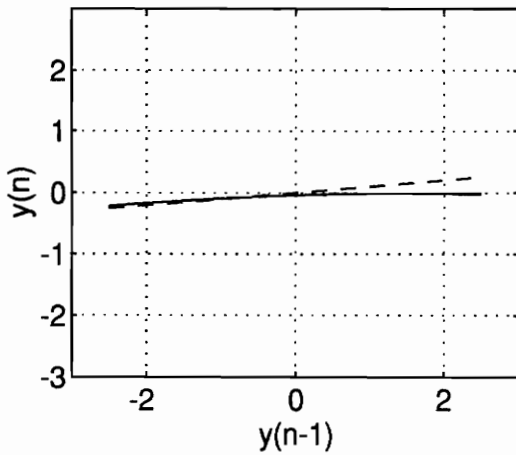
organized and centered by the attractors and separated by the repeller. The dynamic trajectories were generated using a Runge-Kutta method.



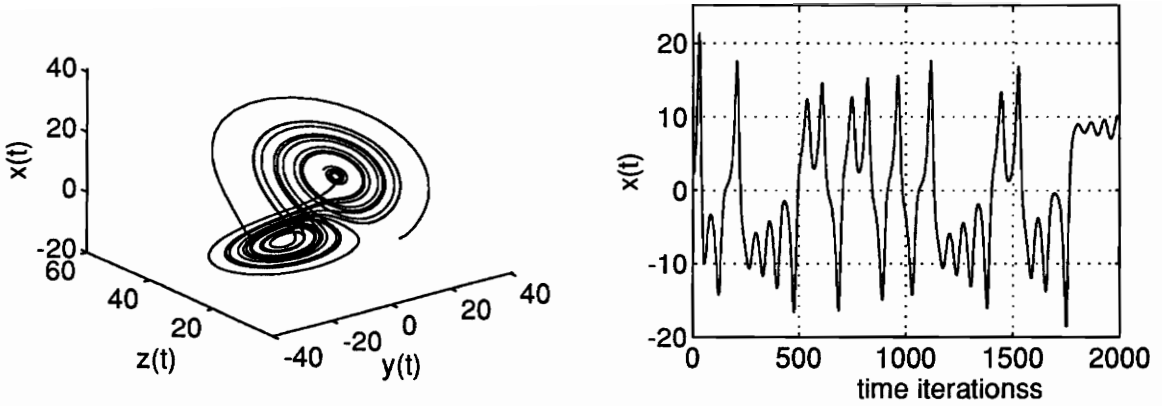
**Figure 7.3** The input space, where the input vectors are plotted as '.', and the RBF centers are plotted as '\*'.



**Figure 7.4** Dynamic plane reconstructed by the RBFN.



**Figure 7.5a, b** Cross sections of the estimated (solid) dynamic plan and the ideal (dashed) plane.



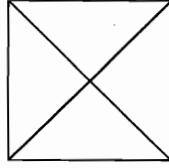
**Figure 7.6 (a) 3 dimensional dynamic trajectory of the Lorenz equation; (b) 1 dimensional trajectory  $x(t)$ .**

In the simulation, a third order AR forward estimator was used to perform piecewise linear one step forward prediction of the one dimensional trajectory  $x(t)$ . The parameters of the AR model were continuously estimated by the WRLS algorithm with a sliding window of 50 data points. The residual  $e(t)$  of the AR forward prediction of  $x(t)$  (Figure 7.7a) is relatively small compared with the actual values. The variance of the residual of AR prediction is  $5.4794 \times 10^{-5}$ . We also use the relative residual power as a performance measure. The relative residual power is defined by

$$\rho_e = \frac{\text{variance of the residual (power of linearly unpredictable part of the signal)}}{\text{variance of the signal ( total signal power)}} \quad (7.10)$$

$\rho_e$  is a ratio measuring the percentage of linearly unpredictable signal power. The relative residual power of the piecewise linear AR estimation is  $7.9356 \times 10^{-7}$ . This experiment demonstrates that if the tracking speed of a linear estimator is fast enough, the piecewise linear method can produce a good approximation to a nonlinear system.

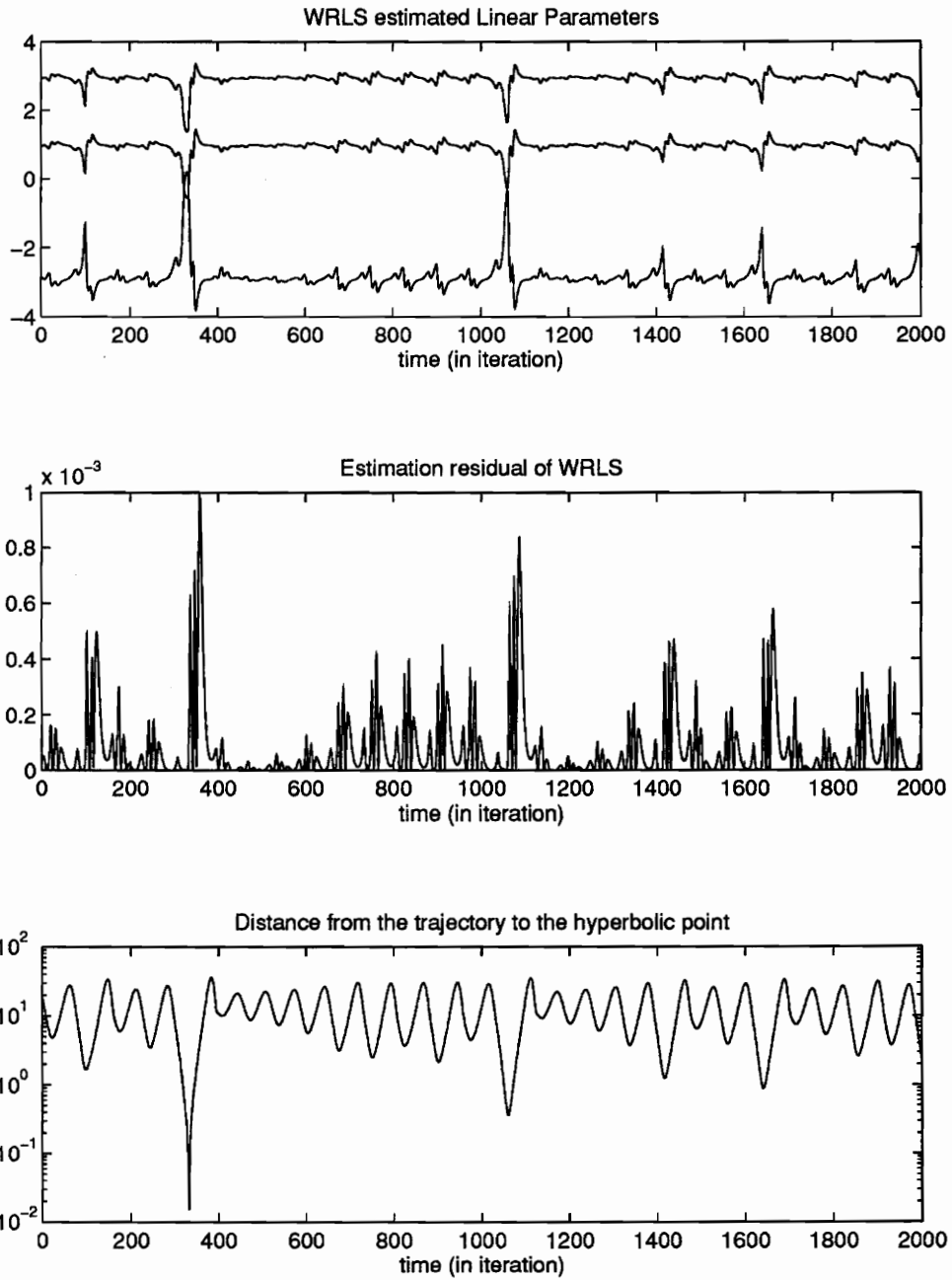
The same Lorenz trajectory was used to test the RBF nonlinear prediction model. The total number of basis functions used for modeling the Lorenz equation is 17 after the input vector space was sampled according to the EDSR. The variance of the resulting prediction error is  $1.8623 \times 10^{-6}$  (Figure 7.7b), or the relative residual power is  $2.6971 \times 10^{-8}$ , which is about 30 times lower than that obtained by the linear AR predictor.



**Figure 7.7 (a) Residual for WRLS prediction; (b) Residual for RBF prediction.**

To study how time-varying linear approximation works for a nonlinear system, such as the Lorenz dynamic system, we provide the history of the estimated linear model parameters, the residual power of the forward prediction and the distance from the dynamic trajectory to the hyperbolic point in Figure 7.8a-c respectively. The history of the estimated model parameters indicates that during most of the time the model parameters are nearly time-invariant, i.e. the dynamics of the unknown system act as a linear dynamic system. When the system is in a linear region, the residual is also small. Compared with the distance from the dynamic trajectory to the hyperbolic point, the system acts linearly when it is surrounded by one of the dynamic basins and far away from the repeller. When the trajectory approaches the repeller the nonlinearity increases. The model parameters show quick changes, indicating that the WRLS algorithm tries to adapt itself and to follow the dynamic change. Meanwhile, the residual increases. When the dynamic trajectory is very close to the hyperbolic point, and causes the dynamic switch from one basin into another, the residual power increases dramatically. This provides evidence that the nonlinearity of the Lorenz system is concentrated at the hyperbolic point. The closer we are to the unstable equilibrium point, the stronger the nonlinearity is. At a certain point, the WRLS linear approximation can not track fast enough, and the linear approximation error becomes large. Since the time for the dynamics to pass the hyperbolic point is only a short moment, the WRLS algorithm can catch the trajectory again after passing the unstable point and the average of the residual power returns to a low level.

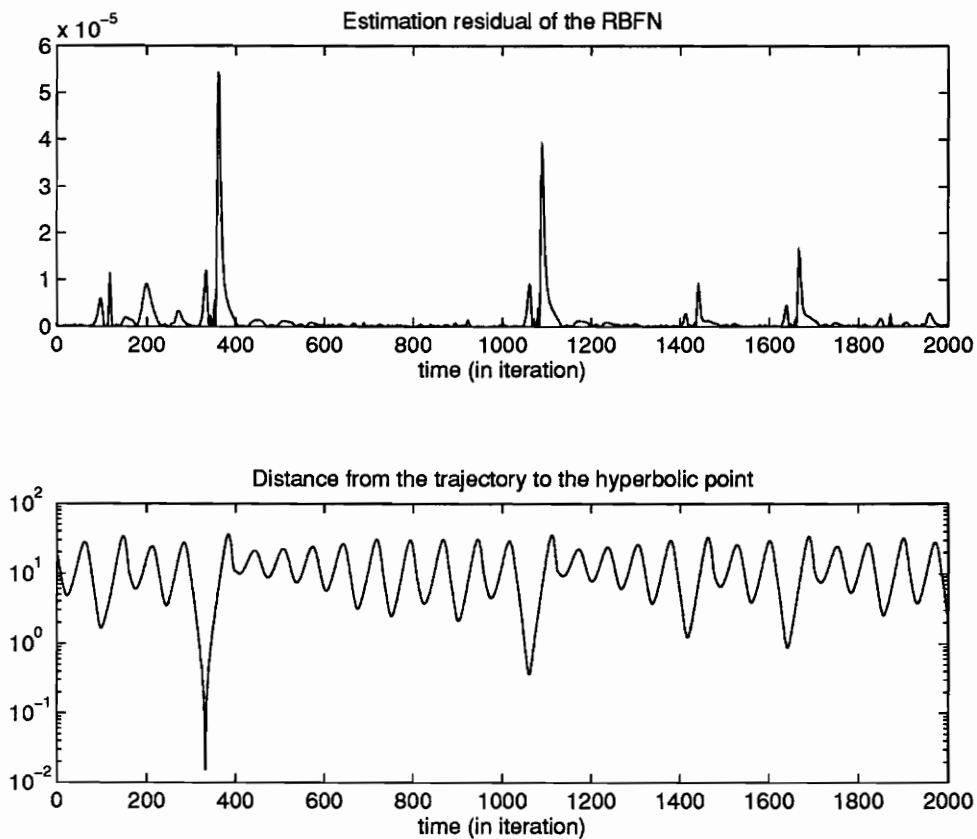




**Figure 7.8a-c. Relations of the linear coefficients, the residual power of WRLS and the distance to the unstable point.**

For comparison we also provide the residual of the RBF approximation together with the distance from the dynamic trajectory to the hyperbolic point (Figure 7.9a,b). The history of the RBFN residual shows that it can follow the Lorenz dynamic system much better than the linear

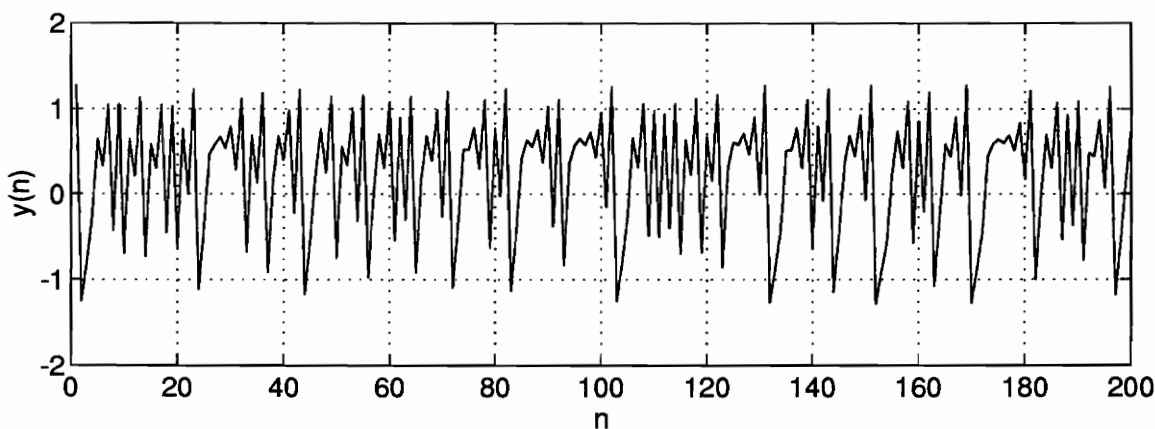
approximation. Still, the largest error occurs when the dynamics approach the unstable hyperbolic point. This reflects the nature of chaos. When the dynamics approach the repeller, the required precision for tracking or predicting the behavior of the chaotic system is infinite. Practically, in a small region of the phase space surrounding the repeller, the dynamic behavior of a chaotic system is unpredictable even when the dynamic equation is known exactly. The shorter lasting period and the lower level of residual power indicates that the RBFN can track much higher nonlinear dynamics than WRLS could.



**Figure 7.9a,b Relations of the residual power of RBFN and the distance to the unstable point.**

In the last computer simulation, a discrete chaotic dynamic Lorenz equation was used. We next use the dynamic function given by the Henon equation (7.5). The parameters selected in our

simulation are  $a=1.4$ ,  $b=0.3$ , and  $c=1$ , to place the Henon dynamic system in the chaotic region (Figure 7.10). The input vectors  $\mathbf{y}_k = [y_{k-1}, y_{k-2}]$ ,  $k = 1, 2, \dots, N$ , are plotted in the input vector space spanned by  $y_{n-1}$  and  $y_{n-2}$  (Figure 7.11). In physics, this input vector diagram is also called the *return map* of the Henon equation. In the computer simulation, a linear forward predictor was used to predict the discrete trajectory  $y_n$ . The orders used in the autoregressive model were 2, and 4. Both experiments yielded large prediction errors with relative residual powers of 0.7150 and 0.6368 respectively. The given results indicate that about 72% to 64% of the information in  $y_n$  is unpredictable by a piecewise linear estimator. Note that increasing the order of the linear predictor does not improve its performance much if the residual is mainly caused by nonlinearity.



**Figure 7.10 A Henon equation generated time series.**

Next, an RBF nonlinear predictor was used to analyze the dynamics of the Henon equation. The number of basis functions used was 11 (Figure 7.11). These basis functions constructed a nonlinear dynamic surface defined by the Henon equation (Figure 7.12). To compare the estimated dynamic surface with the ideal one, Figure 7.13 provides the cross sections of the estimated

dynamic surface (the solid lines), and the cross sections of the original dynamic surface (dashed lines; these are covered by the solid lines), i.e.

$$\begin{cases} y_n = 1 - 1.4y_{n-1}^2 & \text{the projection on } y_{n-2} = 0 \\ y_n = 1 + 0.3y_{n-2} & \text{the projection on } y_{n-1} = 0 \end{cases} \quad (7.11)$$

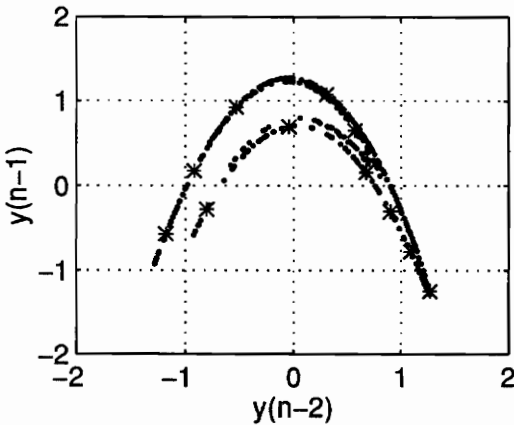


Figure 7.11 The input vectors ‘.’ and the selected centers for the RBF ‘\*\*’.

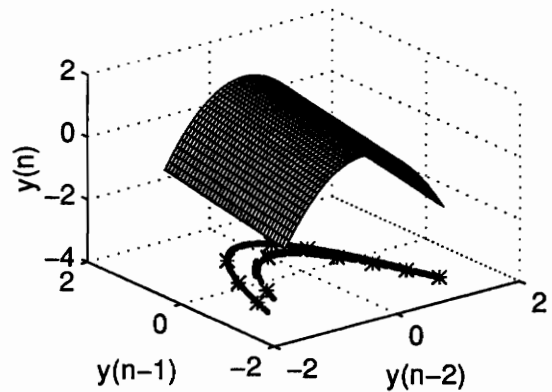
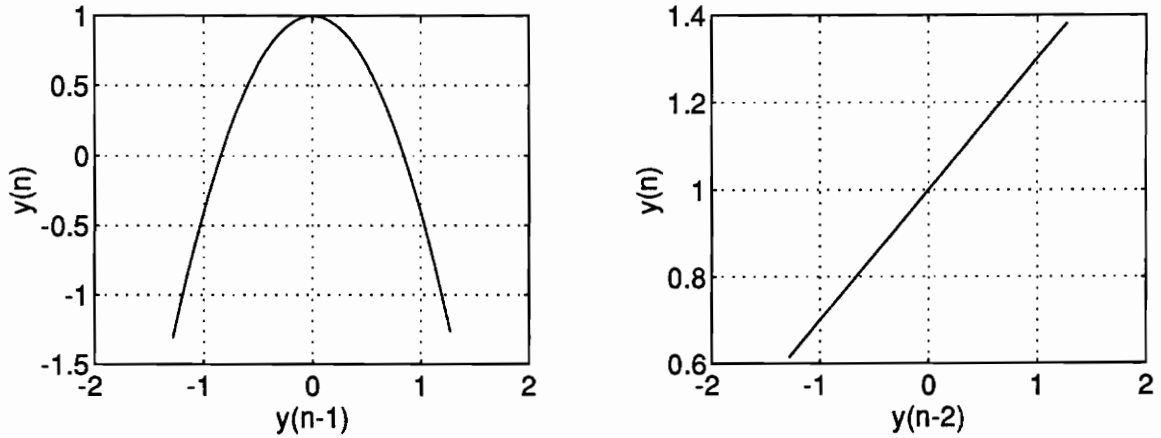


Figure 7.12 The reconstructed dynamic surface of Henon equation.

The results show a perfect match between the estimated and actual values. The variance of the residual is  $9.6623 \times 10^{-10}$ , or the relative residual power is  $1.9325 \times 10^{-9}$ , which reveals the fact that the deterministic Henon equation is perfectly predictable in the short term.

This final experiment demonstrates that if the tracking speed of the linear predictor can not be faster than the changes in the actual dynamics, the performance of the piecewise linear approximation is poor. In the example of predicting the Lorenz equation, where the dynamics are continuous, we could reduce the step-size of the Runge-Kutta algorithm (which is the same as increasing the sampling rate) to give the linear estimator enough time to follow the changes in the nonlinear dynamics. In the case of the Henon equation prediction, where the dynamics are discrete, the linear estimator could not track fast enough to follow its dynamic changes. In all three

experiments, the RBF decomposition method provided satisfactory performance. The results have shown that the RBF decomposition is not only good for nonlinear modeling, it can also be implemented in the linear situation.



**Figure 7.13a,b** The cross sections of the estimated (solid) and actual (dashed) dynamic surface.

#### 7.4 Implementation of the RBFN for biological data analysis

One of the most pervasive enigmas regarding brain function is how information is transmitted from one location to another. In view of the fact that nerve impulses *per se* are more or less all-or-none in character, attention has been focused on the pattern of interspike, i.e. inter impulse, intervals as the carriers of information. A good deal of evidence has accrued to the effect that in an anesthetized unstimulated brain, the interspike intervals recorded from single neurons in a variety of locations are essentially random in their distribution. Models of the interspike interval process have therefore been constructed based on the assumption that this randomness reflects a stochastic process. Stochastic resonance models [140-142] and stochastic resonance with noise models [143] have been especially fruitful in simulating data obtained from spike trains. However, recording randomness does not in itself insure that a process is stochastic. Recently a surge of interest has developed for the possibility that the behavior of spike trains, though random, could be generated

by a deterministic nonlinear process which results in chaos. This section sets out to test whether deterministically chaotic or stochastic processes best characterize the patterns of interspike intervals recorded from the hippocampus and the somatosensory cortex in the lightly anesthetized rat [144].

Deterministic chaos is defined as a process which can be described precisely by a deterministic dynamic function. This deterministic function generates an 'unpredictable' bounded stable state [138]. The 'unpredictability,' here, reflects the fact that the system is not periodic, quasi-periodic, or at equilibrium, i.e. converges onto a point attractor. The definition highlights two important considerations. The first is that the generator of the behavior is deterministic, even though the behavior currently displays randomness. The second consideration is that the observed randomness is not due to noise or interference from outside the system under observation; rather, the apparent randomness is due to the internal properties of the system, i.e. its internal dynamics. Furthermore, to be chaotic, its randomness must reflect the fact that the system is sensitive to initial conditions, small perturbations, and numerical errors caused by finite data length.

There is no widely accepted definition for stochastic processes. Any process will be called stochastic if its behavior is unpredictable from available past information. Instead of being generated by a deterministic function, the randomness of a stochastic system is due to possible interference from outside the system under observation, improper selection of observations, or lack of knowledge about its coding structure. This ambiguity must not be reducible by improving the precision of measurement and / or computation. For example, the trajectory of the orbit of Uranus was confusing before Neptune was discovered. The unpredictability in the trajectory of Uranus could not be reduced by just increasing the precision of measurement and computation. Once the interaction from Neptune was taken into account, the trajectory became predictable. Thus, the similarity between chaotic and stochastic systems is that both currently display random behavior; the major difference is that chaotic randomness is due to an internal deterministic generator while stochastic randomness is not.

Furthermore, as a result of the operation of a deterministic generator, the trajectory of a chaotic process is structured while that of a stochastic process is not. The structure imbedded in the trajectory will therefore be reflected in its prediction error (residual). Ideally, this error reduction is unbounded when the frequency and precision of measurement and computation are improved. This

unbounded characteristic of the reduction of the residual is one of the most important measurements characterizing the chaotic process. In a stochastic system the variance of the residual will be much larger than that of the computational error, and can not be significantly reduced by increasing the rate of measurement and improving the prediction techniques.

In the previous sections, both linear and nonlinear methods have been delineated for testing linear stochastic and nonlinear chaotic processes. The computer simulations showed these methods to be promising in modeling nonlinear chaotic processes. The structure embedded in the prediction error was shown to be especially useful in understanding the dynamic structure of the unknown system. This structure can, then, be further used to discriminate a stochastic process from chaotic dynamics. In this section, these techniques will be implemented in an analysis of interspike intervals. The experiments are designed to model the interspike interval both by linear and nonlinear methods. Two questions are to be answered. First, whether or not the interspike interval can be modeled as being generated by a deterministic generator. Such a model would show (or not show) a short term predictability of the unknown process. Second, would such a model be linear or nonlinear? Only those 'randomly behaving' processes which proved to be nonlinear and short term perfectly predictable would be classifiable as chaotic.

The raw data were collected from anesthetized (with barbiturate) rat somatosensory cortex and hippocampus. The records are of spontaneously firing neural units (there is no stimulus), from 12 rats. Twenty three units were obtained from 7 hippocampal recordings; 12 units were obtained from 5 somatosensory cortex recordings. Figure 7.14a gives a typical interspike interval record from the somatosensory cortex. The mean value of the interspike intervals is 115 ms, with a standard deviation of 86.65 ms. The interspike intervals appear to occur randomly but they also show a tendency to group at 50 to 125 ms. However, about every 15 to 20 intervals, the interval will suddenly increase to the level of 300 to 400 ms, and oscillate a few times before returning to the lower base line. The autocorrelation functions (Figure 14b) at lags 1 and 2 are higher than the 95% confidence level. There are also some seasonal changes although these fall below the 95% confidence level. The partial correlation function [5] (also referred to as reflection coefficients in parametric spectral analysis) drops abruptly at lag >1, and it approximately equals the 95% level at lag 17.

A seasonal autoregressive model was constructed from one recording of the somatosensory spike-trains according to the autocorrelations and partial autocorrelations. The model equation is

$$(1 - a_1 z^{-1})(1 - a_2 z^{-17})y(n) = e(n) \quad (7.12)$$

where  $z^{-k}$  is a  $k$  step delay operator,  $e(n)$  is the excitation noise of the process or residual of the prediction, and  $[a_1, a_2]$  are the unknown parameters of the model. The estimated results are

$$(1 + 0.5092z)(1 + 0.2924z^{-17})y(n) = e(n) \quad (7.13a)$$

or

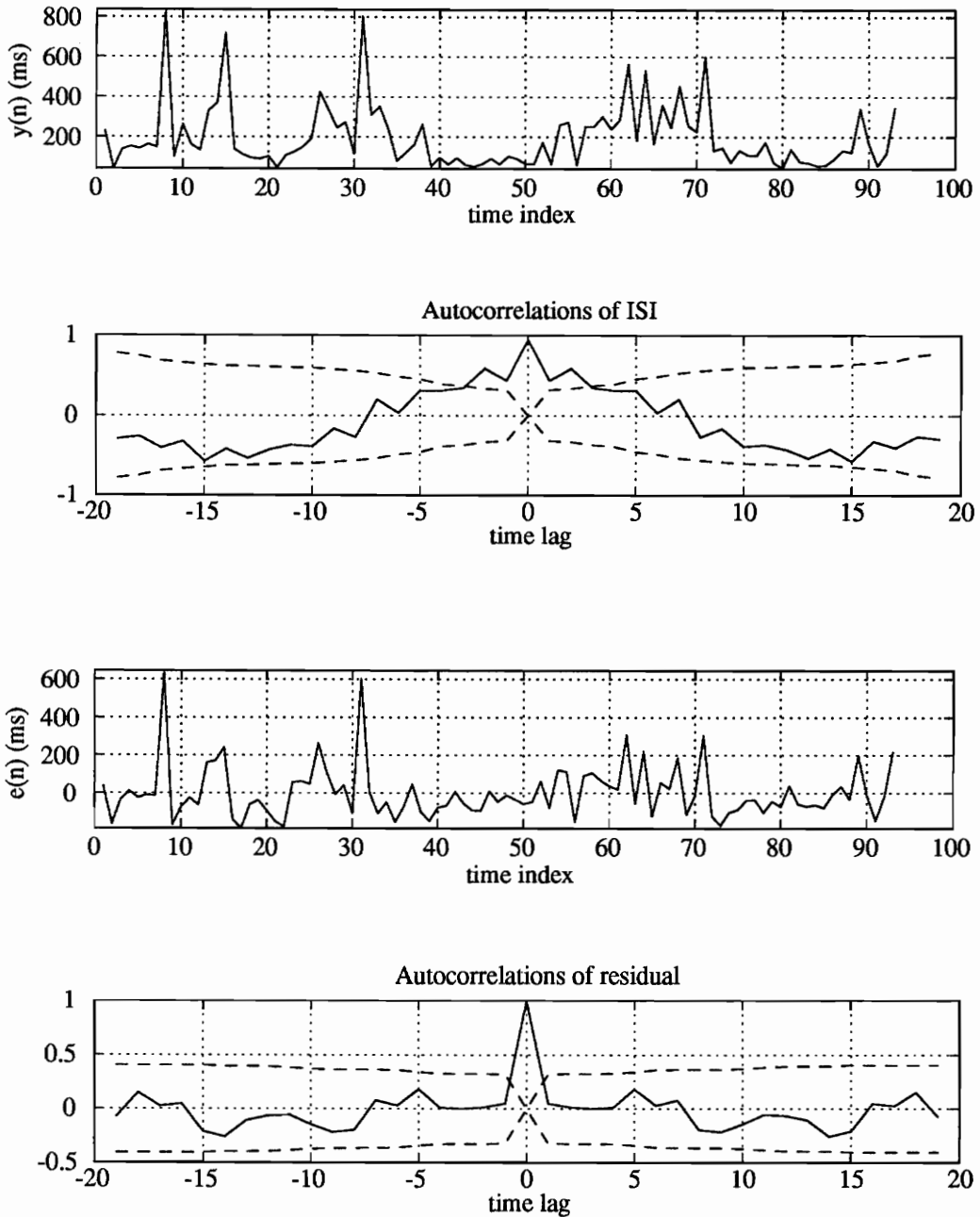
$$y(n) - 0.5092y(n-1) - 0.2924y(n-17) + 0.1489y(n-18) = e(n) \quad (7.13b)$$

Figures 7.14c and 7.14d give the one step forward prediction error  $e(n)$  and its autocorrelation function. The variance of the residual is 5425.9 (the standard deviation = 73.66). All the autocorrelation values of the residual are below the 95% confidence level at lags away from 0. Meanwhile, the T ratio test for the estimated parameters,  $a_1$  and  $a_2$ , in (7.12) are 5.87 and 2.7 respectively. To test the stationarity, a sliding window of 100 sample points was applied to the interspike intervals; the estimated model parameters stay at almost the same values as the window moves across the records, which indicates the interspike intervals are stationary.

Next, the nonlinear radial basis function estimator was applied to the same record. Fourteen bases were used in reconstructing the recorded interval train; the variance of the residual, and its autocorrelations were identical to those obtained with the linear estimator.

The rest of the interspike intervals recorded from the somatosensory cortex were processed using the same techniques. Most can be modeled as first order autoregressive processes plus some seasonal effect at lag intervals of 7 to 17. Only 3 out of the 12 interspike interval records from the somatosensory cortex can be modeled as a first order autoregressive model (AR(1)) or first order autoregressive moving-average model (ARMA(1,1)) without a seasonal period. The relative variance of the residual and the original interspike intervals was 0.914 in the linear prediction cases and 0.903 in the nonlinear cases, which means that less than 10 percent of the information in the interspike interval can be predicted according to previous observation.

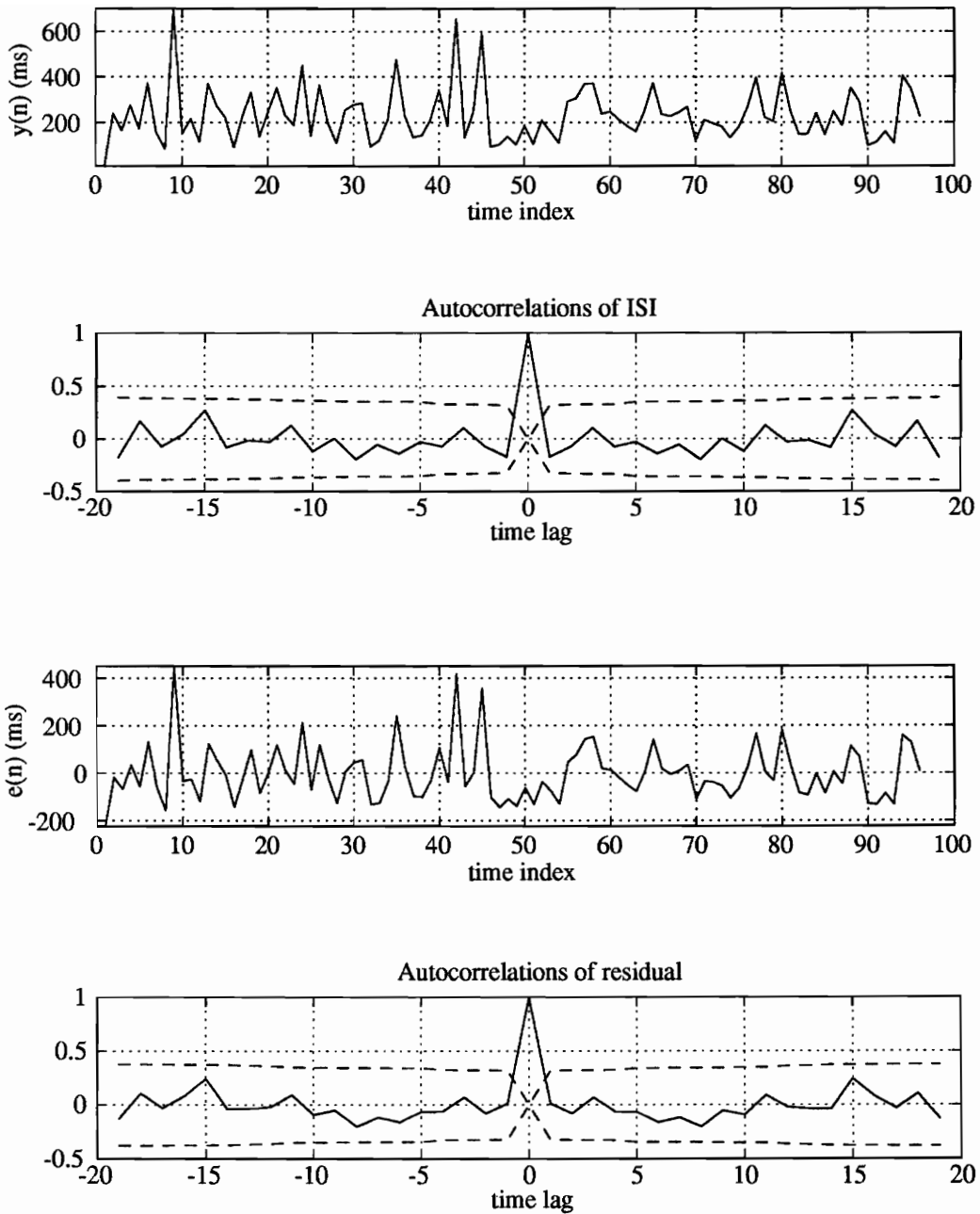




**Figure 7.14.** (a) An example of interspike intervals recorded from the somatosensory cortex; (b) estimated autocorrelation coefficients of interspike intervals (solid line), and  $\pm 2$  times standard deviation of the estimation (dashed lines); (c) the forward prediction residual of interspike intervals; (d) the estimated autocorrelation coefficients of the residual (solid line) and  $\pm 2$  times standard deviation of the estimation (dashed lines).

The results indicate that the interspike intervals of the somatosensory cortex can be modeled as a time-invariant (stationary) linear autoregressive process under the condition of no stimulation. Only a very small fraction of the information at any interval can be predicted from previous intervals. Such a process is not chaotic, it is stochastic.

Figure 7.15a gives an example of an interspike interval record from the hippocampus. As some units (In neural science, unit is a term for neural spikes of the same shape, which are considered to be generated by the same neuron) fire in bursts, only the first spike in a burst was sorted, so that the interspike interval becomes the inter burst interval (assuming that the unit is related to the burst). The same prediction methods as those used for analyzing the recordings from the somatosensory cortex were applied to the 20 different units obtained from the hippocampal records. The typical interburst interval autocorrelation function (Figure 7.15b) and partial correlation functions vanish abruptly at lag 1 or 2, which indicates almost white noise. No seasonal tendency is found in any of the hippocampal records. The first or second order autoregressive processes provide excellent models for the interspike interval processes. The autocorrelations of the prediction errors also point to white noise. The relative variance of the residual and the interspike intervals *per se* are both about 0.974 for the linear estimator and 0.967 for the nonlinear one. These results indicate that the hippocampal interspike intervals are even more unpredictable than those recorded from the somatosensory cortex. Both are stochastic process.



**Figure 7.15.** (a) An example of interspike intervals recorded from the hippocampus; (b) estimated autocorrelation coefficients of interspike intervals (solid line), and  $\pm 2$  times standard deviation of the estimation (dashed lines); (c) the forward prediction residual of interspike intervals; (d) the estimated autocorrelation coefficients of the residual (solid line) and  $\pm 2$  times standard deviation of the estimation (dashed lines).

## **7.5 Experiments in structure adaptation for the radial basis function network.**

### **7.5.1 Overview of structure adaptation**

The most common networks, such as most of the ANNs and all the GMDH networks, are topological structure based networks. That is, the network establishes a relational map for the nodes (neurons) in the network with a set of network parameters or weighting coefficients. There is no geometric meaning of neighboring or distance between the nodes. Two nodes will be considered as neighboring each other if there is a weighting coefficient between them; geometrically they can be located at opposite sides of the network. Nodes will not be considered to be neighbors as long as there is no weighting coefficient directly between them.

One major problem associated with the topological structure based network is that knowledge in the network can not be accumulated. That is, the network can not be trained gradually during its operation to reach a proper structure and set of network parameters. Instead, all the possible input-output pairs have to be obtained before the training, and used during the training period. Due to its topological structure, any modification in the structure or its weighting coefficients during later network operation will cause unpredictable results. Thus, these kinds of networks are not adaptable once they have been trained. Any modification or increase in the knowledge base (new relations of the input-output pairs) will necessitate retraining of the network. To preserve the original network function, the retraining procedure must use all the old training data together with the new input-output pairs, which means that the network has to be completely reconstructed based on all the input-output pairs. The more target events (outputs) are to be learned or to be discriminated simultaneously, the more sophisticated the problem is, and the more local minima will be encountered in the merit function for solving the optimal network coefficients. This is the main limitation to the learning of any complex problems by a topological structure based network.

The RBFNs offer the possibility of training the network gradually. Due to its locally supported basis functions, the entire input space can be partitioned into segments. Each segment will be dominated by the basis function defined in the corresponding region. Therefore, the RBFN can learn one bit of knowledge after the other without the necessity of retraining the entire network. Moreover, by learning gradually, the RBFN can possibly be used as a structure-adaptive signal

processor, in which the center locations and radii of the basis functions could be updated dynamically, i.e. as the system under observation changes.

The main issue related to the structure adaptation of the RBFN is the adaptation of the free parameters during operation. The free parameters are the weighting coefficients, the dilations, the central locations and most importantly, the number of basis functions. The adaptation in the number of basis functions provides the capability for the network to grow itself based on its operating environment; the central location adaptation reallocates the RBFs to fit the hypersurface better; controlling the dilations of the RBF is important, as it repartitions the input space with the basis functions. So far these adaptations or the structure adaptation of the network have not been addressed by existing RBFN construction methods, nor by ANN construction techniques.

The structure adaptation and adaptive learning is a long term research goal. The purpose for this section is to explore the potentials and possible problems related to the idea with some simple experiments. To demonstrate the possibility of structure adaptation for the RBFN, the experiments are designed to train the network to learn the relation between input-output pairs one at a time, such that the RBFN gradually learns an unknown nonlinear function in a given dynamic region. To simplify the design, we limit the input space to being one dimensional, i.e. the network is trained to approximate a one dimensional function; a proper constant value is added to the function to make the function value larger than zero in the input region, so that all the weighting coefficients are positive. In real applications, such a bias can be removed from the estimates of the RBFN. The structure adaptable RBFN is named an ARBFN in the rest of the section.

The general idea of structure adaptation for an RBFN is to adaptively adjust the network parameters. The key parameters for the adjustment are the dilations for the RBF. Instead of using a fixed value for all of the basis functions, in an ARBFN the dilations in different directions of different basis functions can be modified independently. When the network can not correctly estimate the output value given the input, the first parameter set to be adapted is the dilation set. By spreading and contracting the RBFs, the network reshapes its performance surface; the tails of some of the basis functions will reach into unoccupied regions, and for others their sizes may be reduced. Since each basis function is locally defined, only those next to the new input experience a large influence. The error of the reshaping is measured as the mean square error for all the input-output pairs that have been learned by the ARBFN. If this reshaping process can not reduce the

error at the new input-output pair to below a threshold level, a new basis function will be put at the new input location because the performance surface has a ‘hole’ there. This learning process can be in parallel with network operation until no more holes are left within the dynamic input region. During operation, after the dilation set has been tuned, the weighting coefficient set and the center locations of the RBFs are tuned to optimal values (small local adjustment) in succession. The LMS algorithm is used for all the parameter adaptations.

### 7.5.2 RBFN adaptation

The ARBFN is initialized from very few input-output pairs as a training data set, such as a single input-output pair,  $x_1 \leftrightarrow y_1$ ; i.e. the initial ARBFN is constructed with one radial basis function centered at the input location  $x_1$ . The only weighting coefficient is selected as the value of the output  $y_1$ . The dilations of the basis function at the left side and right side are named  $\sigma^L$  and  $\sigma^R$  respectively (note that the superscripts  $L$  and  $R$  indicate the dilation  $\sigma$  at the left or right side); both of these are positive with initial values arbitrarily chosen, such as  $\sigma^L = \sigma^R = 0.01$ . The network function with a single basis function can thus be represented as:

$$\hat{y} = \begin{cases} y_1 e^{-\frac{(x-x_1)^2}{\sigma^R}} & \text{for } x - x_1 > 0 \\ y_1 e^{-\frac{(x-x_1)^2}{\sigma^L}} & \text{for } x - x_1 < 0 \end{cases} \quad (7.14)$$

When the input  $x = x_1$ , the output of the network  $y = y_1$ , i.e. the network has learned to perform the unknown function at this position. When the input  $x$  is different from  $x_1$ , the output value will be estimated by the exponential function. The error for the estimation depends on how close the exponential function can fit the unknown function. There is no need to solve any merit function in constructing an initial RBFN with a single basis function.

After the initial ARBFN is built, the network starts its operation. When a new input-output pair  $x_2 \leftrightarrow y_2$  is presented, the ARBFN estimates the output based on its current knowledge by using (7.14). If the estimation error  $|y_2 - \hat{y}_2|$  is less than a given threshold, the ARBFN will consider its present knowledge adequate, and make only a small *local* modification to its parameter set to

improve the estimate, such as the dilations, weighting coefficient and the center location. If the network estimation error at  $x_2$  is larger than the threshold, the relation of  $x_2 \Leftrightarrow y_2$  is considered unknown and a new *neuron* will ‘grow’ at location  $x_2$ . That is, a new RBF center will be located at  $x_2$ , ( $c_{new} = x_2$ ); the initial weighting coefficient is chosen as  $0.625y_2$ , or 62.5% of the function value at  $x = c_{new}$  is contributed by the RBF at this location and the rest of the function value comes from its neighboring RBFs. With the new RBF, the parameters of the ARBFN will be adjusted by using the LMS algorithm. Since RBFs are locally supported, only the new *neuron* and its immediate neighbors need to be fully adjusted, the rest of the network requires only a minor modification. The adaptation in the ARBFN starts by adjusting the dilation coefficients along different directions, then tuning the weighting coefficients and the center locations in succession. Gradually, the ARBFN learns the unknown function in the given region.

Let  $M$  be the number of basis functions for the present ARBFN, the adaptation of the dilation parameter set is performed based on the following procedure:

The right dilation parameter set  $\{\sigma_n^R: n = 1, 2, \dots, M\}$  is updated at the  $i$ -th LMS iteration by

$$\sigma_n^R(i) = \sigma_n^R(i-1) - \frac{\mu}{M_n^R y_n} \sum_{m=1}^{M_n^R} (\hat{y}_m - y_m) \quad n = 1, 2, \dots, M \quad (7.15a)$$

The indices of the RBFs are organized from left to right on the axis, that is, the center location  $c_1$  corresponding to the output  $y_1$  is the smallest one (on the left side of the axis), and  $c_M$  corresponding to  $y_M$  is the largest one (on the right side of the axis) among all the center locations of the RBFs. In (7.15a)  $M_n^R$  is the number of observed input-output pairs located between the  $n$ -th basis function and its immediate neighbor to the right, the  $(n+1)$ -st basis function (inclusive of this immediate right side neighbor). Note that the error used for adjusting  $\sigma_n^R$  is the mean error of the ARBFN from the region to the immediate right, since changing the right side dilation of the  $n$ -th basis function will mostly affect the performance of the network in that region. Using only the error in its immediate right neighboring region can greatly help the LMS algorithm to converge. This error is normalized by the output  $y_n$ . In the experiments presented later, a fixed size buffer for each region between the basis functions is used to store those most recent input-output pairs fallen into the corresponding region. Like the time window, a large size buffer provides stable estimates and a

small buffer re-shapes the performance surface of the network fast while the nonlinear system under modeling is time-varying. In (7.15a) the constant  $\mu$  is the step size of the LMS algorithm.

The left side dilation parameter set  $\{\sigma_n^L: n = 1, 2, \dots, M\}$  is updated in the same way,

$$\sigma_n^L(i) = \sigma_n^L(i-1) - \frac{\mu}{M_n^L y_n} \sum_{m=1}^{M_n^L} (\hat{y}_m - y_m) \quad n = 1, 2, \dots, M \quad (7.15b)$$

where  $M_n^L$  is the number of input-output pairs located between the  $n$ -th basis function and its neighbor to the immediate left, the  $(n-1)$ -st basis function (inclusive of this immediate left side neighbor). The error used for adjusting  $\sigma_n^L$  is the mean error of the estimates of the ARBFN in the region to its immediate right.

Figure 7.27 provides an example of constructing the performance curve by adapting the dilations of the RBFs. The unknown function is a linear function,  $y = x + 2$ . The dynamic input region is from -1.0 to 1.0. 10 RBFs are chosen; their center locations are equally distributed from -0.9 to 0.9 with an increment of 0.2. The weighting coefficients are pre-selected as

$$w_i = f(c_i) \times 0.625 = (c_i + 2) \times 0.625 \quad i = 1, 2, \dots, 10 \quad (7.16)$$

The initial dilations equal 0.01. Figure 7.27a is the learning curve of the LMS algorithm; Figure 7.27b provides the dilations of the basis functions. Figure 7.27c is the performance curve of the network (solid line) superimposed with the ideal linear function (dashed line). The results show a good match between the network estimates and the ideal linear function.

After the dilations of the RBFs have been adjusted, the weighting coefficients and the center locations of the basis functions can be adapted by the LMS algorithm as well. Since the performance of the ARBFN is close to the ideal one, the adaptations of the weighting coefficients and the center locations are very small (using smaller step size  $\mu$  than that used for adapting dilations). The weighting coefficients are adapted, at  $i$ -th iteration, by

$$w_n(i) = w_n(i-1) - \frac{\mu}{M_n y_n} \sum_{m=1}^{M_n} (\hat{y}_m - y_m) \quad n = 1, 2, \dots, M \quad (7.17)$$



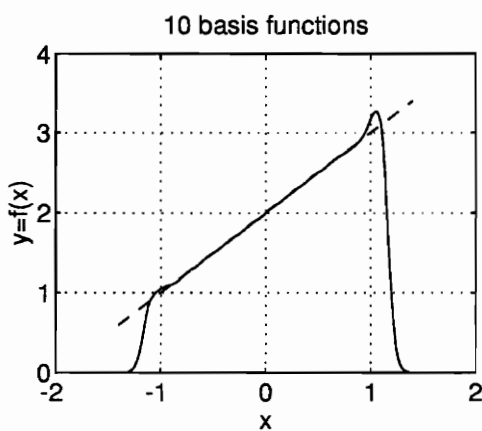
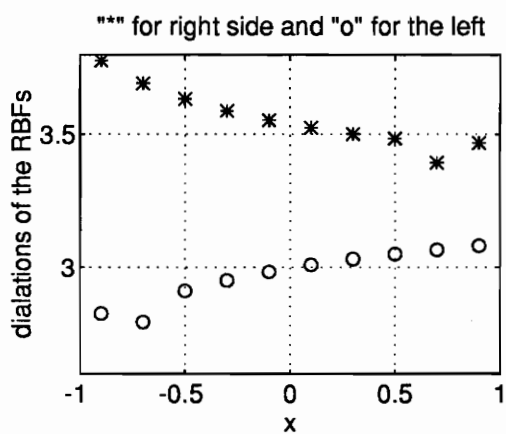
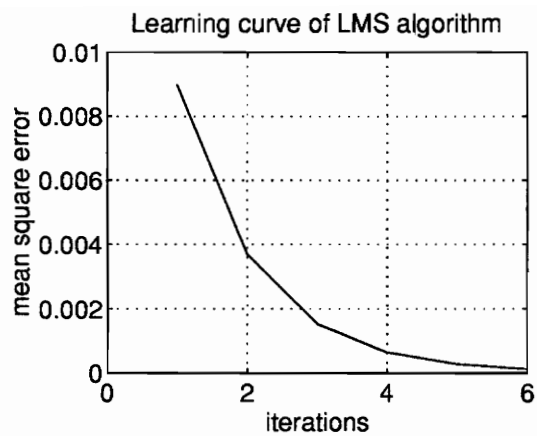


Figure 7.27a,b,c RBFN construction via dilation parameter adjustment.

where the mean error is computed from both the immediate right and left side regions of the  $n$ -th basis function including  $x = c_n$ , i.e. the total number of observed input-output pairs in the immediate neighborhood.

$$M_n = M_n^L + M_n^R + 1 \quad (7.18)$$

The center locations are adapted at the  $i$ -th iteration by

$$c_n(i) = c_n(i-1) - \frac{\mu}{M_n^L y_n} \sum_{m=1}^{M_n^L} (y_m - \hat{y}_m) - \frac{\mu}{M_n^R y_n} \sum_{m=1}^{M_n^R} (\hat{y}_m - y_m) \quad n = 1, 2, \dots, M \quad (7.19)$$

### 7.5.3 Experiments for ARBFN

A random number generator is used in the experiments (Figure 7.28), with a PDF that is uniform in the region between -1.1 and 1.1. 4 different functions are used for the simulations. They are the sine function, a linear function, an exponential function, and an exponentially damped sine function, i.e.,

$$\begin{aligned} y &= f_1(x) = x + 2 \\ y &= f_2(x) = e^x \\ y &= f_3(x) = \sin(x) + 1.5 \\ y &= f_4(x) = e^{-x} \sin(x) + 3 \end{aligned} \quad (7.20)$$

For the linear and exponential functions, after 20 basis functions have been selected, the ARBFN estimates the unknown function values within the input region [-1.1, 1.1] with less than 0.1% mean square error (Figure 7.29a,b). For the sine and exponentially damped sine functions, 25 basis functions are needed for the same level of performance (Figure 7.29c,d). Some examples of the learning procedures in progress are presented in Figure 7.30a-f for the linear function and in Figure 7.31a-f for the sine function.

In these experiments, the RBFN always converged to the optimal solution after less than 26 RBFs have been selected. For each new RBF added to the ARBFN, the adaptation takes place in

about 50 LMS iterations. The computation in flops is given in Table 7.1. The Artificial Neural Network with back propagation was used to identify the same functions. The results are provided in Figure 7.32. For the linear and exponential function, within less than 2,000 epochs (defined as the number of the iterations of the backpropagation algorithm), the ANN converges to the solution, while for the sine and exponentially damped sine functions, after 10,000 epochs, the ANN has not converged to the solution. Certainly, the back propagation network has no capability for adaptation or knowledge accumulation. Table 7.1 provides the computation count and the mean square errors of the ARBFN and ANN.

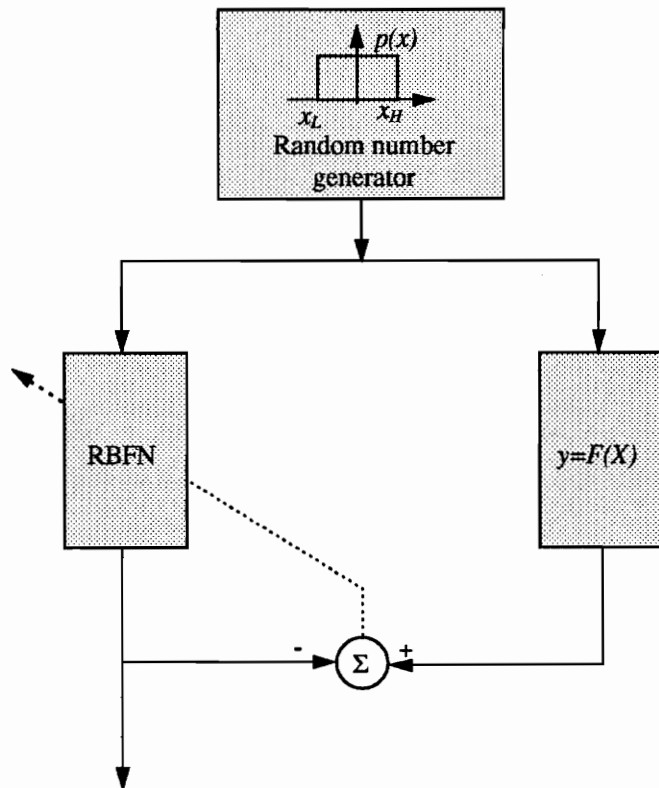
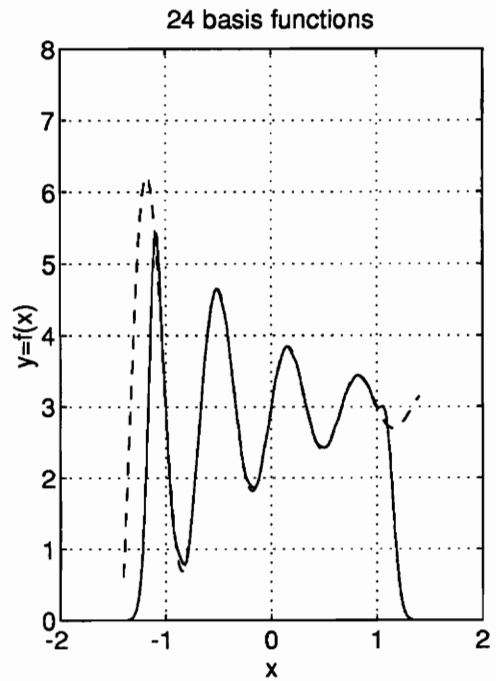
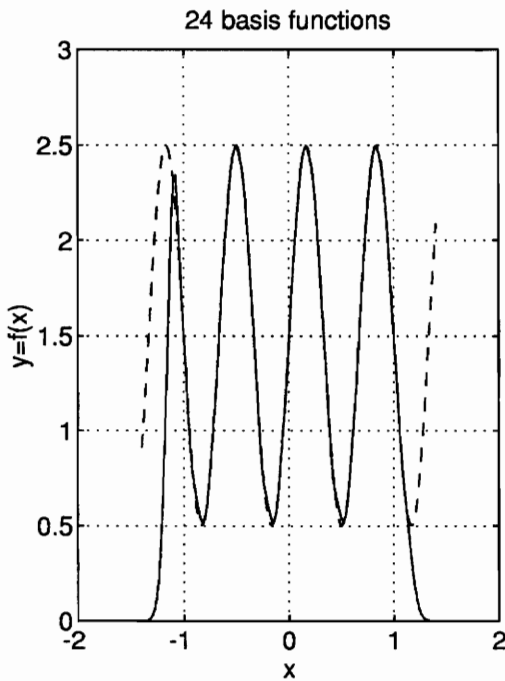
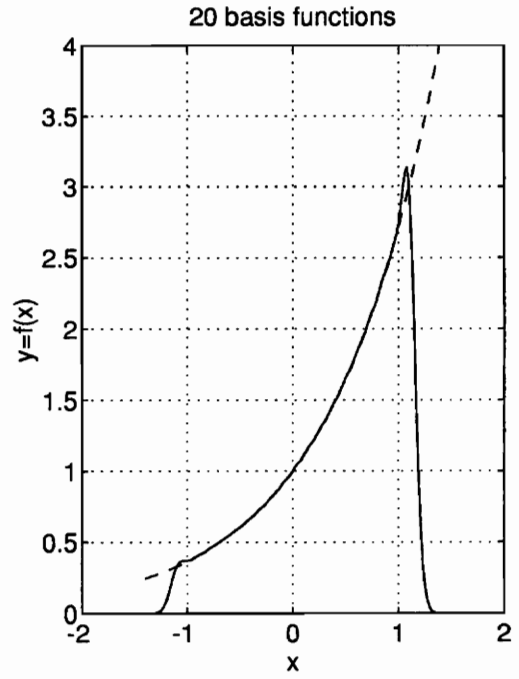
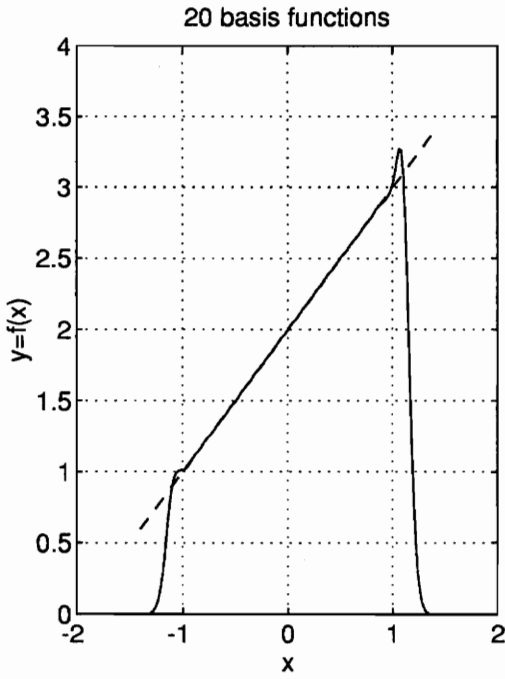


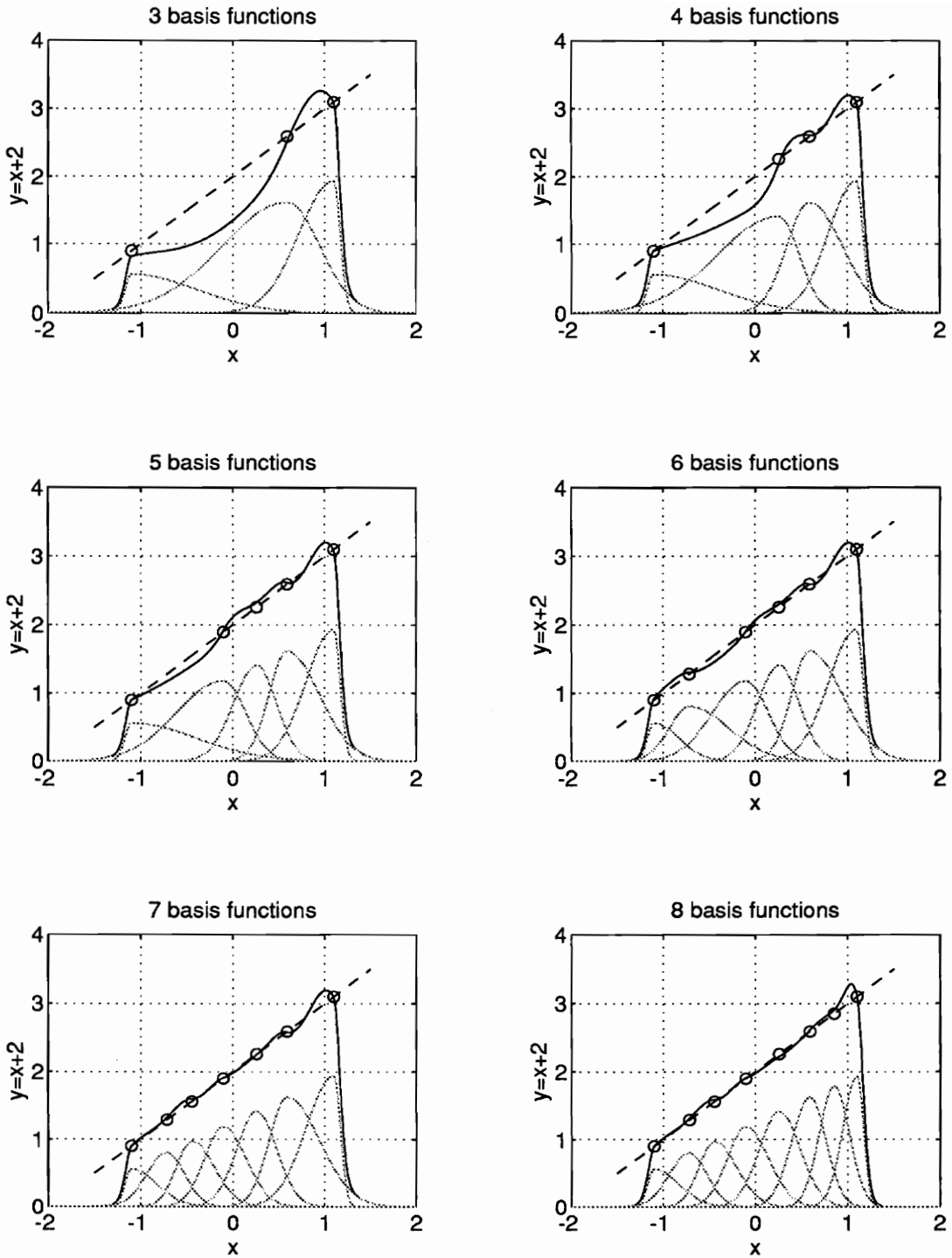
Figure 7.28 ARBFN training diagram.

**Table 7.1 Comparison between the ARBFN and the ANN.**

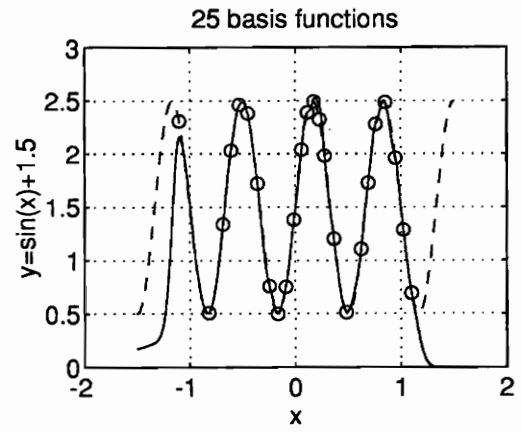
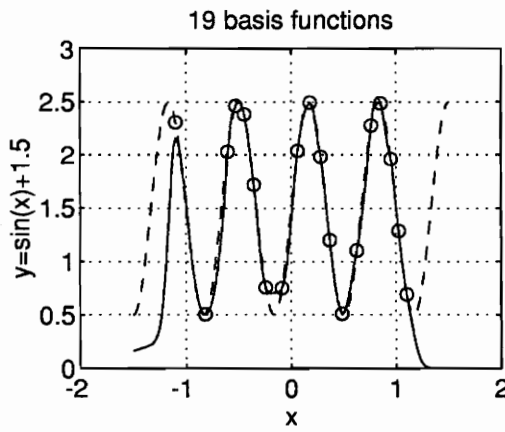
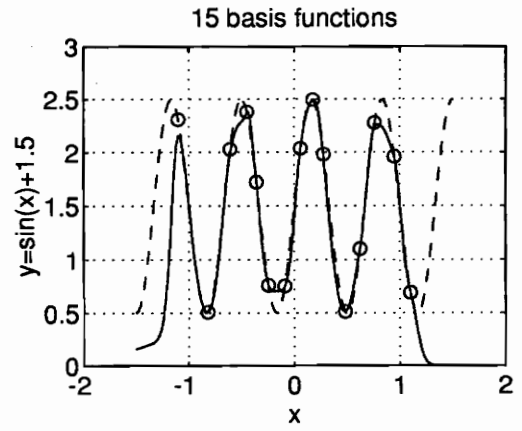
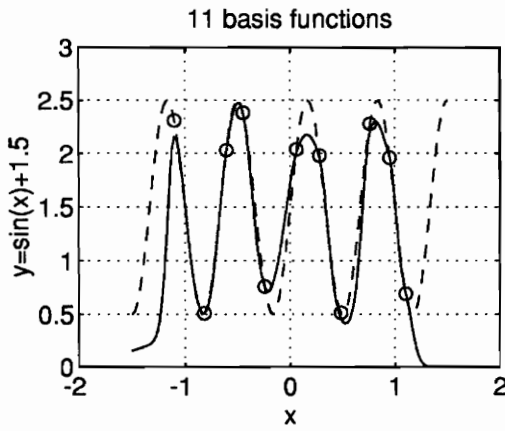
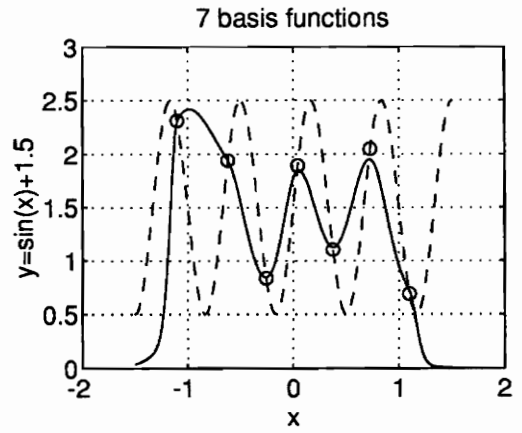
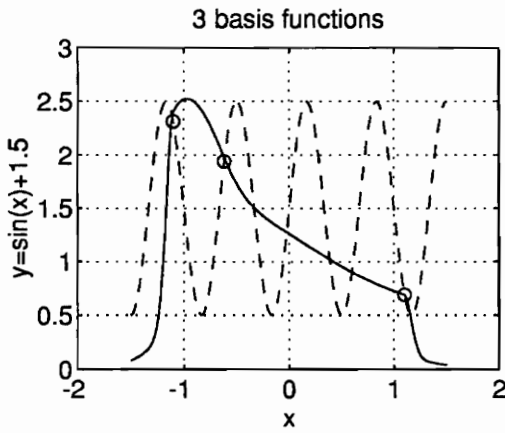
	ARBFN		ANN	
	$\sigma^2$	flops	$\sigma^2$	flops
$y = x + 2$	$1.7618 \times 10^{-5}$	376,378	$4.2000 \times 10^{-3}$	1,425,769
$y = e^x$	$1.9533 \times 10^{-5}$	344,963	$7.0261 \times 10^{-4}$	2,056,644
$y = \sin(x) + 1.5$	$8.9401 \times 10^{-6}$	385,125	N/A	N/A
$y = e^{-x} \sin(x) + 3$	$1.1000 \times 10^{-3}$	412,311	0.1767	30,882,681



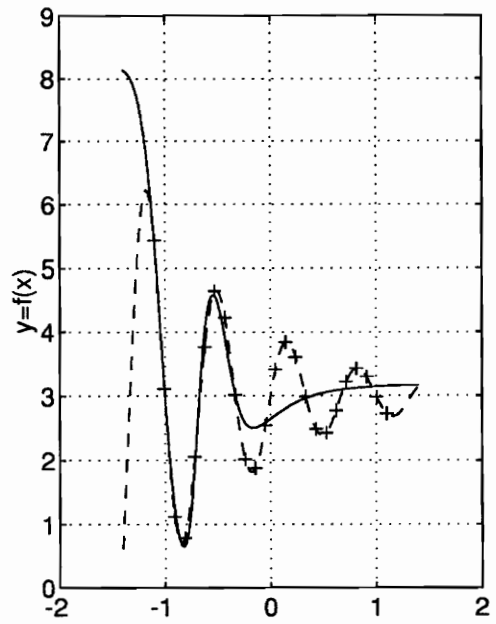
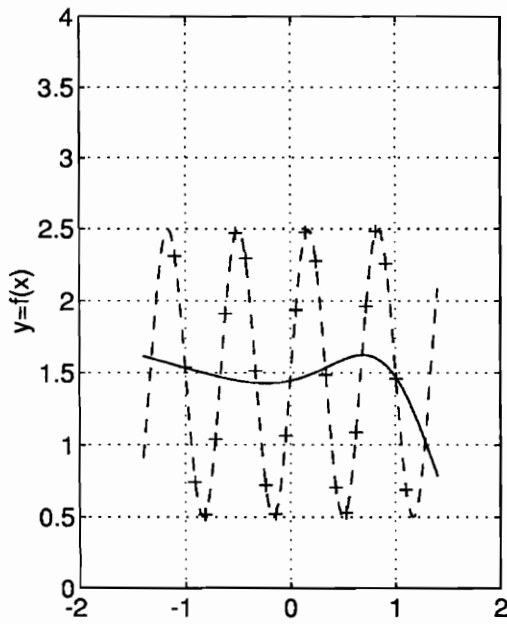
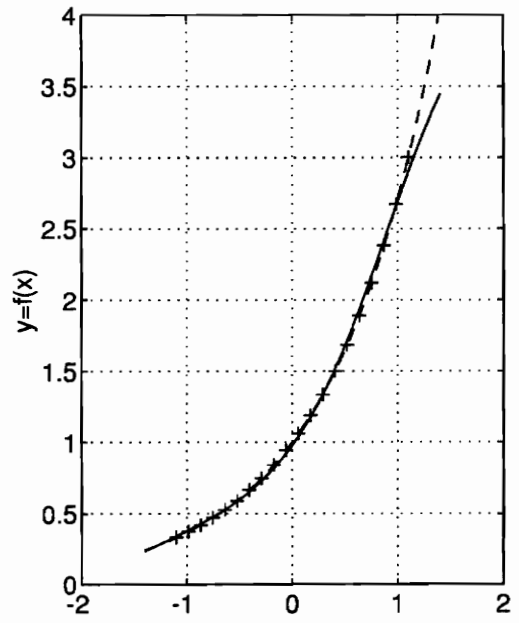
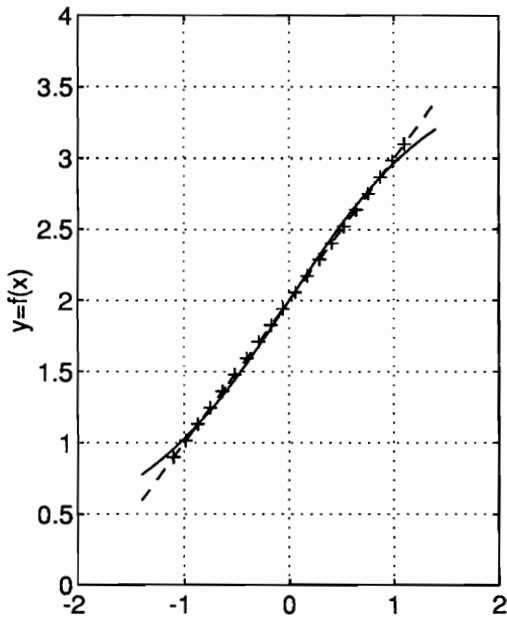
**Figure 7.29** Performance of the ARBFN in approximation (the dashed line is the ideal function; the solid line is the ARBFN response curve).



**Figure 7.30 Snap shots of the ARBFN learning process ('o' indicates the RBF center locations; the solid line is the ARBFN function; dotted lines are the responses of each RBF in the network; the dashed line is the ideal function).**



**Figure 7.31 Snap shots of the ARBFN learning process ('o' indicates the RBF center locations; the solid line is the ARBFN function; the dashed line is the ideal function).**



**Figure 7.32 Performance of the ANN in approximation ('+' indicates input-output pairs used for ANN training; the solid line is the ANN function; the dashed line is the ideal function.**



## 7.6 Summary

In dealing with 'time-varying' processes caused by the linearization of a nonlinear system, the RBF modeling technique has shown better performance than piecewise linearization techniques. An equal distance sample rule is proposed for constructing the RBFN. Experiments indicate that the RBFN is a promising method for modeling deterministic chaos as well as stochastic processes, be it linear or nonlinear. The RBFN also outperforms the topology based Artificial Neural Networks (ANNs) in learning speed, computational simplicity, and general performance. The 'local' to 'global' approach of the RBFN provides great potential to overcome the biggest difficulty of ANNs in structure adaptation, self-learning and knowledge accumulation.

## 8. SUMMARY AND FUTURE RESEARCH DIRECTIONS

This dissertation research pertains to two signal decomposition methods. One decomposes the signal in the time-frequency phase space. The Multiresolution Parametric Spectral Estimation (MPSE) approach is proposed as an effective method for processing nonstationary signals. The other decomposition method is the Radial Basis Function (RBF) for nonlinear function approximation, which decomposes the signal in the phase space (or state space) spanned by the derivatives of different degrees.

For the time-invariant case, an advantage of MPSE, over fixed resolution AR spectral estimation, is that a high order AR process can be decomposed into a number of lower order AR processes; this improves the statistical performance of the estimation since there is less cross-talk between the component processes. Also, MPSE can be used for processes which contain a mix of broadband and narrowband frequency components, whereas a fixed resolution AR estimator does not perform well in that case. For the time-varying case, MPSE provides much better frequency tracking capability than the conventional AR estimator. It also improves frequency resolution, by increasing SNR in each subband, and time resolution, since lower order processes require shorter data window widths.

In comparison with the wavelet transform, MPSE provides better time and frequency resolution. The MPSE advantage of providing the flexibility of selecting different time windows in different frequency bands contrasts with the wavelet transform where the window width at different frequencies is determined by the non-arbitrary choice of basis function. In addition, MPSE is computationally simpler to implement.

In the resolution study, a virtual transformation function is used to derive the time characteristics of AR estimators. The mathematical representations for time-frequency resolution are provided for windowed Fourier transforms, wavelet transforms, AR spectral estimators, and MPSE. Both the theoretical and the experimental results show that MPSE is the best in resolution of all the above techniques. Furthermore, under certain conditions, MPSE can yield better resolution than the lower resolution bound defined by the Heisenberg uncertainty principle. This super high resolution is obtained by using *a priori* knowledge such as the structure and order of the dynamic signal model.

By trading-off magnitude with time-frequency resolution, MPSE can yield finer resolution in the time-frequency space.

Three areas of application for MPSE illustrate that the newly developed method is good for nonstationary signal processing including de-noising (optimal filtering), tracking and analyzing of time-varying signals (EEGs), and enhancing other signal processing operations by providing time-varying information about the signal (optimal wavelet design). The high resolution and good noise immunity lend the MPSE algorithm great potential for the nonstationary environment, especially for those applications where parametric models are appropriate for the time-varying signals.

In dealing with 'time-varying' processes caused by the linearization of a nonlinear system, the RBF modeling technique has shown better performance than piecewise linearization techniques. Meanwhile, the RBF model also performed competitively in a linear environment. The experimental results have demonstrated that the Equal-Distance Sample Rule (EDSR) for constructing the RBFN is simple and effective. It provides the same performance as the orthogonal least squares method, without large size matrix operations. The RBFN also has the potential for structure adaptation, since the basis functions are locally defined. Examples show how a function can be learned gradually by an RBFN.

**The main contributions of the dissertation can be summarized as follows:**

- Developed the MPSE algorithm for nonstationary signal processing.
- The resolution study provides mathematical representations of resolutions for different decomposition methods and conditions for yielding high resolution.
- The MPSE-based time-varying minimum variance magnitude estimator extends the conventional technique from stationary conditions to a nonstationary environment.
- Developed different MPSE-based forms of optimal Wiener filtering algorithms for recovering time-varying signals from broadband background noise.
- Developed MPSE-based high resolution time-varying spectral estimation for the analysis of EEG signals.
- Developed a systematic MPSE-based method for the construction of super wavelets for better time-frequency representation of nonstationary processes.

- Developed simple EDSR for the efficient construction of RBFNs for modeling of nonlinear dynamics.
- Developed a proof-of-concept experiment for structure adaptation of the RBFN for nonlinear function approximation.

**The author also lists the following future research directions :**

- Develop more applications for tracking and processing of nonstationary signals, such as radar or sonar signals, speech signals, and biomedical signals, especially for those applications to which the wavelet transforms have been applied.
- Develop a systematic order estimation method for the MPSE in a nonstationary environment.
- Study the possibility of using MPSE for ARMA model estimation, since after band-splitting a broadband ARMA process is decomposed into several relatively narrow-band signals which can be modeled as AR processes. Thus the MPSE can provide good estimation of the poles for an ARMA process. The remaining problem is to develop a method to estimate the ARMA process zeros based on the estimated poles.
- Develop fast algorithms for MPSE-based optimal filtering for real-time applications. Simplifying or avoiding spectral factorization is the key to this goal.
- Develop the “adaptive phase space sample rule,” such that the RBF samples the phase space according to the shape and the structure of the dynamic hypersurface, instead of using equal distance.
- Develop the structure adaptable RBFN network for linear or nonlinear adaptive filtering, system identification, and signal modeling. This will surely be one of the most important future research directions.
- Studying the criterion for refreshing RBFN knowledge in a time-varying nonlinear environment is also another challenging task.

## Bibliography

- [1] L. Franks, *Signal Theory*, Englewood Cliffs, NJ: Prentice-Hall, 1969.
- [2] C. W. Groetsch, *Elements of Applicable Functional Analysis*, New York, NY: Marcel Dekker, 1980.
- [3] N. Boccara, *Functional Analysis: An Introduction for Physicists*, San Diego, CA: Academic Press, 1990.
- [4] B. Aupetit, *A Primer on Spectral Theory*, New York, NY: Springer-Verlag, 1991.
- [5] G. E. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, Revised Edition, Oakland, CA: Holden-Day, 1976.
- [6] M. J. D. Powell, "Radial Basis Functions for Multivariable Interpolation: A Review," *IMA Conference on Algorithms for the Approximation of Functions and Data*, Shrivvenham, July 1985.
- [7] L. Cohen, "The Scale Representation," *IEEE Trans. on Signal Processing*, Vol. 41, No. 12, pp. 3275-3292, December 1993.
- [8] F. Hlawatsch and G. F. Boudreaux-Bartels, "Linear and Quadratic Time-Frequency Signal Representations," *IEEE Signal Processing Magazine*, Vol. 9, No. 2, pp. 21-67, April 1992.
- [9] O. Rioul and M. Vetterli, "Wavelets and Signal Processing," *IEEE Signal Processing Magazine*, Vol. 8, No. 4, pp. 15-37, October 1991.
- [10] I. Daubechies, "The Wavelet Transform: A Method for Time-Frequency Localization," in *Advanced Spectrum Analysis and Array Processing*, Vol. 1, Chapter 8, S. Haykin, Ed., Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [11] T. A. C. M. Claasen and W. F. G. Mecklenbrauker, "The Wigner Distribution - A Tool for Time-Frequency Signal Analysis Part I: Continuous-Time Signal," *Philips J. Res.*, Vol. 35, pp. 217-250, 1980.
- [12] T. A. C. M. Claasen and W. F. G. Mecklenbrauker, "The Wigner Distribution - A Tool for Time-Frequency Signal Analysis Part II: Discrete-Time Signal," *Philips J. Res.*, Vol. 35, pp. 276-300, 1980.

- [13] T. A. C. M. Claassen and W. F. G. Mecklenbrauker, "The Wigner Distribution - A Tool for Time-Frequency Signal Analysis Part III: Relations with Other Time-Frequency Signal Transformations," *Philips J. Res.*, Vol. 35, pp. 372-389, 1980.
- [14] B. Boashash, "Time-Frequency Signal Analysis," in *Advanced Spectrum Analysis and Array Processing*, Vol. 1, Chapter 9, S. Haykin, Ed., Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [15] C. A. Micchelli, "Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions," *Constructive Approximation*, Vol. 2, pp. 11-22, 1986.
- [16] M. J. D. Powell, "Radial Basis Function Approximations to Polynomials," *12th Biennial Numerical Analysis Conference*, Dundee, pp. 223-241, 1987.
- [17] M. D. Buhmann and M. J. D. Powell, "Radial Basis Function Interpolation on an Infinite Regular Grid," in *Algorithm for Approximation II*, J. C. Mason and M. G. Cox, Eds, London: Chapman and Hall, 1990.
- [18] B. Widrow and M. E. Hoff Jr., "Adaptive Switching Circuits," *IRE WESCON Convention Record*, Pt. 4, pp. 96-104, 1960.
- [19] M. E. Hoff Jr., *Learning Phenomena in Networks of Adaptive Switching Circuits*, Ph.D. Thesis, Stanford University, Stanford, CA, April 1962.
- [20] B. Widrow, J. R. Glover Jr., J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, J. Dong Jr., and R. C. Goodlin, "Adaptive Noise Canceling: Principles and Applications," *Proc. of the IEEE*, Vol. 63, No. 12, pp. 1692-1716, December 1975
- [21] P. J. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, Ph.D. Thesis, Harvard University, Cambridge, MA, August 1974.
- [22] D. B. Parker, "Learning-Logic: Casting the Cortex of the Human Brain in Silicon," *Technical Report TR-47*, Center for Computational Research in Economics and Management Science, MIT, Cambridge, MA, 1985.
- [23] Y. LeCun, "Une procedure d'apprentissage pour reseau a seuil assymetrique." *Cognitive*, Vol. 85, pp. 599-604, 1985.
- [24] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation," in *Parallel Distributed Processing*, Vol. 1, Chapter 8, D. E. Rumelhart and J. L. McClelland, Eds., Cambridge, MA: MIT Press, 1986.

- [25] D. B. Parker, "Optimal Algorithms for Adaptive Neural Networks: Second Order Back Propagation, Second Order Direct Propagation, and Second Order Hebbian Learning," *Proc. of the First IEEE International Conference on Neural Networks*, Vol. 2, pp. 593-600, San Diego, CA, June 1987.
- [26] D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*, New York, NY: Wiley, 1949.
- [27] S. Grossberg, *Studies of Mind and Brain*, Boston, MA: Reidel, 1982.
- [28] T. Kohonen, "Self-Organized Formation of Topologically Correct Feature Maps," *Biological Cybernetics*, Vol. 43, pp. 59-69, 1982.
- [29] J. J. Hopfield, "Neural Network and Physical Systems with Emergent Collective Computational Abilities," *Proc. of the National Academy of Sciences of the USA*, Vol. 79, pp. 2554-2558, 1982.
- [30] T. Kohonen, *Self-Organization and Associative Memory*, 2nd Edition, Berlin: Springer-Verlag, 1987.
- [31] M. A. Cohen and S. Grossberg, "Absolute Stability of Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-13, pp. 815-826, 1983.
- [32] D. S. Broomhead and D. Lowe, "Multivariable Functional Interpolation and Adaptive Networks," *Complex System*, Vol. 2, pp. 321-355, 1988.
- [33] S. W. Kuffler, "Discharge Patterns and Functional Organization of Mammalian Retina," *Journal of Neurophysiology*, Vol. 16, pp. 37-69, 1953.
- [34] A. Haar, "Zur Theorie der Orthogonalen Functionensysteme," *Math. Ann.*, Vol. 69, pp. 331-371, 1910.
- [35] D. Gabor, "Theory of Communication," *IEE Proc. (London)*, Vol. 93, No. 3, pp. 429-457, November 1946.
- [36] A. P. Calderon and A. Zygmund, "On the Existence of Certain Singular Integrals," *Acta. Math.*, Vol. 88, pp. 85-139, 1952.
- [37] A. Croisier, D. Esteban, and C. Galand, "Perfect Channel Splitting by Use of Interpolation, Decimation, and Tree Decomposition Techniques," *International Conference on Information Sciences/Systems*, Patras Greece, pp. 443-446, August 1976.

- [38] R. E. Crochiere, A. S. Weber and J. L. Flanagan, "Digital Coding of Speech in Subbands," *Bell Syst. Tech. J.*, Vol. 55, pp. 1069-1085, October 1976.
- [39] J. Morlet, G. Arens, I. Fourgeau, and D. Giard, "Wave Propagation and Sampling Theory," *Geophysics*, Vol. 47, pp. 203-236, February 1982.
- [40] A. Grossmann and J. Morlet, "Decomposition of Hardy Functions Into Square Integrable Wavelets of Constant Shape," *SIAM J. Math. Anal.*, Vol. 15, No. 4, pp. 723-736, July 1984.
- [41] P. Burt and E. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Trans. on Communications*, Vol. 31, No. 4, pp. 532-540, April 1983.
- [42] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. PAMI-11*, No. 7, pp. 674-693, July 1989.
- [43] P. Burt, "Multiresolution Techniques for Image Representation, Analysis and 'Smart' Transmission," *Proc. SPIE Conference on Visual Communication and Image Processing IV*, Vol. 1199, pp. 2-15, November 1989.
- [44] M. Antonini, M. Barlaud, D. Mathieu, and I. Daubechies, "Image Coding Using Wavelet Transform," *IEEE Trans. on Image Processing*, Vol. 1, No. 2, pp. 205-220, April 1992.
- [45] A. Grossmann, M. Holschneider, R. Kronland-Martinet, and J. Morlet, "Detection of Abrupt Changes in Sound Signal with the Help of Wavelet Transforms," in *Inverse Problems: An Interdisciplinary Study, Advance in Electronics and Electron Physics*, Supp. 19, P.C. Sabatier, Ed., pp. 289-306, New York, NY: Academic Press, 1987.
- [46] J. S. Lienard and D. d'Alessandro, "Wavelets and Granular Analysis of Speech," in *Wavelets: Time-Frequency Methods and Phase Space*, J. M. Combes, A. Grossmann, and Ph. Tchamitchian, Eds., Second Edition, New York, NY: Springer-Verlag, pp. 83-88, 1992.
- [47] S. Kadambe and G. F. Boudreaux-Bartels, "Applications of the Wavelet Transform for Pitch Detection of Speech Signals," *IEEE Trans. on Information Theory*, Vol. 38, No. 2, pp. 917-924, 1992.
- [48] J. Kovacevic and M. Vetterli, "Non-Separable Multidimensional Perfect Reconstruction Filter Banks and Wavelet Bases for  $\mathfrak{R}^n$ ," *IEEE Trans. on Information Theory*, Vol. 38, No. 2, pp. 533-555, 1992.



- [49] P. Flandrin, "Some Aspects of Nonstationary Signal Processing with Emphasis on Time-Frequency and Time-Scale Methods," in *Time-Frequency Methods and Phase Space*, J. M. Combes, A. Grossmann, and Ph. Tchamitchian, Eds., Second Edition, New York, NY: Springer-Verlag, pp. 83-88, 1992.
- [50] M. Basseville and A. Benveniste, "Multiscale Statistical Signal Processing," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 23-26, New York, NY, April 1989.
- [51] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1983.
- [52] M. J. Smith and T. P. Barnwell, "Exact Reconstruction Techniques for Tree Structured Subband Coders," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, pp. 434-441, June 1986.
- [53] M. Vetterli, "A Theory of Multirate Filter Banks," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-35, No. 3, pp. 356-372, March 1987.
- [54] P. P. Vaidyanathan, "Quadrature Mirror Filter Banks, M-Band Extensions and Perfect Reconstruction Techniques," *IEEE ASSP Magazine*, Vol. 4, No. 3, pp. 4-20, July 1987.
- [55] I. Daubechies, "Orthogonal Bases of Compactly Supported Wavelets," *Comm. in Pure and Applied Math.*, Vol. 41, No. 7, pp. 909-996, 1988.
- [56] I. Daubechies, "The Wavelet Transform, Time-Frequency Localization and Signal Analysis," *IEEE Trans. on Information Theory*, Vol. 36, No. 5, pp. 961-1005, September 1990.
- [57] R. K. Young, *Wavelet Theory and Its Applications*, Kluwer Academic Publishers, 1993.
- [58] de Prony, Baron (Gaspard Riche), "Essai Expérimental et Analytique: sur les lois de la dilatabilité des fluides élastiques et sur celles de la force expansive de la vapeur de l'eau et de la vapeur de l'alkool, à différentes températures," *J. Ec. Polytech.*, Vol. 1, cahier 2, pp. 24-76, 1795.
- [59] G. U. Yule, "On a Method of Investigating Periodicities in Disturbed Series, with Special Reference to Wolfer's Sunspot Numbers," *Philos. Trans. R. Soc. (London), Ser. A*, Vol. 226, pp. 267-298, July 1927.
- [60] S. G. Walker, "On Periodicity in Series of Related Terms," *Proc. R. Soc. (London), Ser. A*, Vol. 131, pp. 518-532, 1931.

- [61] N. Wiener, "Generalized Harmonic Analysis," *Acta Math.*, Vol. 55, pp. 117-258, 1930.
- [62] A. J. Khintchine, "Korrelationstheorie Der Stationaren Stochastischen Prozesse," *Math. Ann.*, Vol. 109, pp. 604-615, 1934.
- [63] N. Levinson, "The Wiener (Root Mean Square) Error Criterion in Filter Design and Prediction," *J. Math. Phys.*, Vol. 25, pp. 261-278, 1947.
- [64] J. P. Burg, "Maximum Entropy Spectral Analysis," *Proc. 37th Meeting of Society of Exploration Geophysicists*, Oklahoma City, Oklahoma, October 1967.
- [65] S. L. Marple Jr., "Resolution of Conventional Fourier, Autoregressive, and Special ARMA Methods of Spectral Analysis," *ICASSP-77*, pp. 74-77, May 1977.
- [66] S. Kay, *Modern Spectral Estimation: Theory and Application*, Englewood Cliffs, NJ: Prentice Hall, 1988.
- [67] S. Grossberg, "Adaptive Pattern Classification and Universal Recording I: Parallel Development and Coding of Neural Feature Detectors," *Biolog. Cybernetics*, Vol. 23, pp. 121-134, 1976.
- [68] G. A. Carpenter and S. Grossberg, "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine," *Computer Vision, Graphics, and Image Processing*, Vol. 37, pp. 54-115, 1983.
- [69] G. A. Carpenter and S. Grossberg, "Art 2: Self-Organization of Stable Category Recognition Codes for Analog Output Patterns," *Applied Optics*, Vol. 26, pp. 4919-4930, December 1987.
- [70] G. A. Carpenter and S. Grossberg, "Art 3: Hierarchical Search: Chemical Transmitters in Self-Organizing Pattern Recognition Architectures," *Proc. International Joint Conference on Neural Networks*, Vol. 2, pp. 30-33, Washington, DC, January 1990.
- [71] R. P. Lippmann, "Review of Neural Networks for Speech Recognition," *Neural Computation*, Vol. 1, 1-38, 1989.
- [72] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme Recognition Using Time-Delay Neural Networks," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-37, pp. 328-339, March 1989.
- [73] F. Palmieri, M. Datum, A. Shah, and A. Moriseff, "Sound Localization with a Neural Network Trained with the Multiple Extended Kalman Algorithm," *International Joint Conference on Neural Networks*, Vol. 1, pp. 125-131, Seattle, WA, 1991.

- [74] M. Elbaum and M. Syrkin, "Resolution-Limited Statistical Image Classification," *Proc. SPIE Conf. on Applications of Artificial Neural Networks IV*, Vol. 1965, pp. 120-141, Orlando, FL, April 1993.
- [75] C. A. Mead and M. A. Mahowald, "A Silicon Model of Early Visual Processing," *Neural Networks*, Vol. 1, pp. 91-97, 1988.
- [76] Y. T. Zhou, R. Chellappa, A. Vaid, and B. K. Jenkins, "Image Restoration Using Neural Network," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-36, No. 7, July 1988.
- [77] E. Oja, "Self-Organizing Maps and Computer Vision," in *Neural Networks for Perception*, H. Wechsler, Ed., Vol. 1, pp. 368-385, San Diego, CA: Academic Press, 1992.
- [78] V. C. Beiu, "Neural Network for Digital Image Enhancement," in *Artificial Neural Networks*, T. Kohonen, K. Makisara, O. Simula, and J. Kangas, Eds., Vol. 2, pp. 1071-1074, Amsterdam: North-Holland, 1991.
- [79] S. Skoneczny and A. Dzielinski, "Image Processing by Neural Network Implementation of the 2-D Kalman Filter," in *Artificial Neural Networks*, T. Kohonen, K. Makisara, O. Simula, and J. Kangas, Eds., Vol. 2, pp. 1075-1078, Amsterdam: North-Holland, 1991.
- [80] A. R. Barto, R. S. Sutton, and C. J. C. H. Watkins, "Learning and Sequential Decision Making," in *Learning and Computational Neuroscience: Foundations of Adaptive Neural Networks*, M. Gabriel and J. Moore, Eds., pp. 539-602, Cambridge, MA: MIT Press, 1990.
- [81] E. O. Postma, H. J. van den Herik, and P. T. W. Hudson, "Autonomous Control of Selective Attention: SCAN Architectures," in *Artificial Neural Networks*, T. Kohonen, K. Makisara, O. Simula, and J. Kangas, Eds., Vol. 2, pp. 1735-1738, Amsterdam: North-Holland, 1991.
- [82] S. Haykin and C. Deng, "Classification of Radar Clutter Using Neural Networks," *IEEE Trans. on Neural Networks*, Vol. 2, pp. 589-600, 1991.
- [83] S. Haykin and H. Leung, "Model Reconstruction of Chaotic Dynamics: First Preliminary Radar Results," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 4, pp. 125-128, San Francisco, CA, 1992.

- [84] G. Whittington, C. T. Spraklen, J. M. Haugh, and H. Faulkner, "Automated Radar Behavior Analysis Using Neural Network Architectures," *Proc. SPIE Conf. on Applications of Artificial Neural Networks IV*, Vol. 1965, pp. 44-59, Orlando, FL, April 1993.
- [85] A. P. Russo, "Neural Networks for Sonar Signal Processing," *Tutorial No. 8, IEEE Conference on Neural Networks for Ocean Engineering*, Washington, DC, 1991.
- [86] S. Chen, S.A. Billings, C. F. N. Cowan, and P.M. Grant, "Practical Identification of NARMAX Models Using Radial Basis Functions," *Int. J. Control*, Vol. 52, No. 6, pp. 1327-1350, 1990.
- [87] J. Conner, "Bootstrap Methods in Neural Networks and Time Series Prediction," in *Applications of Neural Networks to Telecommunications*, J. Alspector, R. Goodman, and T. X. Brown, Eds., pp. 125-131, Hillsdale, NJ: Lawrence Erlbaum, 1993.
- [88] R. M. Sanner and J. J. E. Slotine, "Gaussian Networks for Direct Adaptive Control," *IEEE Trans. on Neural Networks*, Vol. 3, No. 6, November 1992.
- [89] A. Zaknich, C. J. S. deSilva, and Y. Attikiouzel, "A Modified Probabilistic Neural Network (PNN) for Nonlinear Time Series Analysis," *1991 IEEE International Joint Conference on Neural Networks*, Vol. 2, pp. 1530-1535, 1991.
- [90] I. Grabec, "Modeling of Chaos by a Self-Organizing Neural Network," in *Artificial Neural Networks*, T. Kohonen, K. Makisara, O. Simula, and J. Kangas, Eds., Vol. 1, pp. 151-156, Amsterdam: North-Holland, 1991.
- [91] K. S. Narendra and K. Parthasarathy, "Identification and Control of Dynamic System Using Neural Networks," *IEEE Trans. on Neural Networks*, Vol. 1, pp. 4-27, March 1990.
- [92] O. A. Bashkirov, E. M. Braverman, and I. B. Muchnik, "Potential Function Algorithms for Pattern Recognition Learning Machines," *Automation and Remote Control*, Vol. 25, pp. 629-631, 1964.
- [93] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer, "Theoretical Foundations of the Potential Function Methods in Pattern Recognition Learning," *Automation and Remote Control*, Vol. 25, pp. 202-217, 1964.
- [94] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, New York, NY: Wiley, 1973.

- [95] J. Meinguet, "Multivariate Interpolation at Arbitrary Points Made Simple," *J. Appl. Math. Phys. (ZAMP)*, Vol. 30, pp. 292-304, 1979.
- [96] R. Franke, "Scattered Data Interpolation: Tests of Some Methods," *Math. Comp.*, Vol. 38, pp. 181-200, 1982.
- [97] S. Rippa, *Interpolation and Smoothing of Scattered Data by Using Radial Basis Functions*, M.S. Thesis, Tel Aviv University, Israel, 1984.
- [98] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal Least Square Learning Algorithm for Radial Basis Function Networks," *IEEE Trans. on Neural Networks*, Vol. 2, No. 2, March 1991.
- [99] M. Xie and A. A. (Louis) Beex, "Multiresolution Parametric Spectral Estimation," *26th Southeastern Symposium on System Theory*, Athens OH, pp. 432-436, March 10-22, 1994.
- [100] M. Xie and A. A. (Louis) Beex, "Multiresolution Parametric Spectral Estimation," *IEEE Trans. on Signal Processing*, submitted July 1994.
- [101] B. Y. Choi and Z. Bien, "Sliding-Windowed Weighted Recursive Least-Squares Method for Parameter Estimation," *Electronics Letters*, Vol. 25, No. 20, pp. 1381-1382, 1989.
- [102] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [103] S. McLaughlin, B. Mulgrew, and C. F. N. Cowan, "Performance Bounds for Exponentially Windowed RLS Algorithm in a Nonstationary Environment," *Mathematics in Signal Processing II*, J. G. McWhirter Ed. Oxford: Clarendon Press, 1990.
- [104] S. Haykin, *Adaptive Filter Theory*, 2nd Edition, Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [105] L. Ljung, *System Identification-Theory for the User*, Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [106] S. Y. Kung, "Systolic Arrays (for VLSI)," *Sparse Matrix Proc. 1978, Soc. Ind. Appl. Math.*, pp. 256-282, 1978.
- [107] M. Woodbury, "Inverting Modified Matrices," *Mem. Rep. 42*, Statistical Research Group, Princeton University, Princeton, NJ, 1950.
- [108] J. Capon, "High-Resolution Frequency-Wavenumber Spectrum Analysis," *Proc. IEEE*, Vol. 57, pp. 1408-1418, August 1969.

- [109] R. T. Lacoss, "Data Adaptive Spectral Analysis Methods," *Geophysics*, Vol. 36, pp. 661-675, August 1971.
- [110] W. Y. Chen and G. R. Stegen, "Experiments with Maximum Entropy Power Spectra of Sinusoids," *J. Geophys. Res.*, Vol. 79, pp. 3019-3022, July 1974.
- [111] D. N. Swingler, "Frequency Errors in MEM Processing," *IEEE Trans. ASSP-28*, pp. 257-295, April 1980.
- [112] T. J. Ulrych and R. W. Clayton, "Time Series Modeling and Maximum Entropy," *Phys. Earth Planet. Inter.*, Vol. 12, pp. 188-200, August. 1976.
- [113] R. Kronland-Martinet, J. Morlet, and A. Grossman, "Analysis of Sound Patterns Through Wavelet Transforms," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 1, No. 2, pp. 273-302, 1987.
- [114] J. S. Lim and A. V. Oppenheim, "Enhancement and Bandwidth Compression of Noisy Speech," *Proc. IEEE*, Vol. 67, pp. 1586-1604, December 1979.
- [115] A. A. (Louis) and M. Xie, "Time-varying Filtering Via Multiresolution Parametric Spectral Estimation," *1995 Int. Conf. on ASSP*, Detroit MI, 9-12 May 1995.
- [116] A. A. (Louis) and M. Xie, "Time-varying Filtering Via Multiresolution Parametric Spectral Estimation," *IEEE Trans. on Signal Processing*, submitted February 1995.
- [117] Charles W. Therrien, *Discrete Random Signal and Statistical Signal Processing*, Englewood Cliffs, NJ: Prentice Hall, 1988.
- [118] J. S. Lim, "All Pole Modeling of Degraded Speech," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-26, pp. 197-209, June 1978.
- [119] A. A. (Louis), "Optimal Filtering," *Lecture Notes on Parametric Modeling*, Virginia Polytechnic Institute and State University, 1992.
- [120] E. Basar, (Ed.), *EEG-brain dynamics. Relation between EEG and Brain Evoked Potentials*, Amsterdam: Elsevier/North Holland, 1980.
- [121] M. Steriade, D. A. McCormick, and T. J. Sejnowski, "Thalamocortical oscillations in the sleeping and aroused brain." *Science*, Vol. 262, pp. 679-685, 1993.
- [122] W. Singer, "Synchronization of Cortical Activity and its Putative Role in Information Processing and Learning." *Annual Review of Physiology*, Vol. 55, pp. 349-374, 1993.

- [123] Electro-cap International, Inc. *Instruction Manual for the ECI Electro-Cap Electrode System*, Eaton, OH, 1983.
- [124] K. H. Pribram, *Brain and Perception: Holonomy and Structure in Figural Processing*. Hillsdale, NJ: Erlbaum, 1991.
- [125] K. H. Pribram and D. McGuinness, "Arousal, Activation, and Effort in the Control of Attention," *Psychological Review*, Vol. 82, pp. 116-149, 1975.
- [126] M. I. Posner and S. E. Petersen, "The Attention Span of the Brain." *Annual Review of Neuroscience*, 13, pp. 23-42, 1990.
- [127] A. A. (Louis) and H. J. Crawford, *Time-Frequency Analysis of Electro-Encephalographic Signals*, A proposal for 1994 NSF special program, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1994.
- [128] H. Tiitinen, J. Sinkkonen, K. Reinikainen, K. Alho, J. Lavikainen, and R. Naatanen, "Selective Attention Enhances the Auditory 40-Hz Transient Response in Humans," *Nature*, Vol. 364, pp. 59-60, 1993.
- [129] R. K. Martinet, J. Morlet, and A. Grossmann, "Analysis of Sound Patterns through Wavelet Transforms," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 1, No. 2, pp. 273-302, 1987.
- [130] M. Xie and A. A. (Louis), "Optimal Wavelet Design Via Multiresolution Parametric Spectral Estimation," *1995 Int. Conf. on Wavelet Applications for Dual Use*, Orlando, April 1995.
- [131] H. H. Szu, B. Telfer, and S. Kadambe, "Neural Network Adaptive Wavelets for Signal Representation and Classification," *Optical Engineering*, Vol. 31, No. 9, pp. 1907-1916, September 1992.
- [132] H. H. Szu and B. Telfer, "Mathematics of Adaptive Wavelet Transforms: Relating Continuous with Discrete Transforms," *Optical Engineering*, Vol. 33, No. 7, pp. 2111-2124, July 1994.
- [133] D. F. Specht, "Generation of Polynomial Discriminant Functions for Pattern Recognition," *IEEE Trans. Electron. Comput.*, Vol. EC-16, pp. 308-319, June 1967.
- [134] A. G. Ivakhnenko, "The Group Method of Data Handling - A Rival of the Method of Stochastic Approximation," *Soviet Automatic Control*, Vol. 13, No. 3, pp.43-55, 1968.

- [135] A. G. Ivakhnenko, "Heuristic self-organization in problems of engineering cybernetics," *Avtomatika*, Vol. 6, pp. 207-219, 1970.
- [136] A. G., Ivakhnenko, "Polynomial Theory of Complex Systems," *IEEE Trans. on System, Man, and Cybernetics*, Vol. SMC-1, No. 4, pp. 364-378, October 1971.
- [137] B. Widrow and M. A. Lehr, "30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation," *Proc. IEEE*, Vol. 78, No. 9, September 1990.
- [138] T. S. Parker and L. O. Chua, *Practical Numerical Algorithms for Chaotic System*, New York, NY: Springer-Verlag, 1989.
- [139] C. Frank Hoppensteadt, "Analysis and Simulation of Chaotic System" in *Applied Mathematical Sciences*, F. John, J. E. Marsden, and L. Sirovich Eds., Volume 94, Chapter 2, New York, NY: Springer-Verlag, 1993.
- [140] D. Berger, K. Pribram, H. Wild, and C. Bridges, "An Analysis of Neural Spike-Train Distributions: Determinants of the Response of Visual Cortex Neurons to Changes in Orientation and Spatial Frequency," *Experimental Brain Research*, Vol. 80, No. 1, pp. 129-134, 1990.
- [141] D. Berger and K. Pribram, "The Relationship between the Gabor Elementary Function and Stochastic Model of the Inter-spike Interval Distribution in the Responses of Visual Cortex Neuron," *Biological Cybernetics*, Vol. 67, 1991.
- [142] D. Berger and K. Pribram, "From Stochastic Resonance to Gabor Functions: an Analysis of the Probability Function of Interspike Intervals Recorded from Visual Cortical Neurons," *Rethinking Neural Networks: Quantum Fields and Biological Data*, K. Pribram ed., Hillsdale, NJ: Inns Press, 1993.
- [143] A. R. Bulsara and A. J. Maren, "Coupled Neural-Dendritic Processes: Cooperative Stochastic Effects and Analysis of Spike Trains," *Rethinking Neural Networks: Quantum Fields and Biological Data*, K. Pribram ed., Hillsdale, NJ: Inns Press, 1993.
- [144] M. Xie, K. Pribram, and J. King, "Are Neural Spike Trains Deterministically Chaotic or Stochastic Processes? the Dynamics of Spontaneous Neural Interspike Intervals," *Origins: Brain and self organization*, K. Pribram Ed., Hillsdale NJ: Lawrence Erlbaum Associates, 1994.



## Vita

Min Xie earned the B.S. in Electrical Engineering from Wuhan University of Geodesy and Cartography, Wuhan, China, in 1982, and the M.S. in Electrical Engineering from Shanghai University of Technology, Shanghai, China, in 1985. He then was a lecturer in the Electrical Engineering Department at Shanghai University of Technology until 1988. Since then he has been pursuing higher education in the U.S., first at the University of Illinois at Urbana-Champaign, and more recently at Virginia Polytechnic Institute and State University, while also working as a research engineer with the Center for Brain Research and Information Sciences at Radford University. He anticipates receiving his Ph.D. in Electrical Engineering from Virginia Tech in May of 1995.

His research interests are in the areas of signal processing, spectral analysis, adaptive filtering, system identification, linear and nonlinear system control, automation and biomedical instrumentation. His hobbies include computer graphics and animation, the making of electronic toys, gardening, swimming, wood working, and traveling.

He is a member of IEEE (Societies: Signal Processing; Control System; Engineering in Medicine and Biology).

*Min Xie*