University of Alberta

Library Release Form

Name of Author: Soner Bekleric
Title of Thesis: Nonlinear Prediction via Volterra Series and Applications to Geophysical Data
Degree: Master of Science
Year this Degree Granted: 2008

Soner Bekleric

Date:

University of Alberta

Nonlinear Prediction via Volterra Series and Applications to Geophysical Data

by

Soner Bekleric ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**

in

Geophysics

Department of Physics

Edmonton, Alberta
Spring 2008

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis, entitled **Nonlinear Prediction via Volterra Series and Applications to Geophysical Data** submitted by Soner Bekleric in partial fulfillment of the requirements for the degree of **Master of Science**.

Prof. Mauricio D. Sacchi (Supervisor)

Dr. Vadim A. Kravchinsky

Dr. Jeff Gu

Dr. Osmar R. Zaïane

Date:

to

My Family

Abstract

Linear filter theory has proven useful in many seismic data analysis applications. However, the general development of linear filter theory is limited by the implicit approximations typically found in seismic processing; one reason for this is to avoid effects of nonlinearity. This thesis concentrates on the implementation of nonlinear time series modeling based on an autoregressive method. The developed algorithm utilizes third-order Volterra kernels to improve predictability of events that cannot be predicted using linear prediction theory.

Volterra series are analyzed. The application and implementation of a nonlinear autoregressive algorithm to the problem of modeling complex waveforms in the $f-x$ domain is studied. Problems of random noise attenuation and adaptive subtraction of multiples are reexamined by the new Volterra autoregressive algorithm. Synthetic and field data examples are used to illustrate the theory and methods presented in this thesis.

Acknowledgements

I would like to express my thanks to my supervisor Prof. Mauricio D. Sacchi for his guidance, endless support, patient, and help.

I would like to acknowledge all members of my committee for their useful suggestions.

My sincere thanks to my former and present friends/colleagues in the Signal Analysis and Imaging Group for sharing their knowledge and their scientific suggestions, particularly Cristina Moldoveanu-Constantinescu, Mostafa Naghizadeh, Naser Yousefzadeh, Sam Kaplan, and Somanath Misra.

I wish to thank my good friends Barkim and Derya Demirdal for the beautiful time we shared together in Edmonton.

I would like to express my deepest gratitude to my family for their constant support and boundless love.

Contents

1 Introduction **1**

1.1 Applications 3

1.2 Motivation and Goals of the Thesis 5

1.3 Thesis Outline 5

2 Linear Systems and Prediction **7**

2.1 Introduction 7

2.2 Linear Process 8

2.3 Linear Prediction 10

2.3.1 Forward and backward prediction 12

2.3.2 Estimating AR coefficients via Yule-Walker Equations 12

2.3.3 Estimating AR coefficients via Burg's algorithm 14

2.3.4 Computing the AR coefficients without limiting the aperture 16

2.4 1-D Synthetic and Real Data Examples 19

2.5 Power Spectrum 23

2.6 Summary 24

3 Nonlinear Prediction **26**

3.1 Nonlinear Processes via the Volterra Series 26

3.1.1 Time domain representation 27

3.1.2 Frequency domain representation of Volterra kernels 28

3.1.3 Symmetry property of Volterra kernels 30

3.2 Nonlinear Modeling of Time Series 32

3.3 1-D Synthetic and Real Data Examples 40

3.4 Summary 46

4 Nonlinear Modeling of Complex Waveforms in the $f-x$ Domain 47

4.1 Linear Prediction in the $f-x$ Domain 47

4.2 Analysis of Optimum Filter Length 50

4.3 Nonlinear Prediction of Complex Waveforms 53

4.4 Noise Removal and Volterra Series 62

4.5 Synthetic and Real Data Examples 63

4.6 Summary 74

5 Adaptive Subtraction of Multiples 75

5.1 Introduction 75

5.2 Prediction Error Operator 77

5.3 Synthetic Data Examples 78

5.4 Real Data Examples 83

5.5 Summary 90

6 Conclusions and Future Directions 91

6.1 Future Directions 93

List of Tables

3.1 Filter length and the number of prediction coefficients. 35

List of Figures

2.1 AR IIR filter representation. 10

2.2 The trade-off parameter μ estimates the optimum solution. 19

2.3 1-D synthetic data for comparison of prediction between linear prediction theories. 21

2.4 Arctic Oscillation data for standardized nonlinear sea-level pressures for comparison of prediction between linear prediction theories. . . . 22

2.5 PSD Estimation of Arctic Oscillation data for standardized nonlinear sea-level pressures. 24

3.1 Schematic representation of a system characterized by a third-order Volterra series. 28

3.2 Frequency domain Volterra model of a cubic nonlinear system. . . . 29

3.3 Volterra AR diagram. 34

3.4 1-D synthetic data for comparison of prediction between linear prediction theory and third order Volterra series. 42

3.5 1-D synthetic data. 43

3.6 Arctic Oscillation data for standardized nonlinear sea-level pressures for comparison of prediction between linear prediction theory and third order Volterra series. 44

3.7 Arctic Oscillation data for standardized nonlinear sea-level pressures. 45

4.1 Optimum filter length for the data in Figure 4.2. 52

4.2 Synthetic data example for different filter lengths. 53

4.3 (a) Prediction of Figure 4.2(b) ($p = 3$). (b) The error between original data and predicted data. 54

4.4 (a) Prediction of Figure 4.2(b) ($p = 6$). (b) The error between original data and predicted data. 55
4.5 (a) Prediction of Figure 4.2(b) ($p = 15$). (b) The error between original data and predicted data. 56
4.6 Synthetic data example for different filter lengths. 57
4.7 Optimality of filter length for the data in Figure 4.6. 58
4.8 (a) Prediction of Figure 4.6(b) ($p = 3$). (b) Error between original data and predicted data. 59
4.9 (a) Prediction of Figure 4.6(b) ($p = 5$). (b) Error between original data and predicted data. 60
4.10 (a) Prediction of Figure 4.6(b) ($p = 15$). (b) Error between original data and predicted data. 61
4.11 Same as the data in Figure 4.6. 63
4.12 2-D synthetic data for comparison of prediction between linear prediction theory and third order Volterra series. 65
4.13 2-D synthetic data for comparison of prediction between linear prediction theory and the cubic part of a Volterra series. 66
4.14 2-D synthetic data. 67
4.15 2-D synthetic data for comparison of prediction between linear prediction theory and a third-order Volterra series. 68
4.16 2-D synthetic data. 69
4.17 2-D synthetic data. 71
4.18 2-D real data for comparison of prediction between linear prediction theory and a third-order Volterra series. 72
4.19 2-D real data. 73

5.1 Multiple types. 77
5.2 Synthetic data example. 79
5.3 Synthetic data example with two multiples and one primary. 80
5.4 Synthetic data example with two multiples and one primary. 81
5.5 Synthetic data example with two multiples and three primaries. . . 82
5.6 Real Data example. Common offset gather. 84
5.7 Real Data example. Common offset gather (closer look). 85

5.8 Real Data example A common offset gather (a closer look of removed multiples). 86
5.9 Real Data example shot gather. 87
5.10 Real Data example shot gather (closer look). 88
5.11 Real Data example shot gather (a closer look of removed multiples). 89

List of symbols

Symbol	Name or description
$(\cdot)^*$	Complex conjugate
$(\cdot)_n^b$	Backward prediction at time index n
$(\cdot)_n^f$	Forward prediction at time index n
$(\cdot)^T$	Matrix transpose
$(\hat{\cdot})$	Hat denotes prediction
$\mathbf{A}$	Filter matrix (either linear or nonlinear)
a_i	Linear prediction filter coefficient at time index i
b_{ij}	Quadratic prediction filter coefficient at time index jk
c_2	Second order cumulant (autocorrelation)
c_{lms}	Cubic prediction filter coefficient at time index lms
C_n	$n^{th}-$ order spectrum
$\mathbf{d}$	Data vector
$E[\cdot]$	Expectation operator
ε_n	Error or innovation process of time index n
$\mathbf{e}$	Noise vector
f	Temporal frequency in Herts (Hz)
$H_1(\cdot)$	Linear transfer function of Volterra series
$H_2(\cdot, \cdot)$	Quadratic transfer function of Volterra series
$H_3(\cdot, \cdot, \cdot)$	Cubic transfer function of Volterra series
h	Offset (distance between source and receiver)
$h_k(\sigma_1, \ldots, \sigma_k)$	Volterra kernels of the system
$\mathbf{I}$	Identity matrix
Im	Imaginary part of a complex variable
J	Cost function
k_p	Prediction coefficient
$\mathbf{m}$	Model vector
μ	Trade-off parameter
N	Number of data samples

Symbol	**Name or description**
$n-$D	$n-$ dimensional: e.g 1-D
ω	Angular/radial frequency (radian/seconds)
$P_{AR}(\omega)$	AR Power spectrum estimator
$P_{xx}(\omega)$ or $C_2(\omega)$	Power spectrum estimation
p	Linear filter length or AR model order
q	Quadratic filter length or AR model order
$\mathbf{R}$	Autocorrelation matrix
R_p	Autocorrelation sequence at lag p
r	Cubic filter length or AR model order
Re	Real part of a complex variable
S	Single waveform in $f-x$ domain
s	Single waveform in $t-x$ domain
σ_ε^2	Innovation of the system
σ_p	White noise variance
t	Travel time in seconds
θ	Apparent slowness
$Z(x_n) = X(z)$	Z-transform of x_n

List of abbreviations

Abbreviation	**Name or description**
AO	Arctic oscillation
AR(p)	Autoregressive model of order p
AR(p, q, r)	Autoregressive Volterra model of order p, q, r
ARMA	Autoregressive moving average
DFT	Discrete Fourier transform
FSME	Free surface multiple elimination
FT	Fourier transform
IIR	Infinite impulse response
MA(q)	Moving average model of order q
PEF	Prediction error filter
PSD	Power spectrum density
RMSE	Root-mean square error
SNR	Signal-to-noise ratio
SRME	Surface related multiple elimination
YW	Yule-Walker equations

Chapter 1

Introduction

Linear prediction theory has an important role in many signal processing and imaging applications. Linear prediction algorithms recursively model future/past samples of a signal using a linear combination of its past/future samples. Linear prediction is not restricted to time and can be used to model the spatial variability of signals. The latter is particularly important in exploration seismology where we record seismic wave field that are a function of time and space and one desires to predict in both space and/or time the evolution of a signal. The mismatch between the original signal and the predicted one is often used as an estimator of additive noise.

The problem of linear prediction of time series can be tracked back to the work of Yule (1927) who described a method to model sun spots numbers using a particular type of linear prediction model called the auto-regressive model. The idea is to find a parametric representation that allows to model the temporal variability of a physical process and, to use the parametric model to estimate the power spectral density of the time series (Marple, 1987).

In applied seismology predictive deconvolution (another form of linear prediction) was introduced by Robinson (1954) to solve the source deconvolution problem when the seismic source function is unknown. Predictive deconvolution is at the heart of seismic exploration seismology and used on a daily basis to enhance the resolution of seismic records and produce estimates of the earth's reflectivity sequence from seismic probes (Robinson and Treitel, 1980).

Canales (1984) proposed to use linear prediction methods to predict 2D wave fields in the $f - x$ domain (frequency-space). The technique proposed by Canales is the cornerstone of methods to increase the signal to noise ratio (SNR) of seismic records. In this case, the signal at a given channel for a monochromatic frequency f can be predicted via a linear combination of waveforms measured at adjacent channels. If a signal can be predicted, the mismatch that exists between the predicted signal and the observed signal is considered an estimate of the noise in the data. The latter is a summarized description of $f - x$ deconvolution for the attenuation of spatially random noise. Linear prediction is also the basis of algorithms for seismic record interpolation (Spitz, 1991; Naghizadeh and Sacchi, 2007).

Nonlinear prediction, on the other hand, is the prediction of a signal from nonlinear combinations of its past or future values (Wiener, 1958). Nonlinear prediction arose in the study of nonlinear systems for the treatment of weak nonlinearities arising in system theory (Cherry, 1994). In addition, nonlinear prediction has been used to predict the behavior of geophysical signals in the context of hydrological studies (Bracalari and Salusti, 1994), ocean waves interactions with platforms (Koh and Powers, 1985; Powers et al., 1990).

1.1 Applications

I will give some examples of previous applications of nonlinear prediction problems that use the truncated Volterra series.

Adaptive Noise Cancellation

The electrocardiograph (ECG) is a signal generated by the hearth and this signal should be eliminated to obtain other generated biological signals (Coker and Simkins, 1980). Adaptive filter cancellation of ECG from biological signals can be achieved via conventional linear estimation processes. In this application Coker and Simkins (1980) proposed a nonlinear adaptive ECG cancellation.

Equalization

Benedetto and Biglieri (1983) focused on a nonlinear equalization of digital satellite channels via a Volterra filter using only first- (linear) and third-order (cubic) components of Volterra series. Nonlinear coefficients are needed to remove distortions in a digital signal channel (Benedetto and Biglieri, 1983).

In another study multi-input multi-output (MIMO) Volterra filter equalization of wireless communication systems using $p^{th}-$order inverse technique is proposed by Fang et al. (2000).

Image analysis

Collis et al. (1997) studied image de-interlacing problems using Volterra filters and applied their algorithm to reconstruct television images. Their nonlinear filter provided more smoother edges and curves than a linear filter. Similar technique was applied to video de-interlacing problem with a Volterra model by Giani et al. (2000).

Ocean waves and platforms

Tension leg platform (TLP) is a floating platform chained to the ocean floor via tendons which are under a tension; that is why it is named as tension leg platform. Koh and Powers (1985) modeled the irregular wave oscillations in time domain with a second- order Volterra series. Particularly, the second-order (quadratic) component of a Volterra series models the relationship between a wave excitation and surge response of big waves producing by TLP. Their filter also allows dividing into the observed surge response into its linear and nonlinear parts. Powers et al. (1990) improved this study to a low- frequency drift oscillation of a TLP due to irregular sea waves by correlating nonlinear forces with a TLP data at low frequencies. They demonstrated that the nonlinear transfer function may successfully model quadratic nonlinear mechanisms and measured a TLP response. Furthermore, Kim et al. (1994) studied a deconvolution technique based on an impulse invariance standard $Z-$ transform to derive linear and nonlinear coefficients of a surge response.

Biomedical

Zhang et al. (1998) studied how memory length, noise, order of nonlinearity, type of data can affect Volterra kernel estimation; they explored this in the context of nonlinear lung tissue mechanics around ventilatory breathing frequencies. Other applications of Volterra modeling are found in Kellman et al. (2003) and Zhong et al. (2006)

Applications of higher- order spectra (HOS) and Volterra series

The connection between HOS and Volterra series were investigated by researchers under different topics namely; nonlinear ocean waves and HOS (Powers et al., 1997); non-normal random processes (Gurley et al., 1996); nonlinearity detection (LeCaillec and Garello, 2004).

1.2 Motivation and Goals of the Thesis

The motivation of this thesis is to introduce nonlinear prediction to the seismic data processing community and study the feasibility of nonlinear prediction algorithms for problems of waveform modeling for SNR enhancement and adaptive modeling and subtraction for the general problem of coherent noise attenuation.

The goal of this thesis is to study nonlinear prediction algorithms and their applicability to model seismic waveforms in the $f-x$ domain. In particular, I would like to model signals that are not correctly represented by the linear prediction theory. Those signals often involve seismic waveforms with hyperbolic moveout (seismic reflections or diffractions) that are often immersed in noise and one would like to enhance prior to imaging.

1.3 Thesis Outline

- Chapter 2 presents an introduction to linear modeling of time series.
- Chapter 3 reviews the concept of nonlinear prediction and its realization via Volterra series.

- Chapter 4 provides a detailed study of linear and nonlinear prediction applied to SNR enhancement of seismic records.

- Chapter 5 studies the feasibility of using nonlinear prediction for adaptive subtraction of coherent noise from seismic records.

- Chapter 6 contains a summary. In addition, I provide conclusions including the limitations of nonlinear prediction theory in seismic signal processing. Finally, I provide a discussion of future research directions.

Chapter 2

Linear Systems and Prediction

2.1 Introduction

Much research has been done in the area of linear prediction techniques, resulting in applications ranging from communications to physics (Makhoul, 1975; Gulunay, 1986; Collis et al., 1997). The majority of research in time series analysis has been in the analysis of power spectrum estimation (Akaike, 1969; Box and Jenkins, 1970; Marple, 1987).

As I mentioned in the introductory Chapter, linear prediction in geophysics has been mainly used for deconvolution (Robinson, 1954), SNR enhancement (Canales, 1984) and interpolation (Spitz, 1991).

The first part of this chapter provides a background of linear prediction theory from both a seismic processing viewpoint and from a more general signal processing point of view. Different linear prediction techniques are described with associated examples. In particular, I discuss three methods to estimate linear prediction coefficients: the Yule-Walker method, Burg's algorithm and least squares approach.

Finally, I provide examples using geophysical series.

2.2 Linear Process

It should be stressed that any wide sense stationary random process can be represented via the superposition of a deterministic part s_n plus a non deterministic part x_n (random or stochastic part),

$$z_n = s_n + x_n. \tag{2.1}$$

Wold decomposition theorem (Wold, 1965; Ulrych and Sacchi, 2005), in addition states that the non deterministic part can be represented as a filtered sequence of white noise

$$x_n = \sum_{i=1}^{\infty} g_i \varepsilon_{n-i} \tag{2.2}$$

where $g_0 = 1$, $\sum_{i=1}^{\infty} |g_i|^2 < \infty$ (finite length impulse response) and ε_n represents the white noise which is uncorrelated with s_n. I can change the equation above by a finite length discrete general linear model

$$x_n = \sum_{i=1}^{N} g_i \varepsilon_{n-i} \tag{2.3}$$

where ε_n is the innovations process and x_n depends on its past input values (therefore it is casual).

In the z-domain the model becomes

$$X(z) = G(z)E(z) \tag{2.4}$$

where $G(z)$ is the transfer function, and $E(z)$ is the z-transform of the white noise process (innovations).

The transfer function $G(z)$ can be approximated by rational function in the $z-$transform domain:

$$G(z) = \frac{B(z)}{A(z)} \,. \tag{2.5}$$

I can rewrite the equation (2.4) as follows

$$X(z) = \frac{B(z)}{A(z)} E(z) \,. \tag{2.6}$$

The latter, when mapped back to time, is represented as follows:

$$x_n + \sum_{i=1}^{p} a_i x_{n-i} = \varepsilon_n + \sum_{i=1}^{q} b_i \varepsilon_{n-i} \,. \tag{2.7}$$

When all the coefficients $b_i = 0$ the last expression is often called an autoregressive model of order p, AR(p) (Box and Jenkins, 1970; Makhoul, 1975; Chatfield, 1989). Otherwise, when all $a_i = 0$ then the model called a Moving Average (MA) model. The developed algorithm is written in MATLAB.

Linear prediction models for time series modeling use linear autoregressive models. Figure 2.1 represents the infinite impulse response (IIR) of an AR model.

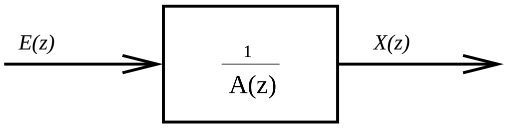

Figure 2.1: AR IIR filter representation. After Makhoul (1975) and Ulrych and Sacchi (2005).

2.3 Linear Prediction

The relationship between input/output for a linear time -invariant system is given by the classical convolution integral (Rugh, 1981; Oppenheim and Schafer, 1989; Schetzen, 2006),

$$y(n) = \int_{-\infty}^{\infty} h(\sigma)x(n-\sigma)d\sigma\,. \tag{2.8}$$

In the last expression $h(\sigma)$ is the impulse response of the system or kernel function that defines the input $x(n)$ /output $y(n)$ of the system.

Equation 2.8 is often used in the discrete form. The convolution integral is replaced by a convolution sum and signals are replaced by a finite length discrete time series:

$$y_n = \sum_{i=1}^{N} a_i x_{n-i}\,. \tag{2.9}$$

The latter is the discrete convolution sum that arises very frequently in geophysics and other sciences. It basically relates the output of a system that is excited with an input signal x_n via the convolution of the signal with the impulse response of the system, in this case called a_n.

In linear prediction theory, one attempts to predict a signal by its past values (or future values) by replacing the output signal y_n by a future (one step ahead) prediction of the input signal. Mathematically this is equivalent to the following model for the signal x_n:

$$\begin{aligned} x_n &= \sum_{i=1}^{p} a_i x_{n-i} + \varepsilon_n, \\ &= a_1 x_{n-1} + a_2 x_{n-2} + \dots a_p x_{n-p} + \varepsilon_n \end{aligned} \tag{2.10}$$

where p is the order of the prediction filter a_n, $n = 1, \dots, p$. I have also introduced a prediction error ε_n to account for the part of the signal that cannot be properly predicted.

These models are of importance for the analysis of time series in both time and frequency domain. This model is also called an Autoregressive model of order p (AR(p)). One can immediately see that the latter entails a regression model but rather that being a regression on arbitrary basis functions, the regression uses past values of the data one is trying to predict. The latter is the reason behind the name Auto-regressive model.

The coefficients of the system (AR coefficients) can be estimated using various methods. Marple (1987) provides a review of methods for solving the prediction problem. In particular, one can adopt a minimum error formulation (least squares approach) to find the coefficients that minimize the error function J given by:

$$J = \sum_n (x_n - \sum_{i=1}^{p} a_i\, x_{n-i})^2 \,. \tag{2.11}$$

The filter a_i, $i = 1 \dots p$ that minimizes the cost function J can be used to predict the signal and to compute the so called AR-spectral estimator (Marple, 1987).

In my thesis I have adopted the least squares approach to estimate the coefficients of linear and non-linear prediction systems but bear in mind that other method (at least for the linear prediction problem) are also plausible. In the following section I will discuss three methods to estimate the coefficients $a(i), i = 1 \ldots p$ from the data $x(n)$: the Yule-Walker method, Burg's algorithm, and the least squares approach.

2.3.1 Forward and backward prediction

Off-line processing allows computation of forward and backward prediction operators. In other words, I can use past samples of data to predict future samples and/or I can use future samples of data to predict past samples. This allows us to formulate a problem with two systems of equations. One for forward prediction,

$$x_n^f = \sum_{i=1}^{p} a_i^f x_{n-i} + \varepsilon_n, \tag{2.12}$$

and, the other for backward prediction,

$$x_n^b = \sum_{i=1}^{p} a_i^b x_{n+i} + \varepsilon_n \, . \tag{2.13}$$

2.3.2 Estimating AR coefficients via Yule-Walker Equations

I can write equation (2.6) as follows

$$X(z) = \frac{E(z)}{A(z)}, \tag{2.14}$$

and the autocorrelation in the z domain (Ulrych and Sacchi, 2005) is

$$R(z) = X(z)X(\frac{1}{z}), \tag{2.15}$$

rewriting to

$$R(z) = \frac{1}{A(z)}\frac{1}{A(\frac{1}{z})}E(z)E(\frac{1}{z}), \tag{2.16}$$

then the white noise $E(z)E(1/z)$ must be equal to variance (σ_ε) as follows

$$A(z)R(z) = A^{-1}(1/z)\sigma_\varepsilon^2\,. \tag{2.17}$$

Transforming to the time domain

$$(1+a_1 z+\cdots+a_p z^p)(\cdots+r_2 z^{-2}+r_1 z^{-1}+r_0+\cdots+r_2 z^2+\cdots) = \sigma_\varepsilon^2(\cdots+x_1 z^{-1}+1)\,. \tag{2.18}$$

and finally, after rearranging the latter as a system of equations I obtain:

$$\begin{array}{ccccccccccc} r_0 & + & a_1 r_{-1} & + & a_2 r_{-2} & + & \cdots & + & a_p r_{-p} & = & \sigma_\varepsilon^2 \\ r_1 & + & a_1 r_0 & + & a_2 r_{-1} & + & \cdots & + & a_p r_{-p+1} & = & 0 \\ \vdots & & \vdots & & \vdots & & \cdots & & \vdots & = & \vdots \\ r_p & + & a_1 r_{p-1} & + & a_2 r_{p-2} & + & \cdots & + & a_p r_0 & = & 0 \end{array} \tag{2.19}$$

which can be written in matrix form as:

$$\underbrace{\begin{bmatrix} r_0 & r_{-1} & \cdots & r_{-p} \\ r_1 & r_0 & \cdots & r_{-p+1} \\ \vdots & \ddots & \ddots & \vdots \\ r_p & r_{p-1} & \cdots & r_0 \end{bmatrix}}_{\mathbf{R}} \underbrace{\begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix}}_{\mathbf{a}} = \underbrace{\begin{bmatrix} \sigma_\varepsilon^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\mathbf{e}} \tag{2.20}$$

where the matrix is the data autocorrelation matrix and σ_ε^2 is the innovation variance .

The system above is written as

$$\mathbf{Ra} = \sigma_\varepsilon^{\mathbf{2}}\mathbf{e_1}, \tag{2.21}$$

where $\mathbf{e}_1^T = [1, 0, \cdots, 0]$ is the zero-delay spike vector. The AR Yule-Walker equations are formed with an autocorrelation matrix (Marple, 1987). Since $R_{-i} = R_i^*$ the autocorrelation matrix, which is almost always invertible in equation (2.21) is both Toeplitz and Hermitian. The Levinson recursion solves the resulting system in $(p+1)^2$ operations.

2.3.3 Estimating AR coefficients via Burg's algorithm

Burg (1975) developed an AR algorithm to estimate AR prediction coefficients when the autocorrelation sequence of the system is unknown. The primary advantage of Burg's method is to estimate prediction coefficients directly form the data, in contrast to the least squares solution and the Yule-Walker method. In addition, the method provides a stable AR model in a time series with low noise levels, and is useful in predicting short data records (Robinson and Treitel, 1980). Burg's method first estimates the reflection coefficients by minimizing the forward and backward prediction errors. The accuracy of the method decreases for higher-order AR prediction models, and time series with longer data samples. Similarly to the Yule-Walker equations, the Levinson recursion determines prediction coefficients defined as the last autoregressive parameter estimates for each model order p (Marple, 1987). I follow Marple's (1987) way to obtain the equations of Burg's algorithm.

The forward and backward prediction errors are, respectively, given by:

$$\varepsilon_n^f = x_n^f + \sum_{i=1}^{p} a_i^f x_{n-i}\,, \tag{2.22}$$

and

$$\varepsilon_n^b = x_{n-p}^b + \sum_{i=1}^{p} a_i^b x_{n+i}. \tag{2.23}$$

Burg's method estimates prediction coefficients via a least squares solution. Minimizing the arithmetic mean of forward and backward prediction error power subject to recursion similar to equations (2.22) and (2.23) as follows:

$$\rho_p^{fb} = \frac{1}{2}[\frac{1}{N}\sum_{n=p+1}^{N} |\varepsilon_p^f|^2 + \frac{1}{N}\sum_{n=p+1}^{N} |\varepsilon_p^b|^2]\,. \tag{2.24}$$

where N denotes the length of recorded data. Note that only data that has been recorded is used in the summation. Therefore, ρ_p^{fb} is assumed as a single parameter, so that the prediction coefficient k_p is a prediction coefficient. Prediction errors from order $p-1$ are found and by setting the complex derivative of the equation (2.24) to zero

$$\frac{\partial \rho_p^{fb}}{\partial Rek_p} + i\frac{\partial \rho_p^{fb}}{\partial Imk_p} = 0 \tag{2.25}$$

where Re and Im are real and imaginary parts, respectively, of the complex derivative. A least squares solution ensures a solution for prediction coefficients, k_p as follows:

$$k_p = \frac{-2\sum_{n=p+1}^{N} \varepsilon_{p-1}^{f}(n)\varepsilon_{p-1}^{b*}(n-1)}{\sum_{n=p+1}^{N} |\varepsilon_{p-1}^{f}(n)|^2 + \sum_{n=p+1}^{N} |\varepsilon_{p-1}^{b}(n-1)|^2} \tag{2.26}$$

where $*$ is the complex conjugation.

2.3.4 Computing the AR coefficients without limiting the aperture

The Yule-Walker and Burg algorithms are often used in signal processing algorithms for their computational efficiency at the time of computing prediction error coefficients. In what follows I will provide a method that I have introduced in order to solve for the coefficients of the linear prediction problem using only the data that are available. In other words, I will avoid creation of the correlation matrix and directly posed the problem as a least-squares problem.

Consider a filter length $p = 3$ and a time series of length $N = 7$. Using equations (2.12) and (2.13) I can write the following equations:

$$\begin{aligned}
x_1 &= a_1 x_2 + a_2 x_3 + a_3 x_4 + \varepsilon_1 \\
x_2 &= a_1 x_3 + a_2 x_4 + a_3 x_5 + \varepsilon_2 \\
x_3 &= a_1 x_4 + a_2 x_5 + a_3 x_6 + \varepsilon_3 \\
x_4 &= a_1 x_5 + a_2 x_6 + a_3 x_7 + \varepsilon_4 \\
x_4 &= a_1 x_3 + a_2 x_2 + a_3 x_1 + \varepsilon_5 \\
x_5 &= a_1 x_4 + a_2 x_3 + a_3 x_2 + \varepsilon_5 \\
x_6 &= a_1 x_5 + a_2 x_4 + a_3 x_3 + \varepsilon_6 \\
x_7 &= a_1 x_6 + a_2 x_5 + a_3 x_4 + \varepsilon_7 .
\end{aligned} \tag{2.27}$$

Notice that only data that has been recorded is used to compute the prediction coefficients a_p. In other words, no assumption about samples of non-recorded data is made. In addition, I are conveniently using forward and backward prediction to avoid any type of truncation or aperture artifact. Data that cannot be predicted with equation (2.12) is predicted via equation (2.13) and and vice versa (Marple, 1987).

The equations in (2.27) can be written in matrix form as follows:

$$\underbrace{\begin{bmatrix} x_l \\ \vdots \\ x_m \\ \vdots \\ x_n \end{bmatrix}}_{\begin{pmatrix} \mathbf{d} \\ N \times 1 \end{pmatrix}} = \underbrace{\begin{bmatrix} x_{l+1} & x_{l+2} & \dots & \dots & x_{l+p} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ x_m & x_m & \ddots & \dots & x_m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n-1} & x_{n-2} & \vdots & \vdots & x_{n-p} \end{bmatrix}}_{\begin{pmatrix} \mathbf{A} \\ N \times P \end{pmatrix}} \times \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ \vdots \\ a_p \end{bmatrix}}_{\begin{pmatrix} \mathbf{m} \\ P \times 1 \end{pmatrix}} + \underbrace{\begin{bmatrix} \varepsilon_l \\ \varepsilon_{l+1} \\ \vdots \\ \vdots \\ \varepsilon_n \end{bmatrix}}_{\begin{pmatrix} \varepsilon \\ N \times 1 \end{pmatrix}} \tag{2.28}$$

I have an overdetermined system (where the number of observations is larger than the number of unknowns (Menke, 1989)) of equations which I will solve using the method of least squares with zero-order quadratic regularization (damped least squares method) (Lines and Treitel, 1984). In this case, I form the following cost function by using the l_2 norm:

$$J = ||\mathbf{Am} - \mathbf{d}||_2^2 + \mu||\mathbf{m}||_2^2 \,. \tag{2.29}$$

The first term of J indicates the modeling error or misfit. This term defines how well the prediction filter can reproduce the data. The second term is a stability or regularization term. The parameter μ is the trade-off parameter that accounts for the amount of weight given to each one of the terms in the cost function (Figure 2.2). The minimum of the cost function is found by taking derivatives with respect to the unknown parameters and setting them to zero. The solution, the damped least squares solution (or the minimum quadratic norm solution), is given by:

$$\mathbf{m} = (\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})^{-1}\mathbf{A}^T\mathbf{d} \tag{2.30}$$

where $\mathbf{I}$ denotes the identity matrix.

The main goal in linear prediction is to model the data with a small set of coefficients. These coefficients can be used to reconstruct (model) a clean version of the data (Canales, 1984) (see Chapter 4) , to compute the AR spectral estimator (Marple, 1980) and to design data compression algorithms (Makhoul, 1975). I are interested in the predictability of seismic events in the spatial domain, not only to

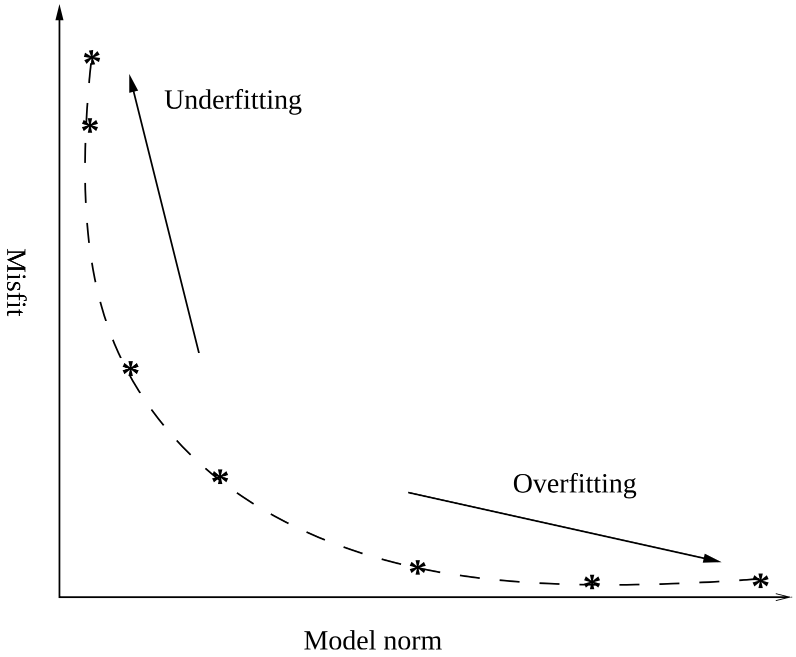

Figure 2.2: The trade-off parameter μ estimates the optimum solution. A large μ will underfit the data, otherwise a small amount of μ will overfit the data.

propose new noise attenuation strategies but also to design methods for optimal reconstruction (interpolation) of seismic waveforms. I will come to this point when focusing on $f - x$ processing.

2.4 1-D Synthetic and Real Data Examples

I have developed an algorithm to invert the coefficients of a first-order Volterra series. I focus on a 1-D synthetic time series which is generated with real AR data. Linear prediction methods provide predictions similar to Yule-Walker equations, Burg's algorithm, and a first-order Volterra series. Figure 2.3 (a) shows linear 1-D input data containing 100 samples.

Figures 2.3 (b) and 2.3 (c) represent the predicted series modeled via Yule-

Walker equations and Burg's method with parameter $p = 4$. Figures 2.3 (d) and 2.3 (e) portray predicted data using linear prediction theory ($p = 4$) and the associated modeling error (which is equivalent to the difference between the original data and the predicted data), respectively. It is clear that the three different prediction algorithms provide similar results. I did not plot differences between original data and these methods (Yule-Walker and Burg's) because the results were quite similar to those obtained in Figure 2.3 (e).

I also attempt to model data corresponding to the so called Arctic oscillation (AO)-a time series from 1950 to 1999 of sea level pressures. These data are used to characterize the long term variability of nonseasonal sea level oscillations (Thomson, 2004).

Figure 2.4 (a) shows nonlinear AO-a data for a period from 1950 to 1999. The data consist of 104 samples (3 observations per year-January, February, March). Figures 2.4 (b) and 2.4 (c) illustrate predicted AO values using Yule-Walker equations and Burg's algorithm with linear terms ($p = 14$). Figures 2.4 (d) and 2.4 (e) represent our attempt to model the data with a linear prediction filter ($p = 14$) and the modeling error, respectively. Again, it is clear that the dynamics of the time series cannot be captured by linear prediction methods.

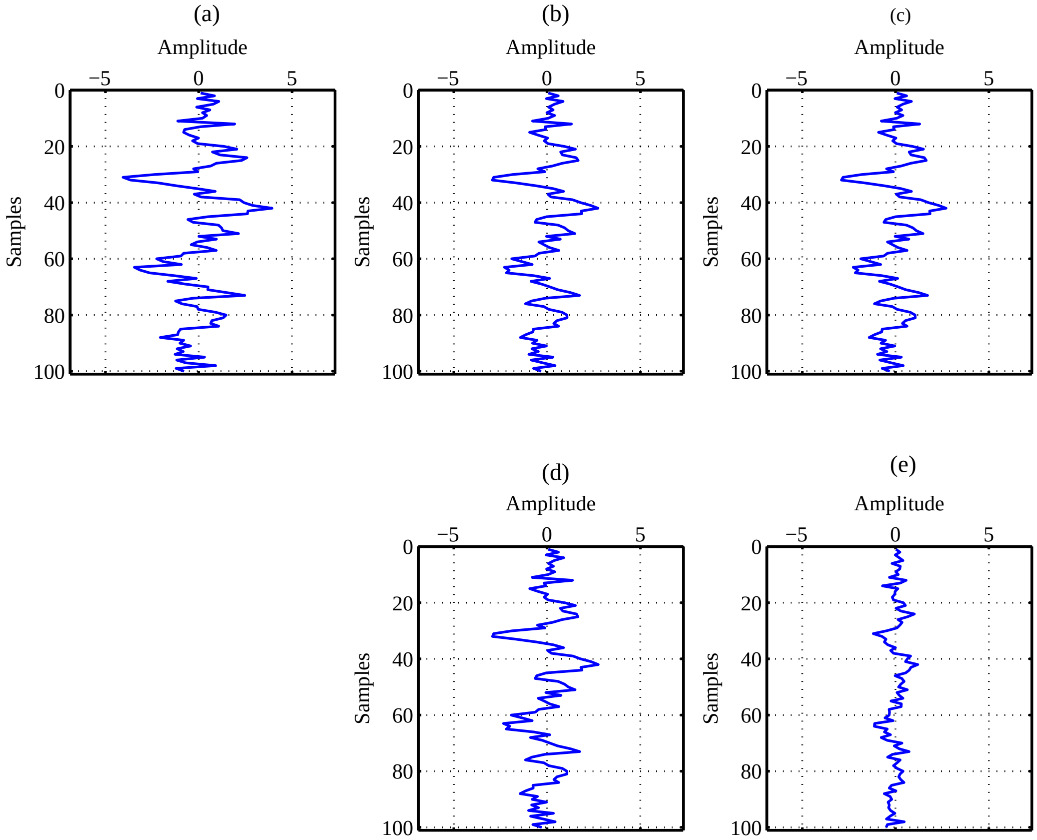

Figure 2.3: 1-D synthetic data for comparison of prediction between linear prediction theories. (a) Original data. (b) Prediction using Yule-Walker equations ($p = 4$). (c) Prediction using Burg's algorithm ($p = 4$). (d) Prediction using the first-order Volterra series which is equivalent to a linear prediction ($p = 4$). (e) The error between the original data and a linear prediction.

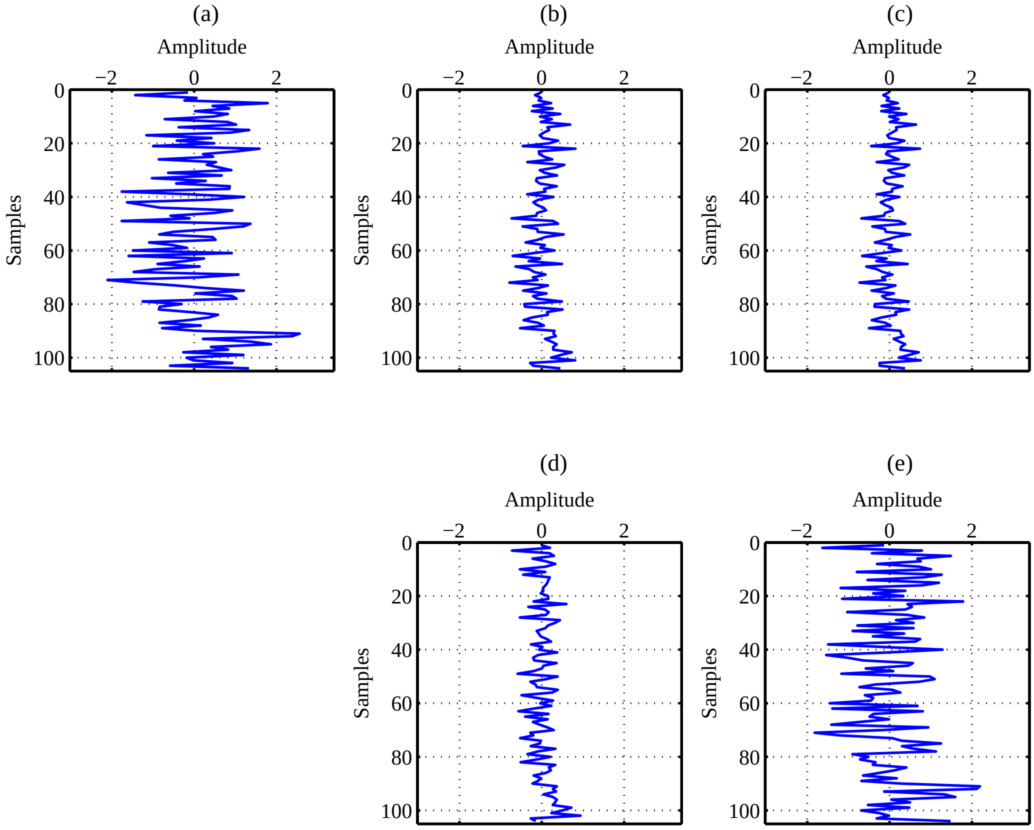

Figure 2.4: Arctic Oscillation data for standardized nonlinear sea-level pressures for comparison of prediction between linear prediction theories. (a) Original data. (b) Prediction using Yule-Walker equations ($p = 14$). (c) Prediction using Burg's algorithm ($p = 14$). (d) Prediction using the first order Volterra series which is equivalent to a linear prediction ($p = 14$). (e) The error between the original data and the linear prediction in (d).

2.5 Power Spectrum

The power spectrum density (PSD) estimation $C_2(\omega)$ (or $P_{xx}(\omega)$) is defined as:

$$C_2(\omega) = \sum_{\tau_1=-\infty}^{\infty} c_2(\tau) \exp(-i(\omega\tau)) \tag{2.31}$$

where $c_2(\tau)$ is the autocorrelation. Equation 2.31 is also known as the Wiener-Khintchine theorem (Nikias and Mendel, 1993).

At this point it is important to clarify that a non parametric estimator of the power spectrum can be computed using the Discrete Fourier Transform (DFT). In this case, the autocorrelation is estimated from the data and the DFT is used to evaluate an estimator of $P_{xx}(\omega)$ from equation 2.31 .

Parametric methods for Power spectrum estimation, on the other hand, operate by defining the power spectrum of the parametric model. In the case of AR process the power spectrum is given by

$$P_{AR}(z) = \frac{\sigma_\varepsilon^2}{|1 - (\sum_{j=1}^{p}) a_j z^j|^2} \tag{2.32}$$

where σ_ε^2 variance of the innovation sequence of the AR model. If $z = \exp(-i\omega)$ then $P_{AR}(z)$ becomes $P_{AP}(\omega)$. In general, I first compute the AR coefficients using one of the methods described in the preceding sections, then the coefficients are plugged in the formula for the power spectrum. The latter is the so called AR-spectrum, one form of parametric spectral analysis, often used because of its ability to produce smooth spectra from short time series. Figures 2.4(a)-(d) I compare the classical spectral density for the AO-a data computed with nonparametric analysis and AR analysis using Yule-Walker, Burg and the LS methods.

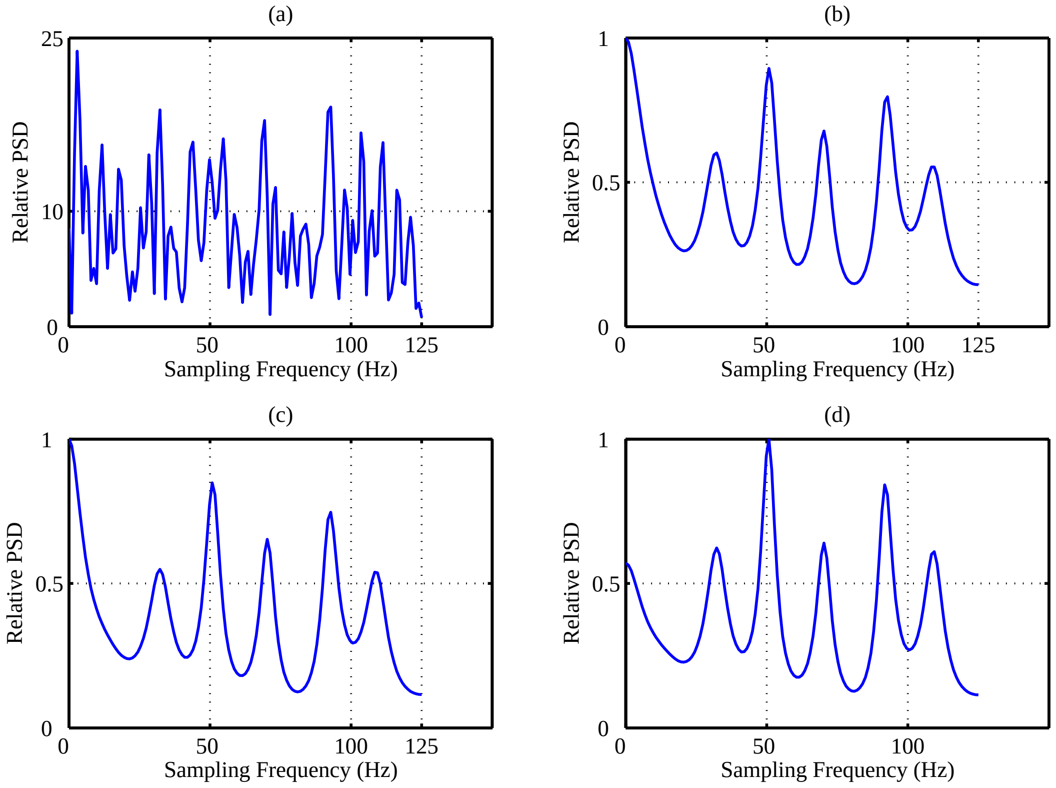

Figure 2.5: PSD Estimation of Arctic Oscillation data for standardized nonlinear sea-level pressures. (a) Nonparametric DFT based PSD estimation. (b) PSD AR estimation using Yule-Walker equations . (c) PSD AR estimation using Burg's algorithm. (d) PSD AR estimation using the first order Volterra series which is equivalent to a linear prediction .

2.6 Summary

In this chapter, I have covered some of the theoretical and practical aspects of linear prediction. Computation of AR coefficients with different AR models has been explored using Yule-Walker equations based on autocorrelation sequences solved with the Levinson algorithm which ensures a fast inversion of the Toeplitz matrix. In addition, Burg's method was used to directly estimate AR coefficients with backward and forward predictions by minimizing the error between the predicted data and original data. Finally, I presented a least squares method that uses only the available data and avoid truncation effects by properly using all the available information at the time of setting the system of linear prediction equations. Because of practical considerations, I will use the least squares approach presented in section 2.3.4 to solve for the coefficients of the nonlinear model that I will present in Chapter 3.

Chapter 3

Nonlinear Prediction

3.1 Nonlinear Processes via the Volterra Series: Background

The failure of linear systems (prediction techniques) to accurately model all physical systems leads to the creation of nonlinear prediction methods (Wiener, 1942; Bracalari and Salusti, 1994). Examples of nonlinear systems in which nonlinear modeling techniques have been applied range from communication to nonlinear interactions of waves (Coker and Simkins, 1980; Benedetto and Biglieri, 1983; Koh and Powers, 1985; Kim et al., 1994; Flioriani et al., 2000). The potential of nonlinear systems in seismic data processing, however, is relatively underutilized.

In this section the Volterra series will be introduced by extending a classical linear prediction technique to nonlinear prediction technique. The goal is to address the modeling waveforms with variable curvature in the $t-x$ domain with nonlinear prediction theory implemented via a Volterra series and provide a set of

AR techniques to address the modeling of complex waveforms in the $f - x$ domain. The approach here will be an extension of the linear AR concept to higher order dimensions for the applications of second and third-order Volterra kernels.

3.1.1 Time domain representation

I now consider a time series that arises from a nonlinear process and that requires a nonlinear modeling method to synthesize its input/output behavior. In addition, I assume a time-invariant system.

I can analyze the Volterra series as an expansion of the linear convolution integral. There is similarity between a Volterra series and a Taylor series. A Taylor series expands a nonlinear function as a superposition of simple polynomial functions. A Volterra series, on the other hand, can expand a system in terms of convolution-like integrals which are linear in the system impulse responses but nonlinear in the input signal (Schetzen, 1980; Cherry, 1994). The general expression for a continuous time-invariant Volterra system is given by:

$$\begin{aligned} y(t) &= \frac{1}{1!}\int_{-\infty}^{\infty} d\sigma_1 h_1(\sigma_1) x(t-\sigma_1) \\ &+ \frac{1}{2!}\int_{-\infty}^{\infty} d\sigma_1 \int_{-\infty}^{\infty} d\sigma_2 h_2(\sigma_1,\sigma_2) x(t-\sigma_1) x(t-\sigma_2) \\ &+ \frac{1}{3!}\int_{-\infty}^{\infty} d\sigma_1 \int_{-\infty}^{\infty} d\sigma_2 \int_{-\infty}^{\infty} d\sigma_3 h_3(\sigma_1,\sigma_2,\sigma_3) x(t-\sigma_1) x(t-\sigma_2) x(t-\sigma_3) \\ &+ \cdots \end{aligned} \tag{3.1}$$

where the first line is the first-order (linear), second line is the second-order (quadratic), and the third line is the third-order (cubic), etc. Notice that first-order Volterra series is equivalent to the convolution representation of a system. The last expression can be represented in a more general form as follows:

$$y(t) = \sum_{k=1}^{\infty} \frac{1}{k!} \int_{-\infty}^{\infty} d\sigma_1 \cdots \int_{-\infty}^{\infty} d\sigma_k h_k(\sigma_1, \sigma_2 \ldots, \sigma_k) \prod_{p=1}^{k} x(t - \sigma_p) \qquad (3.2)$$

where, again, $x(t)$ is the input, $y(t)$ is the output of the system. This functional form was first studied by the Italian mathematician Vito Volterra, so is known as the Volterra series, and the functions $h_k(\sigma_1, \ldots, \sigma_k)$ are known as Volterra kernels of the system. Norbert Wiener (1942) first applied these series to the study of nonlinear systems. As seen in equation (3.2) the Volterra series can be regarded as a nonlinear extension of the classical linear convolution.

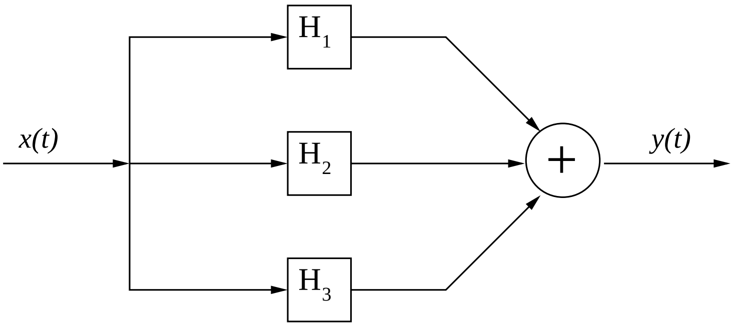

Figure 3.1: Schematic representation of a system characterized by a third-order Volterra series. Modified from Schetzen (1980). H_1, H_2 and H_3 represent the impulse responses of the first, second and third-order Volterra kernels, respectively.

Figure 3.1 illustrates a schematic representation of a system is characterized by a third-order Volterra series.

3.1.2 Frequency domain representation of Volterra kernels

The representation of a Volterra kernel in the Fourier domain is given by

$$H_k(\omega_1, \omega_2, \cdots, \omega_k) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_k(\sigma_1, \ldots, \sigma_k) e^{-i(\omega_1\sigma_1 + \omega_2\sigma_2 + \cdots + \omega_k\sigma_k)} d\sigma_1, \ldots, d\sigma_k. \tag{3.3}$$

The inverse Fourier transform of k^{th}-order Volterra kernels is as follows (Rugh, 1981)

$$h_k(\sigma_1, \sigma_2, \ldots, \sigma_k) = \frac{1}{(2\pi)^k} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} H_k(\omega_1, \cdots, \omega_k) e^{-i(\omega_1\sigma_1 + \omega_2\sigma_2 + \cdots + \omega_k\sigma_k)} d\omega_1, \ldots, d\omega_k. \tag{3.4}$$

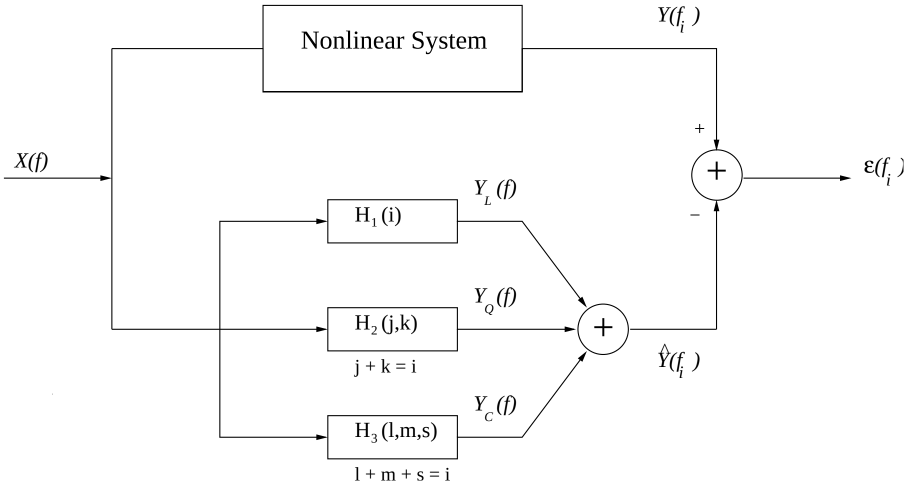

Figure 3.2: Frequency domain Volterra model of a cubic nonlinear system. After Nam and Powers (1994) and Schetzen (2006).

Diagram shown in Figure 3.2 represents a discrete frequency domain third- order Volterra model which is expressed as

$$
\begin{aligned}
Y(f_i) &= \underbrace{H_1(f_i)\,X(f_i)}_{Y_L(f_i)} \\
&\quad + \underbrace{\sum_{f_j+f_k=f_i} H_2(f_j,f_k)X(f_j)X(f_k)}_{Y_Q(f_i)} \\
&\quad + \underbrace{\sum_{f_l+f_m+f_s=f_i} H_3(f_l,f_m,f_s)\,X(f_l)X(f_m)X(f_s)}_{Y_C(f_i)} \\
&= \hat{Y}(f_i) + \varepsilon(f_i) \qquad (3.5)
\end{aligned}
$$

where $X(\cdot)$ and $Y(\cdot)$ are discrete FT's of input and output data, $\hat{Y}(\cdot) = Y_L(\cdot) + Y_Q(\cdot) + Y_C(\cdot)$ is the model output (prediction). ε_{f_i} denotes the difference between original and model output at a given frequency. $H_1(\cdot)$, $H_2(\cdot,\cdot)$, and $H_3(\cdot,\cdot,\cdot)$ are linear, quadratic, and cubic transfer functions of a Volterra series (Nam and Powers, 1994).

3.1.3 Symmetry property of Volterra kernels

Second-order and higher orders of Volterra kernels have symmetries properties that are provided in this section. If I rearrange equation (3.2) and interchange σ's, I have the following symmetry of kernels:

$$
y(t) = \sum_{k=1}^{\infty} \frac{1}{k!} \int_{-\infty}^{\infty} d\sigma_1 \cdots \int_{-\infty}^{\infty} d\sigma_k h_k^*(\sigma_2, \sigma_1 \ldots, \sigma_k) \prod_{p=1}^{k} x(t-\sigma_p). \qquad (3.6)
$$

where in this example σ_1 and σ_2 are switched because all integrations of the system are from $-\infty$ to $+\infty$ and $x(t-\sigma_1)x(t-\sigma_2) = x(t-\sigma_2)x(t-\sigma_1)$ so that equations (3.2) and (3.6) have the same value.

$k!$ is the total number of all possible integrations of k and this can be generalized for Volterra kernels:

$$h_k(\sigma_1, \sigma_2, \ldots, \sigma_k) = \frac{1}{k!} \sum_{i=1}^{k!} h_k^*(\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_k}) \tag{3.7}$$

where $i_1, \ldots, i_k$ shows the i^{th} permutation of 1,2,...,n (Schetzen, 2006). For instance, the second-order kernel for $k = 2$ is given by

$$h_2(\sigma_1, \sigma_2) = \frac{1}{2}(h_2^*(\sigma_1, \sigma_2) + h_2^*(\sigma_2, \sigma_1)). \tag{3.8}$$

The symmetrized kernel $h_2(\sigma_1, \sigma_2)$ and the asymmetric kernel $h_2^*(\sigma_1, \sigma_2)$ are shown in the example above. Any asymmetric kernel also can be symmetrized via the procedure outlined above. It can be seen from the equations above that the order of σ's is not important. The asymmetric form of the Volterra kernels is not unique, but the symmetric form is unique. I can demonstrate why the uniqueness is significant with an example of a second-order Volterra kernel as follows:

$$h_2(\sigma_1, \sigma_2) = h_a(\sigma_1)h_b(\sigma_2), \tag{3.9}$$

and the other kernel is

$$h_2^*(\sigma_1, \sigma_2) = h_a(\sigma_2)h_b(\sigma_1). \tag{3.10}$$

There are two asymmetric kernels, $h_a(\sigma_1)h_b(\sigma_2)$ and $h_a(\sigma_2)h_b(\sigma_1)$there is only one symmetric kernel, $\frac{1}{2}(h_a(\sigma_1)h_b(\sigma_2) + h_a(\sigma_2)h_b(\sigma_1))$.

If the kernel is unique the determination will be simplified for a given nonlinear system, and the uniqueness can be gained by demanding the kernel to be not asymmetric.

Equation (3.7) can be extended to the frequency domain kernels like

$$H_k(\omega_1, \omega_2, \ldots, \omega_k) = \frac{1}{k!} \sum_{i=1}^{k!} H_k^*(\omega_{i_1}, \omega_{i_2}, \ldots, \omega_{i_k}). \tag{3.11}$$

Using symmetry arguments for the reduction of the nonlinear prediction coefficients in the AR model will be discussed in the next section.

3.2 Nonlinear Modeling of Time Series via Volterra Kernels

I propose to replace the linear prediction problem by a nonlinear prediction problem. Our nonlinear problem is a Volterra system with an expansion in terms of three kernels obtained by truncating the third-order of the series: a linear or first-order kernel, a nonlinear quadratic kernel, and a nonlinear cubic kernel

$$\begin{aligned} y(t) &= \frac{1}{1!}\int_{-\infty}^{\infty} d\sigma_1 h_1(\sigma_1)x(t-\sigma_1) \\ &+ \frac{1}{2!}\int_{-\infty}^{\infty} d\sigma_1 \int_{-\infty}^{\infty} d\sigma_2 h_2(\sigma_1,\sigma_2)x(t-\sigma_1)x(t-\sigma_2) \\ &+ \frac{1}{3!}\int_{-\infty}^{\infty} d\sigma_1 \int_{-\infty}^{\infty} d\sigma_2 \int_{-\infty}^{\infty} d\sigma_3 h_3(\sigma_1,\sigma_2,\sigma_3)x(t-\sigma_1)x(t-\sigma_2)x(t-\sigma_3). \end{aligned} \tag{3.12}$$

I call this system a third-order Volterra system. A first-order Volterra system is the classical convolution integral used to described a linear time-invariant system.

Equation (2.9) (linear prediction model) is a time-invariant linear system where I have replaced the output by the one-step ahead prediction of the input. Similarly, I can consider equation (3.12) and construct a time-variant nonlinear prediction operator with a nonlinear Volterra model of the form (order of p, q, r):

$$\begin{aligned} x_n &= \sum_{i=1}^{p} a_i\, x_{n-i} \\ &+ \sum_{j=1}^{q}\sum_{k=1}^{q} b_{jk}\, x_{n-j}x_{n-k} \\ &+ \sum_{l=1}^{r}\sum_{m=1}^{r}\sum_{s=1}^{r} c_{lms}\, x_{n-l}x_{n-m}x_{n-s} \;+\; \varepsilon_n\,, \end{aligned} \tag{3.13}$$

where a_i, b_{jk}, and c_{lms} are the linear, the nonlinear quadratic, and the nonlinear cubic impulse responses of the nonlinear system, respectively, also known as Volterra kernels. The first term on the right hand side of equation (3.13) represents the classical linear prediction problem with a prediction operator of order p. The second and third terms of equation (3.13) represent the expansion of the signal in terms of quadratic and cubic nonlinearities. The modeling error is given by ε_n. Note that equation (3.13) is the nonlinear AR formulation as an extension of linear AR modeling.

The last expression can also be written in prediction error form

$$\begin{aligned} \varepsilon_n = x_n &- \sum_{i=1}^{p} a_i\, x_{n-i} \\ &- \sum_{j=1}^{q}\sum_{k=1}^{q} b_{jk}\, x_{n-j}x_{n-k} \\ &- \sum_{l=1}^{r}\sum_{m=1}^{r}\sum_{s=1}^{r} c_{lms}\, x_{n-l}x_{n-m}x_{n-s}\,. \end{aligned} \tag{3.14}$$

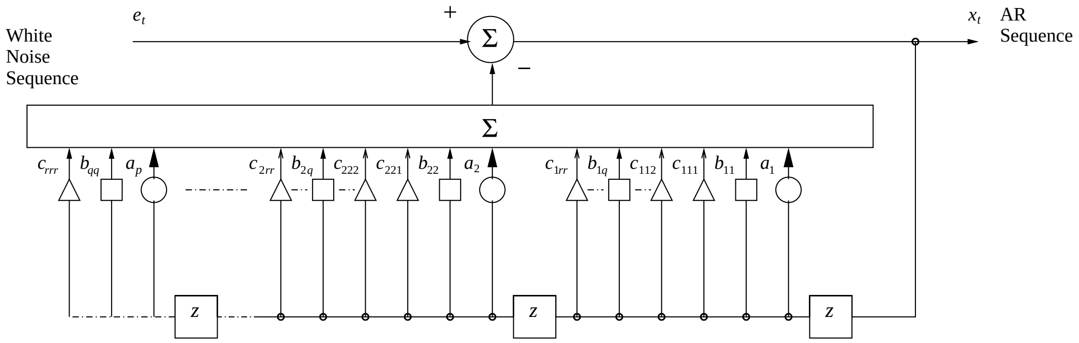

Figure 3.3: Volterra AR diagram. Modified from Marple (1987) and Ulrych and Sacchi (2005). The coefficients of the linear term of the Volterra series are given by $a_1, \cdots, a_p$, the coefficients of the quadratic term of the Volterra series are given by $b_{11}, \cdots, b_{qq}$, and the coefficients of the cubic term of the Volterra series are given by $c_{111}, \cdots, c_{rrr}$.

Equation (3.14) clearly states that I have more flexibility to model the deterministic part of the complex signal when the nonlinear terms are incorporated in the model.

The third-order Volterra representation of the data requires the estimation of $p + q^2 + r^3$ coefficients: a_i, $i = 1, \ldots, p$, b_{jk}, $j, k = 1, \ldots, q$, and c_{lms}, $l, m, s = 1, \ldots, r$. I will use symmetry properties of the Volterra series to reduce the number of coefficients of the quadratic contribution from q^2 to $(q\,(q + 3)/2 - q)$ and the number of coefficients of the cubic contribution from r^3 to $(r^2 + r!/(r - 3)!3!)$.

Table 3.1 shows various filter lengths and the number of prediction coefficients using symmetry arguments of a Volterra series. It can be seen that the number of parameters increases dramatically with a small increment in the filter order.

Again, I can use forward and backward prediction

$$x_n^f = \sum_{i=1}^{p} a_i^f \, x_{n-i} + \sum_{j=1}^{q} \sum_{k=1}^{q} b_{jk}^f \, x_{n-j} x_{n-k} + \sum_{l=1}^{r} \sum_{m=1}^{r} \sum_{s=1}^{r} c_{lms}^f \, x_{n-l} x_{n-m} x_{n-s} + \varepsilon_n \, , \quad (3.15)$$

Operator length	p	$\frac{q\,(q+3)}{2} - q$	$r^2 + \frac{r!}{(r-3)!3!}$	Total	q^2	r^3	$p{+}q^2{+}r^3$
1	1	1	1	3	1	1	3
2	2	3	4	9	4	8	14
3	3	6	10	19	9	27	39
4	4	10	20	34	16	64	84
5	5	15	35	55	25	125	155
10	10	55	220	285	100	1000	1110
20	20	210	1540	1770	400	8000	8420

Table 3.1: Filter length and the number of prediction coefficients. Various operator lengths versus linear, quadratic nonlinear, and cubic nonlinear prediction coefficients of a Volterra system using symmetry arguments and the original number of parameters in the system.

and

$$x_n^b = \sum_{i=1}^{p} a_i^b \, x_{n+i} + \sum_{j=1}^{q}\sum_{k=1}^{q} b_{jk}^b \, x_{n+j}x_{n+k} + \sum_{l=1}^{r}\sum_{m=1}^{r}\sum_{s=1}^{r} c_{lms}^b \, x_{n+l}x_{n+m}x_{n+s} + \varepsilon_n \, . \quad (3.16)$$

By expanding the previous equation I notice that:

$$\begin{aligned} x_n^f &= \underbrace{a_1^f x_{n-1} + a_2^f x_{n-2} + \cdots + a_p^f x_{n-p}}_{I} \\ &+ \underbrace{b_{11}^f x_{n-1}^2 + b_{12}^f x_{n-1}x_{n-2} + b_{21}^f x_{n-2}x_{n-1} + b_{22}^f x_{n-2}x_{n-2} + \cdots + b_{qq}^f x_{n-q}^2}_{II} \\ &+ \underbrace{c_{111}^f x_{n-1}^3 + c_{112}^f x_{n-1}^2 x_{n-2} + \cdots + c_{123}^f x_{n-1}x_{n-2}x_{n-3} + \cdots + c_{rrr}^f x_{n-r}^3}_{III} \\ &+ \varepsilon_n \, , \end{aligned} \quad (3.17)$$

and

$$\begin{aligned} x_n^b &= a_1^b x_{n+1} + a_2^b x_{n+2} \cdots + a_p^b x_{n+p} \\ &+ b_{11}^b x_{n+1}^2 + b_{12}^b x_{n+1} x_{n+2} + b_{21}^b x_{n+2} x_{n+1} + b_{22}^b x_{n+2} x_{n+2} + \cdots + b_{qq}^b x_{n+q}^2 \\ &+ c_{111}^b x_{n+1}^3 + c_{112}^b x_{n+1}^2 x_{n+2} + \cdots + c_{123}^b x_{n+1} x_{n+2} x_{n+3} + \cdots + c_{rrr}^b x_{n+r}^3 \\ &+ \varepsilon_n \, , \end{aligned} \tag{3.18}$$

where I, II, and III are the linear, quadratic nonlinear, and cubic nonlinear contributions, respectively. I can see from equations (3.17) and (3.18) that the quadratic (Powers et al., 1990) and cubic coefficients of the Volterra series must obey symmetry properties as mentioned in the subsection 3.1.3. For instance,

$$b_{12}^f x_{n-1} x_{n-2} = b_{21}^f x_{n-2} x_{n-1} . \tag{3.19}$$

It is clear that the number of coefficients to be computed is reduced from q^2 to $(q\,(q+3)/2-q)$. The cubic part of the Volterra series has also similar symmetry properties, however, the symmetry relations are more complicated than the quadratic part. For example,

$$c_{112}^f x_{n-1} x_{n-1} x_{n-2} = c_{121}^f x_{n-1} x_{n-2} x_{n-1} = c_{211}^f x_{n-2} x_{n-1} x_{n-1} \, , \tag{3.20}$$

$$\begin{gathered} c_{123}^f x_{n-1} x_{n-2} x_{n-3} = c_{132}^f x_{n-1} x_{n-3} x_{n-2} = c_{213}^f x_{n-2} x_{n-1} x_{n-3} = \\ c_{231}^f x_{n-2} x_{n-3} x_{n-1} = c_{312}^f x_{n-3} x_{n-1} x_{n-2} = c_{321}^f x_{n-3} x_{n-2} x_{n-1} \, . \end{gathered} \tag{3.21}$$

Symmetry arguments in equations (3.19, 3.20 and 3.21) are also valid for the backward prediction. The number of coefficients for the cubic part is reduced from r^3 to $r^2 + r!/(r-3)!3!$ and the total number of prediction coefficients is reduced form $p + q^2 + r^3$ to $p + (q\,(q+3)/2 - q) + r^2 + r!/(r-3)!3!$.

The form of the forward and backward prediction equations is now given by:

$$\begin{aligned} x_n^f &= a_1^f x_{n-1} + a_2^f x_{n-2} \cdots + a_p^f x_{n-p} \\ &+ b_{11}^f x_{n-1}^2 + 2b_{12}^f x_{n-1}x_{n-2} + 2b_{13}^f x_{n-1}x_{n-3} + \cdots + b_{qq}^f x_{n-q}^2 \\ &+ c_{111}^f x_{n-1}^3 + 3c_{112}^f x_{n-1}^2 x_{n-2} + \cdots + 6c_{123}^f x_{n-1}x_{n-2}x_{n-3} + \cdots + c_{rrr}^f x_{n-r}^3 \\ &+ \varepsilon_n\,, \end{aligned} \tag{3.22}$$

and

$$\begin{aligned} x_n^b &= a_1^b x_{n+1} + a_2^b x_{n+2} \cdots + a_p^b x_{n+p} \\ &+ b_{11}^b x_{n+1}^2 + 2b_{12}^b x_{n+1}x_{n+2} + 2b_{31}^b x_{n+1}x_{n+3} + \cdots + b_{qq}^b x_{n+q}^2 \\ &+ c_{111}^b x_{n+1}^3 + 3c_{112}^b x_{n+1}^2 x_{n+2} + \cdots + 6c_{123}^b x_{n+1}x_{n+2}x_{n+3} + \cdots + c_{rrr}^b x_{n+r}^3 \\ &+ \varepsilon_n\,. \end{aligned} \tag{3.23}$$

As in the linear prediction problem, I will assume that the Volterra coefficients (linear, quadratic, and cubic) are obtained using actual observations. For example, assume $p = 1$, $q = 2$, $c = 3$, and a time series of $N = 7$ points:

$$
\begin{aligned}
x_1 &= a_1x_2 + b_{11}x_2^2 + 2b_{12}x_2x_3 + b_{22}x_3^2 + c_{111}x_2^3 + \cdots + 3c_{112}x_2^2x_3 + \cdots + 6c_{123}x_2x_3x_4 + \varepsilon_1 \\
x_2 &= a_1x_3 + b_{11}x_3^2 + 2b_{12}x_3x_4 + b_{22}x_4^2 + c_{111}x_3^3 + \cdots + 3c_{112}x_3^2x_4 + \cdots + 6c_{123}x_3x_4x_5 + \varepsilon_2 \\
x_3 &= a_1x_4 + b_{11}x_4^2 + 2b_{12}x_4x_5 + b_{22}x_5^2 + c_{111}x_4^3 + \cdots + 3c_{112}x_4^2x_5 + \cdots + 6c_{123}x_4x_5x_6 + \varepsilon_3 \\
x_4 &= a_1x_5 + b_{11}x_5^2 + 2b_{12}x_5x_6 + b_{22}x_6^2 + c_{111}x_5^3 + \cdots + 3c_{112}x_5^2x_6 + \cdots + 6c_{123}x_5x_6x_7 + \varepsilon_4 \\
x_4 &= a_1x_3 + b_{11}x_3^2 + 2b_{12}x_3x_2 + b_{22}x_2^2 + c_{111}x_3^3 + \cdots + 3c_{112}x_3^2x_2 + \cdots + 6c_{123}x_3x_2x_1 + \varepsilon_4 \\
x_5 &= a_1x_4 + b_{11}x_4^2 + 2b_{12}x_4x_3 + b_{22}x_3^2 + c_{111}x_4^3 + \cdots + 3c_{112}x_4^2x_3 + \cdots + 6c_{123}x_4x_3x_2 + \varepsilon_5 \\
x_6 &= a_1x_5 + b_{11}x_5^2 + 2b_{12}x_5x_4 + b_{22}x_4^2 + c_{111}x_5^3 + \cdots + 3c_{112}x_5^2x_4 + \cdots + 6c_{123}x_5x_4x_3 + \varepsilon_6 \\
x_7 &= \underbrace{a_1x_6}_{\textbf{Linear}} + \underbrace{b_{11}x_6^2 + 2b_{12}x_6x_5 + b_{22}x_5^2}_{\textbf{Quadratic}} + \underbrace{c_{111}x_6^3 + \cdots + 3c_{112}x_6^2x_5 + \cdots + 6c_{123}x_6x_5x_4}_{\textbf{Cubic}} + \varepsilon_7
\end{aligned}
\tag{3.24}
$$

or in matrix form:

$$
\underbrace{\begin{bmatrix} x_l \\ \vdots \\ x_m \\ \vdots \\ x_n \end{bmatrix}}_{\begin{pmatrix} \mathbf{d} \\ N \times 1 \end{pmatrix}}
=
\underbrace{\begin{bmatrix} \\ \textbf{Linear} \;\big|\; \textbf{Quadratic} \;\big|\; \textbf{Cubic} \\ \\ \end{bmatrix}}_{\begin{pmatrix} \mathbf{A} \\ N \times (P + \frac{Q(Q+3)}{2} - Q + R^2 + \frac{R!}{(R-3)!3!}) \end{pmatrix}}
\times
\underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_p \\ b_{11} \\ \vdots \\ b_{qq} \\ c_{111} \\ \vdots \\ c_{rrr} \end{bmatrix}}_{\mathbf{m}}
+
\underbrace{\begin{bmatrix} \varepsilon_l \\ \varepsilon_{l+1} \\ \vdots \\ \vdots \\ \varepsilon_n \end{bmatrix}}_{\begin{pmatrix} \varepsilon \\ N \times 1 \end{pmatrix}}
\tag{3.25}
$$

where **Linear, Quadratic,** and **Cubic** denote the matrices of linear, quadratic, and cubic filter coefficients respectively:

$$\textbf{Linear} = \begin{bmatrix} x_{l+1} & x_{l+2} & \cdots & x_{l+p} \\ \vdots & \ddots & \vdots & \vdots \\ x_m & x_m & \cdots & x_m \\ \vdots & \vdots & \vdots & \vdots \\ x_{n-1} & x_{n-2} & \cdots & x_{n-p} \end{bmatrix}, \tag{3.26}$$

$$\textbf{Quadratic} = \begin{bmatrix} x_{l+1}^2 & 2x_{l+1}x_{l+2} & 2x_{l+1}x_{l+3} & \cdots & x_{l+q}^2 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ x_m^2 & 2x_m x_m & 2x_m x_m & \cdots & x_m^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n-1}^2 & 2x_{n-1}x_{n-2} & 2x_{n-1}x_{n-3} & \cdots & x_{n-q}^2 \end{bmatrix}, \tag{3.27}$$

and

$$\textbf{Cubic} = \begin{bmatrix} x_{l+1}^3 & \cdots & 3x_{l+1}^2 x_{l+2} & 3x_{l+1}^2 x_{l+3} & \cdots & 6x_{l+1}x_{l+2}x_{l+3} & \cdots & x_{l+r}^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_m^3 & \cdots & 3x_m^2 x_m & 3x_m^2 x_m & \cdots & 6x_m x_m x_m & \cdots & x_m^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n-1}^3 & \cdots & 3x_{n-1}^2 x_{n-2} & 3x_{n-1}^2 x_{n-3} & \cdots & 6x_{n-1}x_{n-2}x_{n-3} & \cdots & x_{n-r}^3 \end{bmatrix}. \tag{3.28}$$

The unknown vector **m** contains linear and nonlinear prediction coefficients organized in lexicographic form in the dimension of $P + \frac{Q(Q+3)}{2} - Q + R^2 + \frac{R!}{(R-3)!3!}) \times 1$.

It is clear that the problem is linear in the coefficients. One possible solution vector is given by the regularized least squares solution (damped least squares). Adopting the least squares method with zero-order quadratic regularization leads to the solution of the filter coefficients:

$$\mathbf{m} = (\mathbf{A}^T\mathbf{A}) + \mu\mathbf{I})^{-1}\mathbf{A}^T\mathbf{d}\ . \tag{3.29}$$

For small systems (small order of parameters p, q, and r) one can use direct inversion methods to solve equation (3.29). For systems that involve long operators, I suggest the use of semi-iterative solvers like the method of conjugate gradients (Wang and Treitel, 1973). In our examples, however, I have adopted direct inversion methods. In general, $f - x$ algorithms do not require the inversion of large systems of equations. In many cases this is a consequence of working with small spatial windows.

3.3 1-D Synthetic and Real Data Examples

I have developed an algorithm to invert the coefficients of a third-order Volterra series. I focus on a 1-D synthetic time series which is generated with a real second-order Volterra system. Figures 3.4(a) and 3.5(a) show a 1-D input data for 100 samples.

Figures 3.4(b) and 3.4(c) represent the predicted series modeled via the third-order Volterra series (with parameters $p = 8$, $q = 8$, and $r = 8$) and the associated modeling error, respectively. Figures 3.4(d) and 3.4(e) portray the predicted data using linear prediction theory and modeling error ($p = 8$, $q = 0$, and $r = 0$). It is clear that quadratic and cubic terms are needed to capture the strong variability observed in the time series.

In Figure 3.5(c) I portray the contribution due to the linear terms of a third-order Volterra series to predicted the data in Figure 3.5(b) due to the linear terms of third order Volterra series. Note that this contribution is negligible and shows that linear terms can not model the highly nonstationary part of signal. Figures 3.5(d) and 3.5(e) illustrate the parts of the prediction associated with quadratic ($q = 8$) and cubic ($r = 8$) terms in the third-order Volterra series. Figure 3.5(f) shows the contribution of nonlinear terms associated with quadratic and cubic terms ($q = 8$ and $r = 8$). It is clear that nonlinear quadratic and cubic terms are required to properly model the full aperture.

A real data example corresponding to the so called Arctic oscillation time series (AO)-a time series from 1950 to 1999 of sea level pressures-is used to characterize the long term variability of nonseasonal sea level oscillations (Thomson, 2004).

Figures 3.6(a) and 3.7(a) show the nonlinear AO data for the period from 1950 to 1999. The data consist of 104 samples (3 observations per year-January, February - March). Figures 3.6(b) and 3.6(c) illustrate predicted AO values for a Volterra system consisting of linear and nonlinear terms ($p = 10$, $q = 10$, and $r = 10$) and associated error, respectively. Figures 3.6(d) and 3.6(e) represent our attempt to model the data with a linear prediction filter ($p = 10$) and corresponding error. Again, it is clear that the dynamics of the time series is better captured by the third-order Volterra system.

Figure 3.7(b) is the prediction using a third-order Volterra series with $p = 10$, $q = 10$, and $r = 10$. In Figure 3.7(c) I portray the part of the prediction attributed

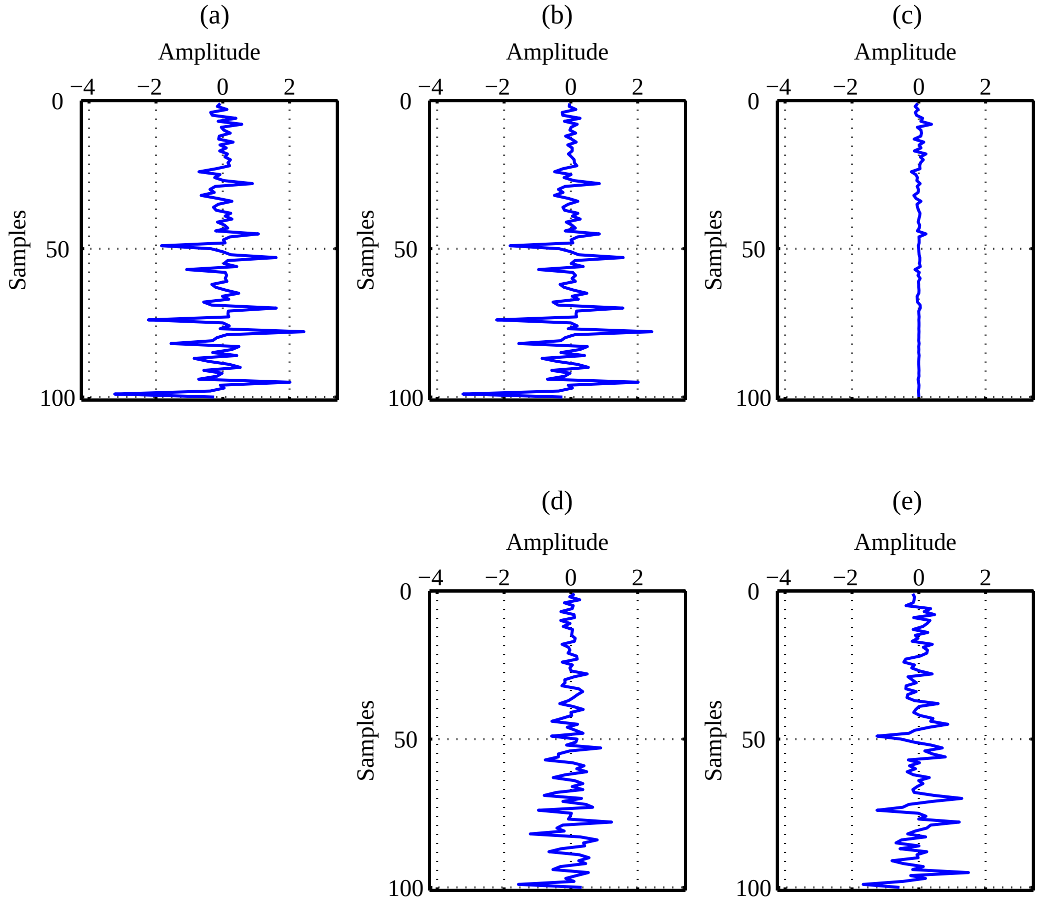

Figure 3.4: 1-D synthetic data for comparison of prediction between linear prediction theory and third order Volterra series. (a) Original data. (b) Prediction using a third-order Volterra series ($p = 8$, $q = 8$, and $r = 8$). (c) The error between the original data and the third-order Volterra prediction. (d) Prediction using the first order Volterra series, which is equivalent to linear prediction ($p = 8$). (e) The error between the original data and linear prediction.

to the linear kernel ($p = 10$). Figures 3.7(d) and 3.7(e) show the parts of the prediction associated with quadratic ($q = 10$) and cubic ($r = 10$) terms in the third-order Volterra series. In addition, Figure 3.7(f) illustrates the part of the

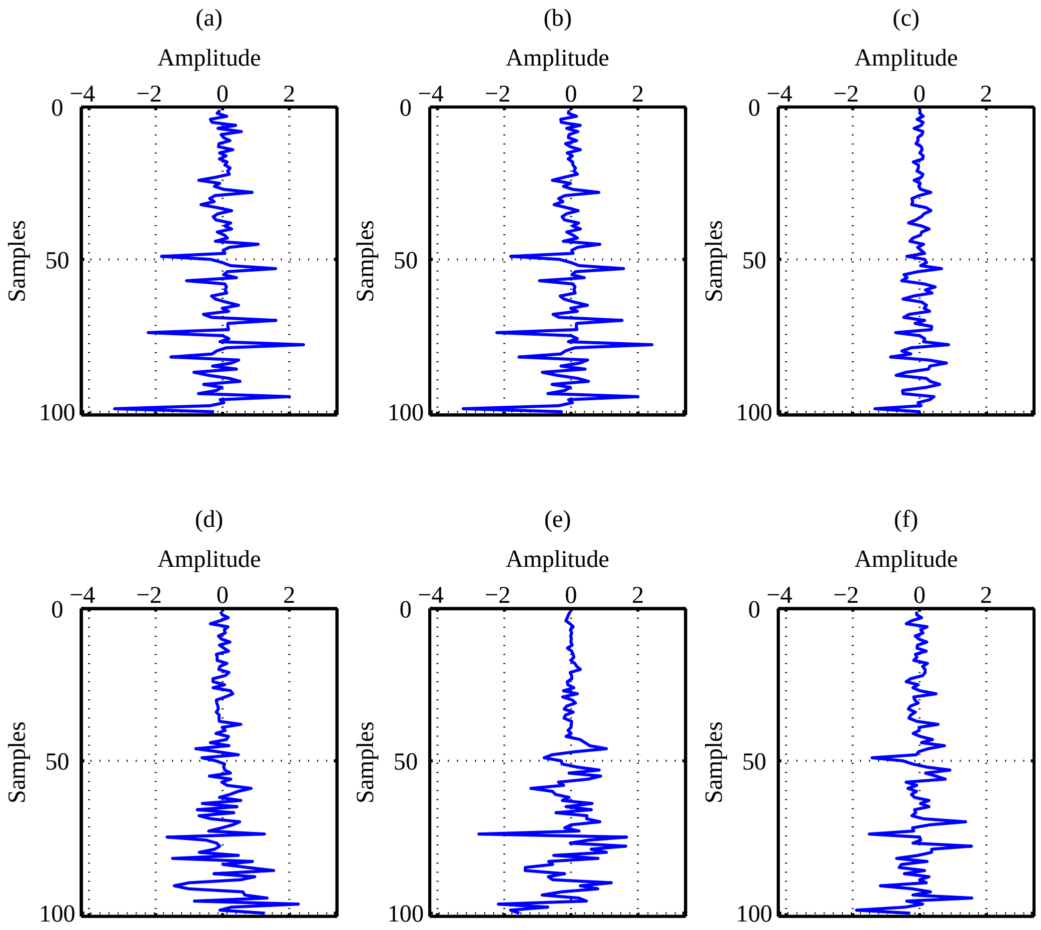

Figure 3.5: 1-D synthetic data. (a) Original data. (b) Prediction using a third-order Volterra series with parameters $p = 8$, $q = 8$, and $r = 8$. (c) Contribution from the linear part. (d) Contribution from the quadratic part. (e) Contribution from the cubic part. (f) Contribution from both quadratic and cubic parts ($q = 8$, and $r = 8$).

prediction associated to both quadratic and cubic terms ($q = 10$ and $r = 10$).

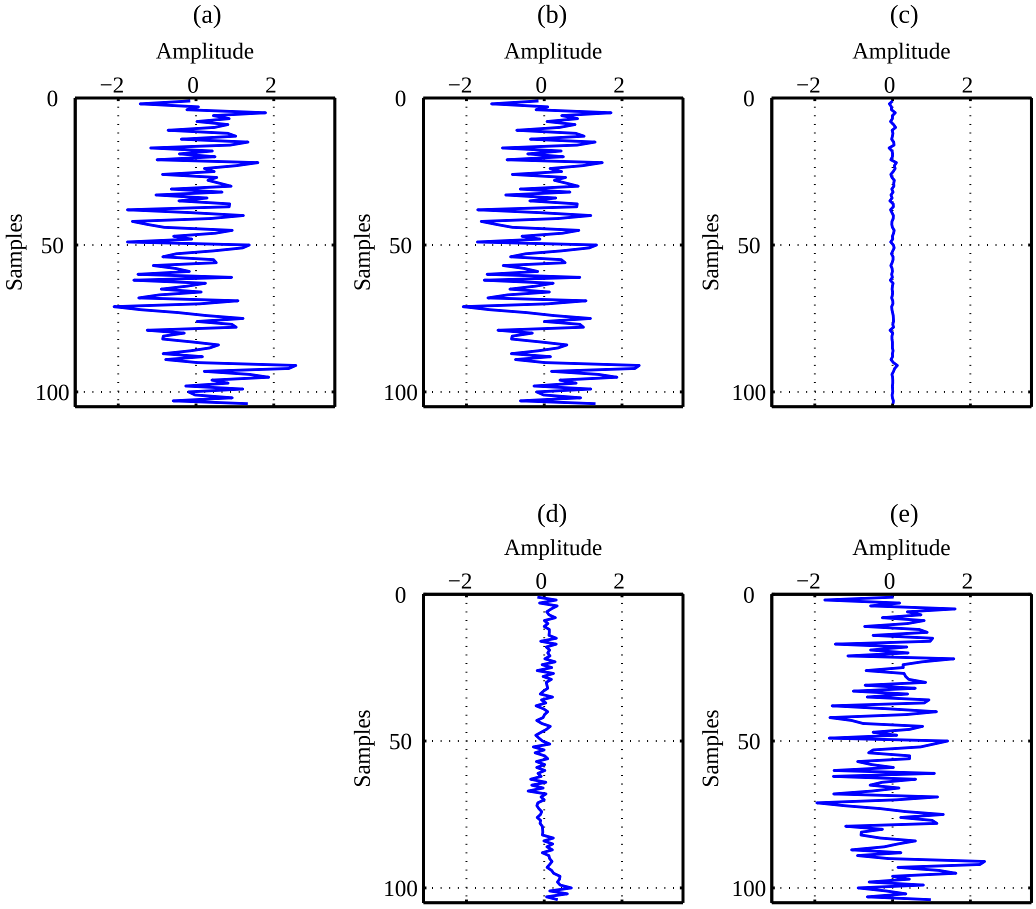

Figure 3.6: Arctic Oscillation data for standardized nonlinear sea-level pressures for comparison of prediction between linear prediction theory and third order Volterra series. (a) Original data. (b) Prediction using a third-order Volterra series ($p = 10$, $q = 10$, and $r = 10$). (c) The error between the original data and the third-order Volterra prediction. (d) Prediction using the first-order Volterra series, which is equivalent to linear prediction ($p = 10$). (e) The error between the original data and a linear prediction.

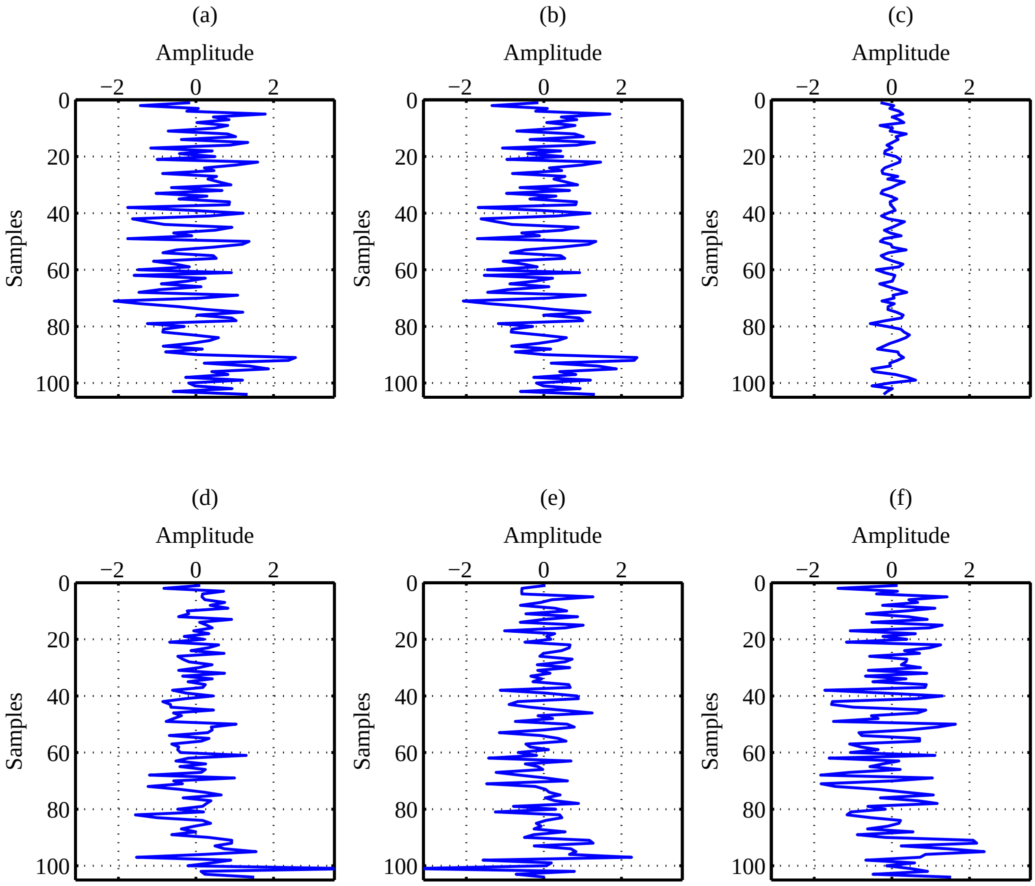

Figure 3.7: Arctic Oscillation data for standardized nonlinear sea-level pressures. (a) Original data. (b) Prediction using a third-order Volterra series with parameters $p = 10$, $q = 10$, and $r = 10$. (c) Contribution from the linear part. (d) Contribution from the quadratic part. (e) Contribution from the cubic part. (f) Contribution from both quadratic and cubic parts ($q = 10$, and $r = 10$).

3.4 Summary

In this chapter, I have covered theoretical and practical aspects of linear and nonlinear systems; particularly, Volterra series and the nonlinear prediction model based on a third-order Volterra series was presented. Linear and nonlinear autoregressive models have been explored. Volterra kernel parameters are obtained via a least squares inversion method. Real and synthetic data examples illustrate these methods. Linear and nonlinear time series modeling of 1-D data has been presented.

Chapter 4

Nonlinear Modeling of Complex Waveforms in the $f - x$ Domain

4.1 Linear Prediction in the $f - x$ Domain

F - X deconvolution is a popular noise attenuation tool first introduced by Canales (1984). Seismic traces are represented in the time-space domain. When one transforms each trace into a Fourier domain, the complex waveforms are represented in the frequency-space domain ($f - x$). Linear events are predicted for each frequency in the spatial direction of a given frequency. Therefore, if a signal can be predicted, the difference between the observed and predicted signals can be considered an estimation of the noise in the data.

Linear prediction filters that map to monochromatic complex sinusoids in the $f - x$ domain can accurately predict linear events in the $t - x$ domain. A superposition of complex harmonics immersed in white noise can be predicted using an ARMA (autoregressive-moving average) model as suggested by Sacchi and Kuehl (2001).

An ARMA model can be approximated by a long autoregressive model (AR), which turns out to be a representation of the linear prediction problem. In summary, linear events in $t-x$ space transform into complex sinusoids in the $f-x$ domain and linear prediction filtering can properly model the spatial variability of the waveforms at any given monochromatic temporal frequency f.

The seismic signal is considered to be an AR model; let us assume a single waveform in time domain. In addition, let's assume that the signal has linear moveout

$$s(x,t) \;=\; a(t-x\theta) \tag{4.1}$$

where x is offset of the trace, t is the time and θ is the slowness of the event. In the frequency domain this signal becomes

$$S(x,f) \;=\; A(f)e^{-i2\pi f\theta x} \tag{4.2}$$

where $A(f)$ denotes the source spectrum and f is the temporal frequency for x. By discretizing $x=(j-1)\delta x, \quad f=f_l$

$$S_{jl} \;=\; A_l e^{-i2\pi f_l\theta(j-1)\delta x} \tag{4.3}$$

I can develop this model as a function of wave number by fixing $k_l = 2\pi f_l\theta$ equation (4.3) becomes

$$S_{jl} \;=\; A_l e^{-ik_l(j-1)\delta x} \tag{4.4}$$

I can define the problem by predicting data along the each trace to fix temporal frequency:

$$S_j = A\, e^{-ik(j-1)\delta x} \qquad j = 1, \dots, N\,. \tag{4.5}$$

One can write a linear event in the $f - x$ domain as a one-step-ahead prediction given by:

$$\begin{aligned} S_{j-1} &= A\, e^{-ik(j-2)\delta x} \\ &= A\, e^{-ik(j-1)\delta x}\, e^{ik\delta x} \\ &= S_j\, e^{ik\delta x} \end{aligned} \tag{4.6}$$

I can write a recursive form for the prediction of the signal recorded at receiver j as a function of the signal at receiver $j - 1$ (along the spatial variable x) as follows:

$$\begin{aligned} S_j &= S_{j-1}\, e^{-ik\delta x} \\ &= a\, S_j \end{aligned} \tag{4.7}$$

The equation above is the basis for the $f - x$ prediction/deconvolution and SNR enhancement (Canales, 1984; Gulunay, 1986; Sacchi and Kuehl, 2001). Similarly, it can be proved that the superposition of p linear events (p complex harmonics) is in a recursive form:

$$\begin{aligned} S_j &= \sum_{n=1}^{p} A\, e^{-ik_n(j-1)\delta x} \\ &= \underbrace{a_1 S_{j-1} + a_2 S_{j-2} + \cdots + a_p S_{j-p}}_{Recursion\, of\, order\, p} \end{aligned} \tag{4.8}$$

The coefficients of the recursion are also called prediction error coefficients when related to the wave number of each linear event. These coefficients can be found using a least squares solution as presented in Chapters 2 and 3.

The $f-x$ domain noise prediction algorithm can simply be summarized to predict both data and noise as follows:

- Original data in $t-x$,
- Transform data to the $f-x$ domain,
- For each frequency f,
- Find $\widehat{\mathbf{m}}$ that solves $\mathbf{d}$=$\mathbf{Am}$+ $\mathbf{e}$,
- Use $\widehat{\mathbf{m}}$ to predict data and noise,
  $\widehat{\mathbf{d}} = \mathbf{A}\widehat{\mathbf{m}}$
  $\hat{\mathbf{e}} = \mathbf{d}$-$\widehat{\mathbf{d}}$
- Transform back to the $t-x$ domain.

4.2 Analysis of Optimum Filter Length

In a AR model of order (p), the best filter length p is not usually known. To continue our analysis I will define two measures of goodness of fit. For that purpose

I first define the observed data as ($\mathbf{D}$), the noise free data or clean data ($\mathbf{D}_c$) and the predicted data ($\mathbf{D}_p$) . I now define the following two measures of goodness of fit:

$$RMSE_1 = \sqrt{\frac{||\mathbf{D} - \mathbf{D_p}||^2}{nx \times nt}} \tag{4.9}$$

$$RMSE_2 = \sqrt{\frac{||\mathbf{D_c} - \mathbf{D_p}||^2}{nx \times nt}} \tag{4.10}$$

where $nx \ \times \ nt$ denotes the dimensions of the data: length of time series (nx) by number of traces (nt), respectively. $RMSE_2$ is not computable for real cases but it can be used gain understanding about the problem of order selection. The best (optimum) operator length is given for the operator that minimizes $RMSE_2$ (Figure 4.1(a)). On the other hand, $RMSE_1$ shows that increasing the filter length leads to a decrease of error that is only accounted by for fitting the noise (Figure 4.1(b)).

In Figure 4.2(a), 2-D synthetic data consisting of three linear events yields predictions for different filter lengths. The signal $\mathbf{D}_c$ is contaminated with additive noise (Figure 4.2(b)) and the signal-to-noise ratio (SNR) has been taken as 4. Figure 4.3(a) corresponds to the prediction of data presented in Figure 4.2(b) with parameter $p = 3$. Noise is rejected but the data are also poorly modeled (Figure 4.3(a)). The prediction of linear events in the prediction panel is not satisfactory because a large amount of energy leaks to the noise panel (Figure 4.3(b)). The prediction with a filter length $p = 6$ provides good noise rejection and the data is properly modeled (Figure 4.4(a)). The noise panel contains a small amount of coherent en-

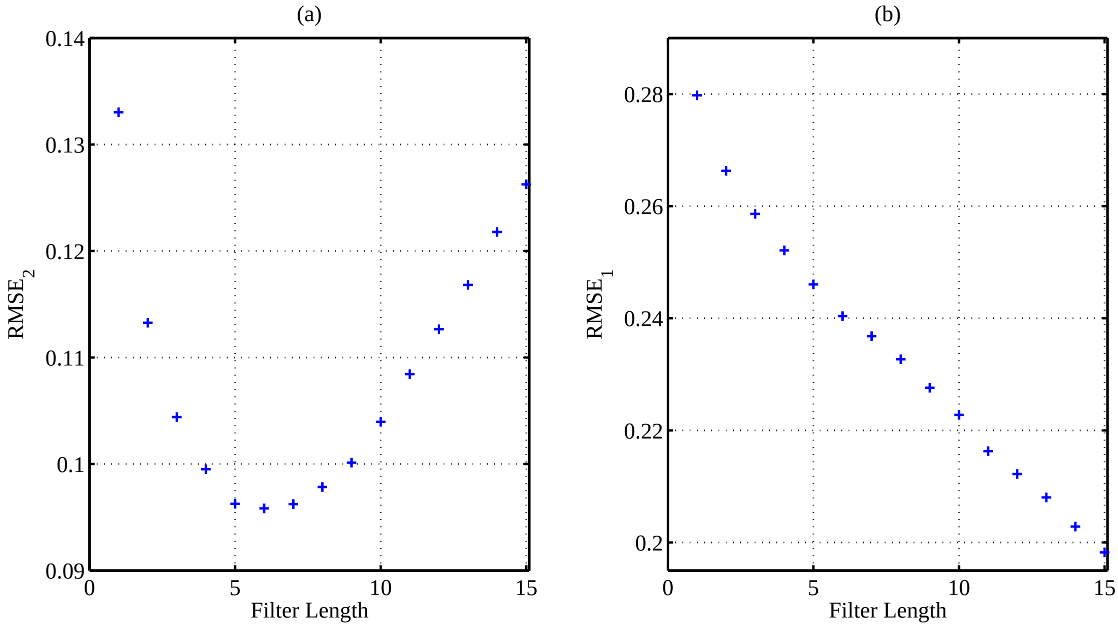

Figure 4.1: Optimum filter length for the data in Figure 4.2. (a) $RMSE_2$. (b) $RMSE_1$.

ergy (Figure 4.4(b)). Note that this figure shows the prediction of the optimum filter length determined with $RMSE_2$ in Figure 4.1(a). The minimum value for the $RMSE$ in that example is 6. The prediction for filter length $p = 15$ is good but the result is not as good as in the optimum case with $p = 6$ (Figure 4.5(a)). Noise is also modeled and incorporated in the predicted data (Figure 4.5(b)). At this point some inferences are in order. First, it is clear that the filter length is very important both to model the data and to reject the noise. However, it is not possible to compute $RMSE_2$ for real situations; it can be used to estimate optimum filter length when working with synthetic examples where one has accessed to data free of noise. Practical experience in seismic data processing shows that one can easily compute the optimum filter length by trying different lengths and observe the amount of coherent energy left in the error panel. At some point one can find a filter length that models the data and produce an error panel containing an important amount of incoherent energy.

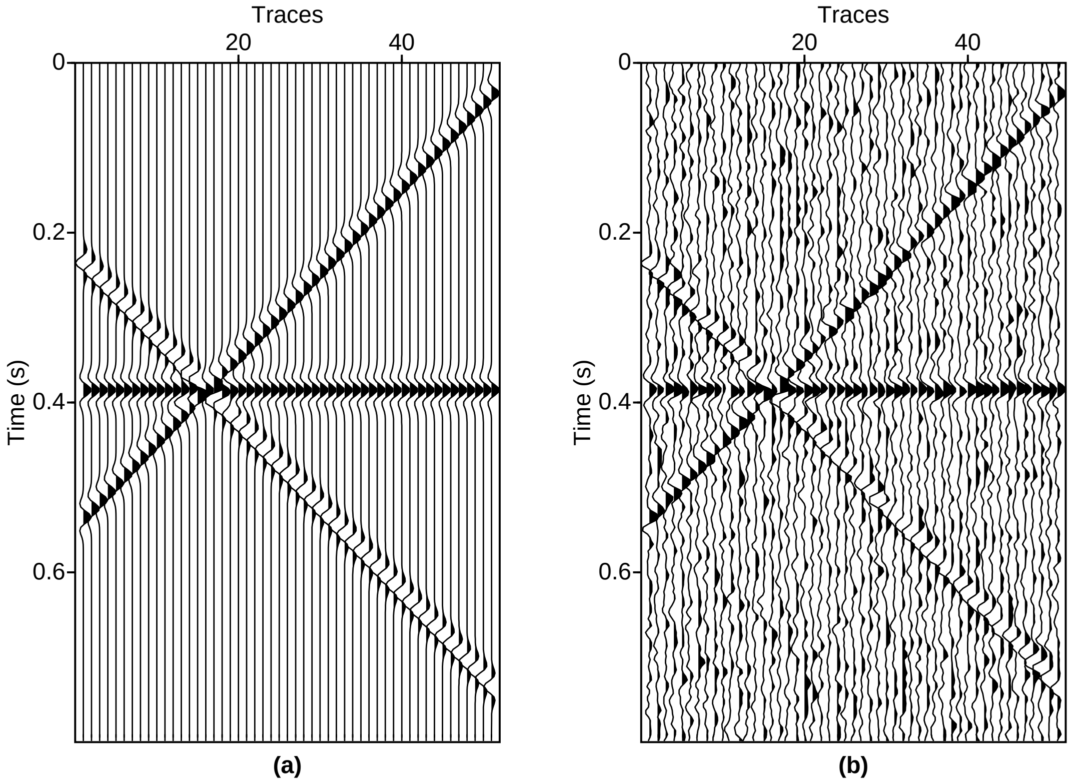

Figure 4.2: Synthetic data example for different filter lengths. (a) Original noise free data. (b) Data contaminated with additive noise ($SNR = 4$).

4.3 Nonlinear Prediction of Complex Waveforms in the $f - x$ Domain

Analysis of the linear events presented above is not valid for events that exhibit curvature in the $t - x$ domain. These events map to the $f - x$ domain as chirp-like

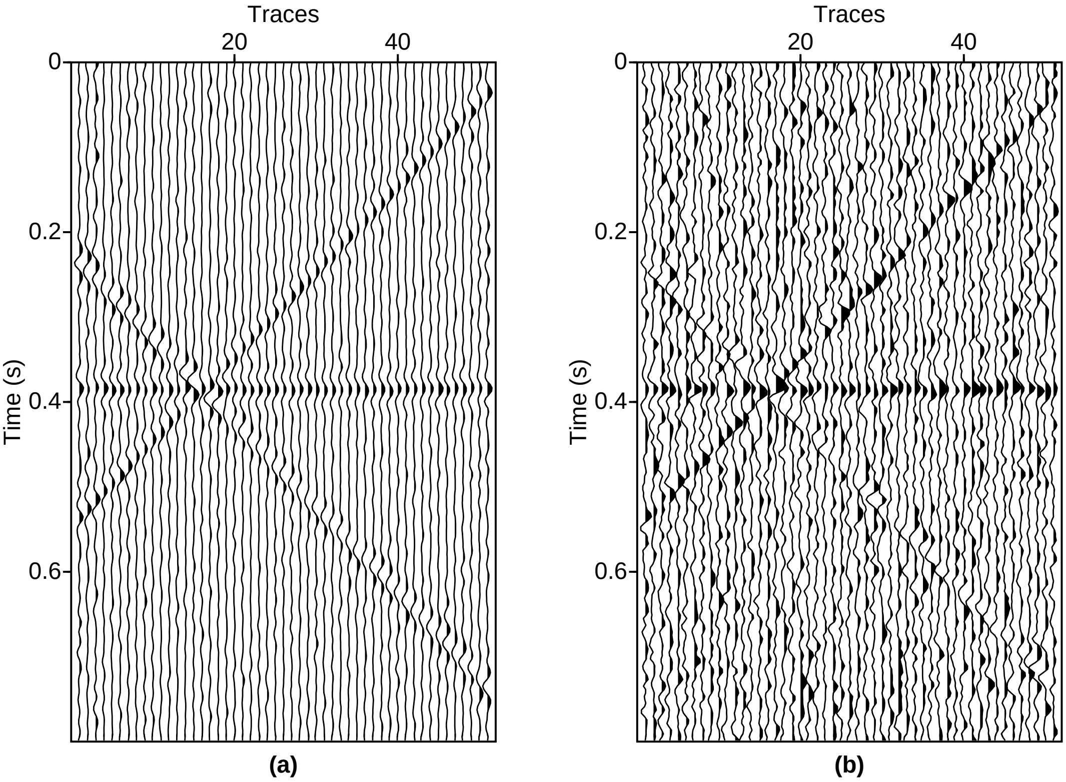

Figure 4.3: (a) Prediction of Figure 4.2(b) ($p = 3$). (b) The error between original data and predicted data.

signals. It is clear that linear prediction will fail in modeling such events. A solution to the problem is to use linear prediction techniques in small windows or to resort to nonstationary linear prediction operators (Sacchi and Kuehl, 2001).

I chose a 2-D synthetic data example consisting of 5 different hyperbolic events. The example does not satisfy the $f - x$ assumption of constant ray parameter waveforms in the $t - x$ domain. In this case I do not have a superposition of complex exponentials in the $f - x$ domain. Therefore the minimum value for the filter length ($p = 5$) in Figure 4.7(a) will be an approximated value for the hyperbolic event presented in Figure 4.6(b). Again, the data in Figure 4.6(a) is contaminated

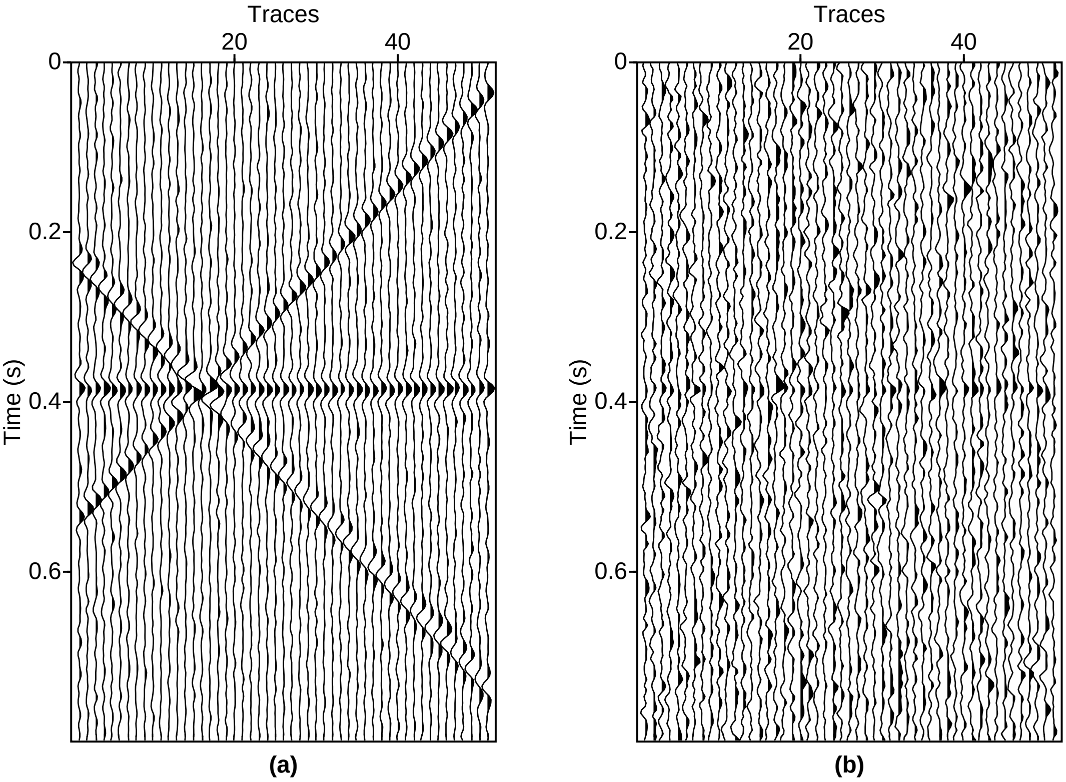

Figure 4.4: (a) Prediction of Figure 4.2(b) ($p = 6$). (b) The error between original data and predicted data.

with additive noise. The prediction for different filter lengths cannot model the data ($p = 3, 5$, and 15) in Figures 4.8, 4.9, and 4.10; $p = 5$ rejects noise but cannot model the data; $p = 15$ models the data better than the optimum filter length but it is also not a perfect solution because it overfits noise in the prediction panel (Figure 4.10(b)).

Events with nonlinear moveout can be modeled with a Volterra series. I begin by considering equation (4.8) as a Volterra series expansion by appending nonlinear coefficients. Remember that although the data vector $\mathbf{m}$ in equation (3.25) contains linear and nonlinear prediction coefficients, the problem is linear in the coefficients.

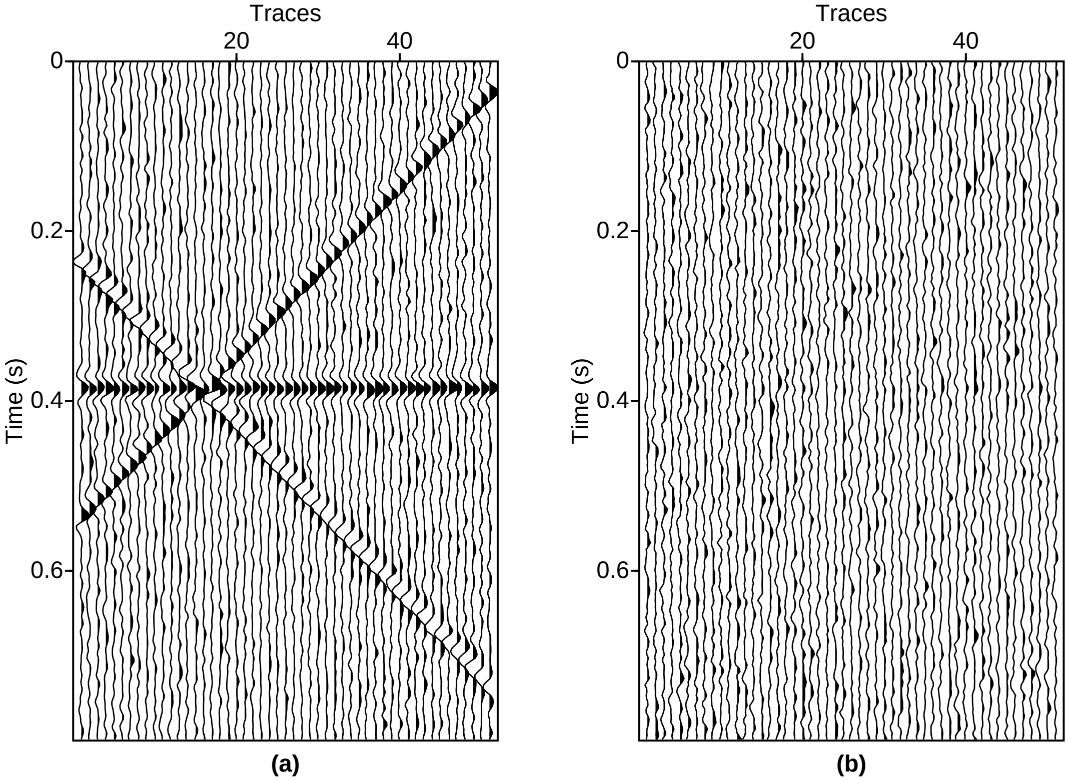

Figure 4.5: (a) Prediction of Figure 4.2(b) ($p = 15$). (b) The error between original data and predicted data.

Equation (4.8) will be changed for the $f - x$ nonlinear predictions as an expression with terms of the following form as an illustration:

$$S_j = a\, S_{j-1} \,+\, b\, S_{j-1} S_{j-2} \,+\, c\, S_{j-1} S_{j-2} S_{j-3}\,. \tag{4.11}$$

I explored the feasibility of using linear plus nonlinear quadratic and nonlinear cubic prediction filters to model waveforms that exhibit moveout curves that are not linear with synthetic and real data examples in the next section. The idea is, again, to operate in the $f - x$ domain with one temporal frequency at a time and

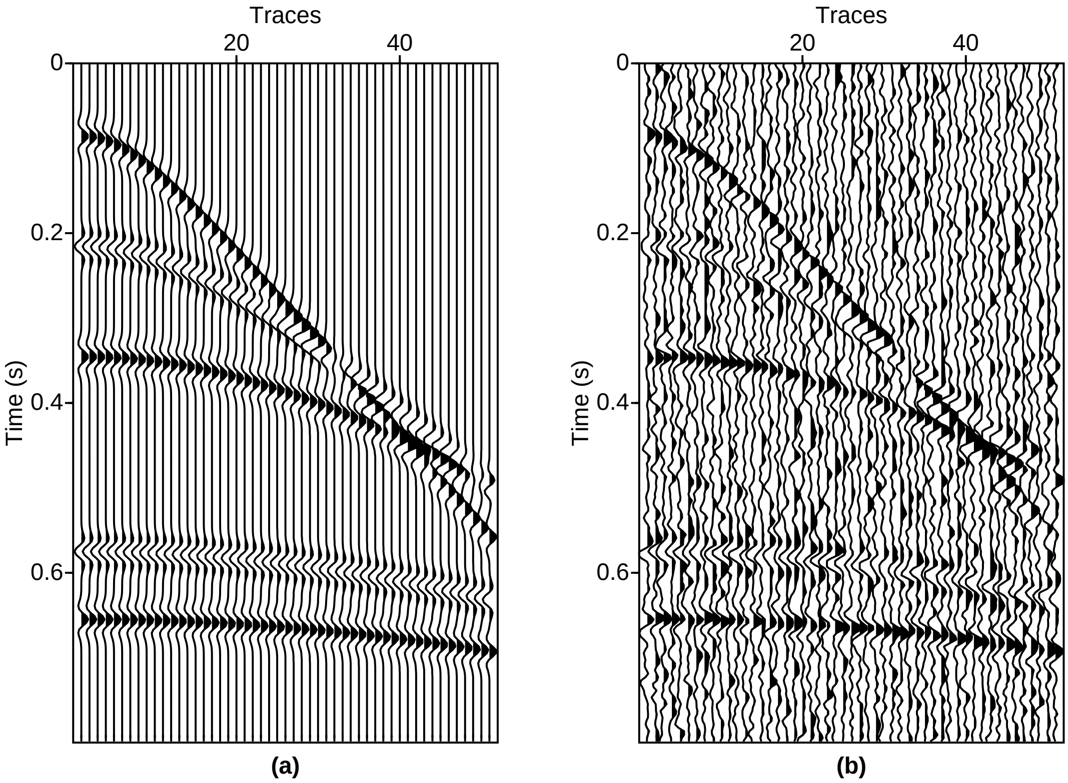

Figure 4.6: Synthetic data example for different filter lengths. (a) Original noise free data. (b) Data contaminated with additive noise ($SNR = 4$).

perform linear and nonlinear predictions in the spatial domain.

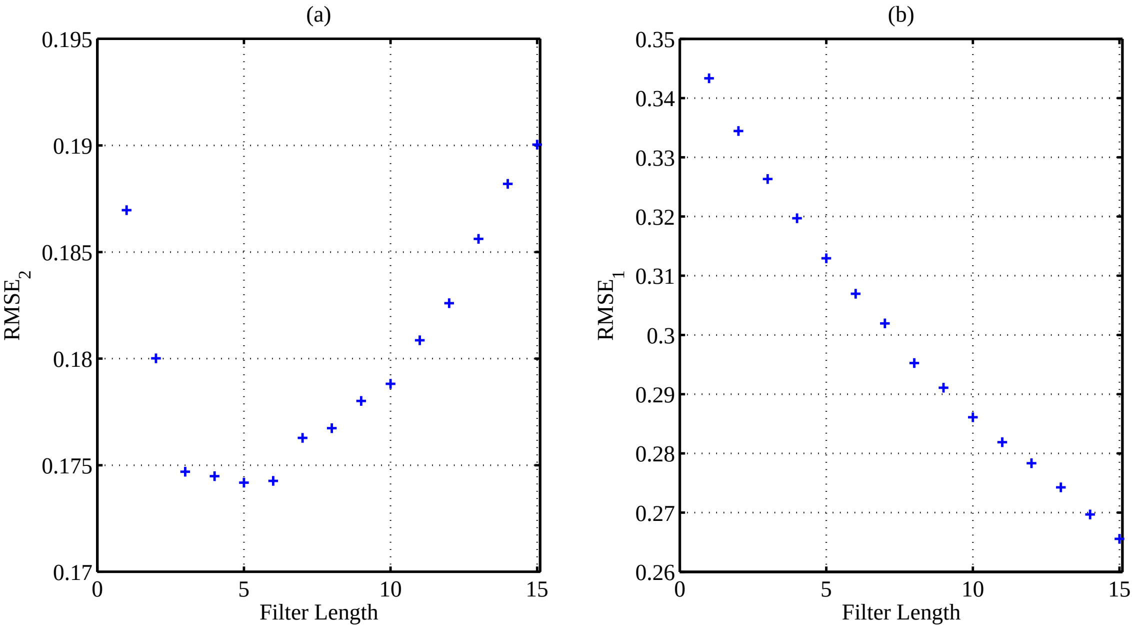

Figure 4.7: Optimality of filter length for the data in Figure 4.6. (a) $RMSE_2$. (b) $RMSE_1$.

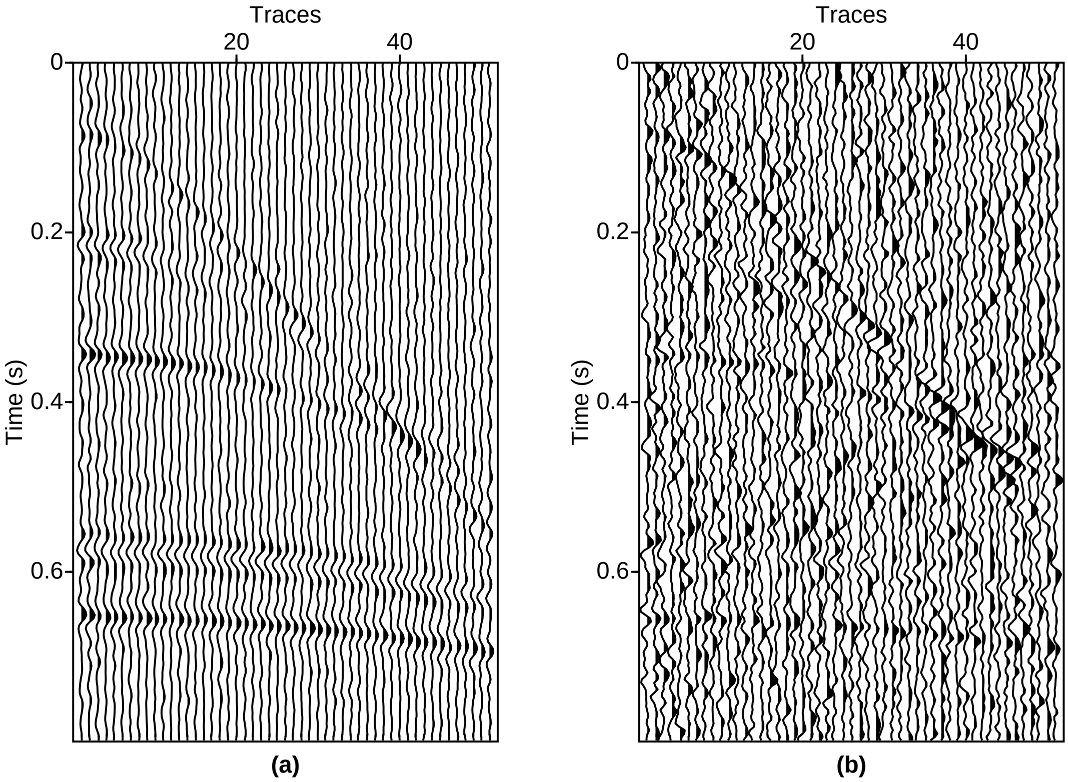

Figure 4.8: (a) Prediction of Figure 4.6(b) ($p = 3$). (b) Error between original data and predicted data.

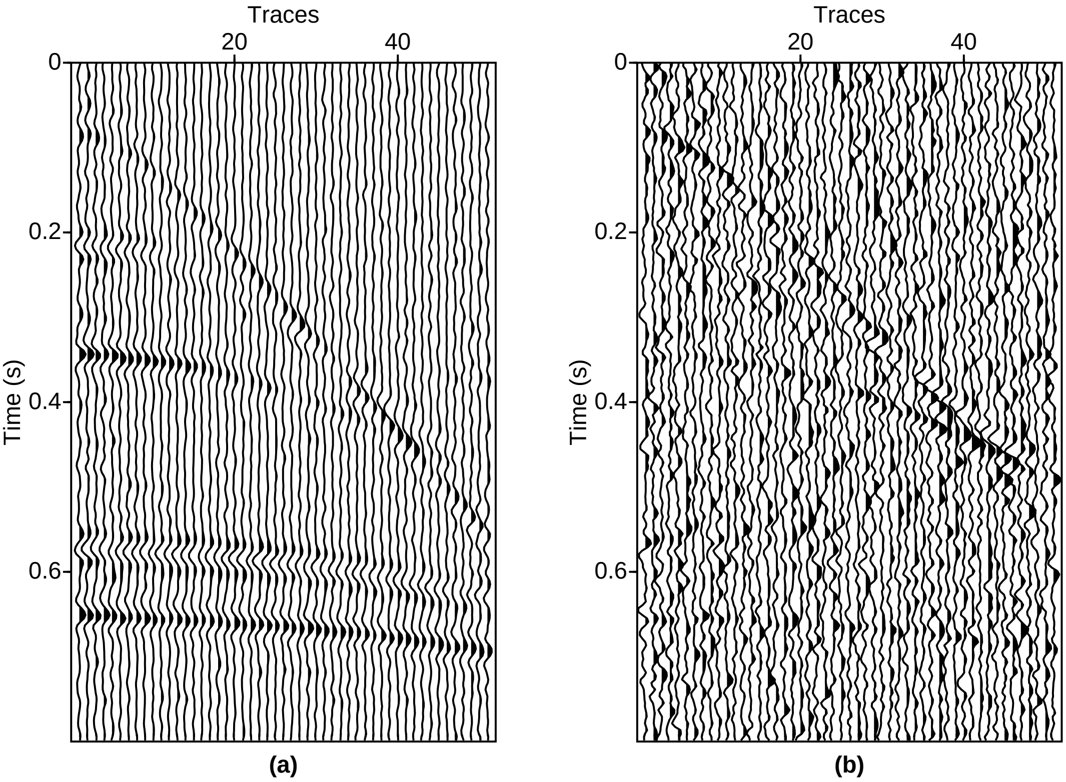

Figure 4.9: (a) Prediction of Figure 4.6(b) ($p = 5$). (b) Error between original data and predicted data.

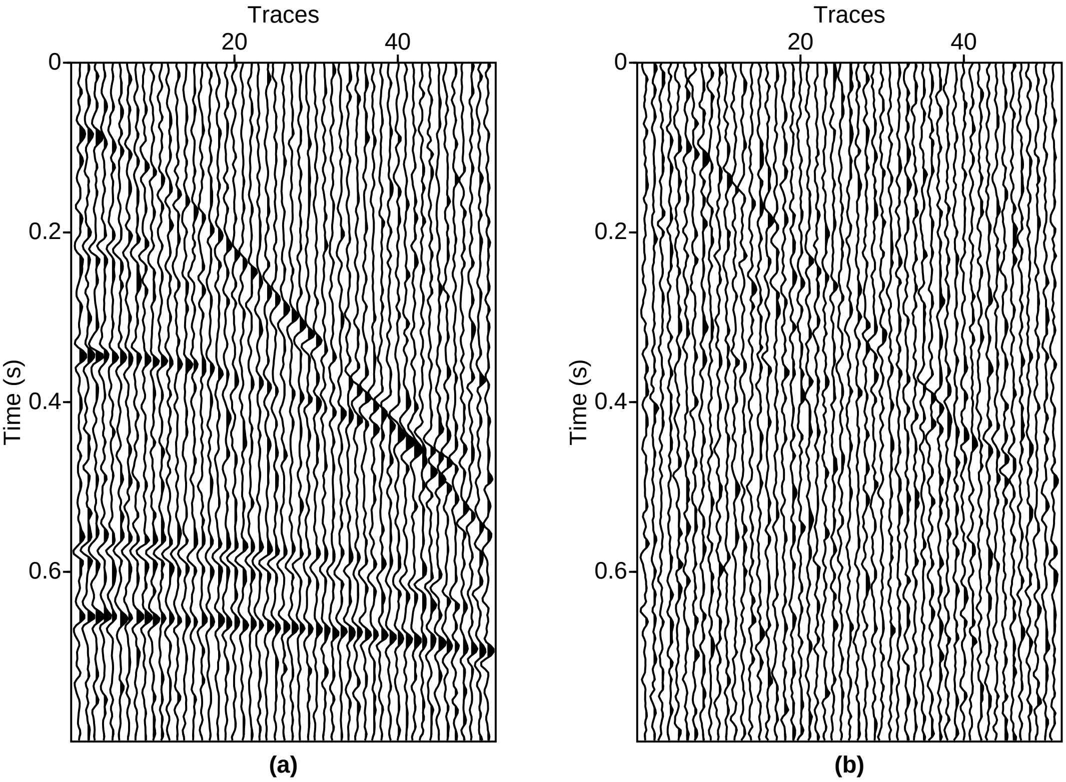

Figure 4.10: (a) Prediction of Figure 4.6(b) ($p = 15$). (b) Error between original data and predicted data.

4.4 Is Noise Removal Possible with a Volterra Series?

I have presented a third- order Volterra model for the prediction of seismic signals. The usual trade-off between noise reduction and signal preservation is controlled by the length of the linear and the nonlinear prediction operators (p, q, and r). It is important to stress that waveforms that exhibit linear moveout can be predicted with a linear system. Conversely, when waveforms exhibit nonlinear moveout, which translates in the $f-x$ domain as chirps, the nonlinear part of the Volterra system helps in modeling the signal.

When I started this project, the idea was to design a new noise reduction system. The premise was to use nonlinear prediction to reconstruct signals in such a way that the reconstruction misfit could be attributed to noise. The latter is the basis of $f-x$ deconvolution for signal-to-noise ratio enhancement. However, nonlinear filtering requires too many coefficients for the quadratic and cubic parts of the operator, therefore, signal and noise are simultaneously modeled (overfitting) (Figure 4.11). The data in the Figure 4.6(a)) and the prediction of this data via a third-order Volterra series predicts the signal as well as the original data but noise is also fitted with the prediction. A small amount of coherent energy leaks to the noise panel. The third-order Volterra series can perfectly model the data, but for $f-x$ noise attenuation it is not useful as it produces many coefficients and therefore models both signal and noise. The next step is to study how one can introduce regularization (smoothing) to the estimation of Volterra kernels and avoid fitting the noise.

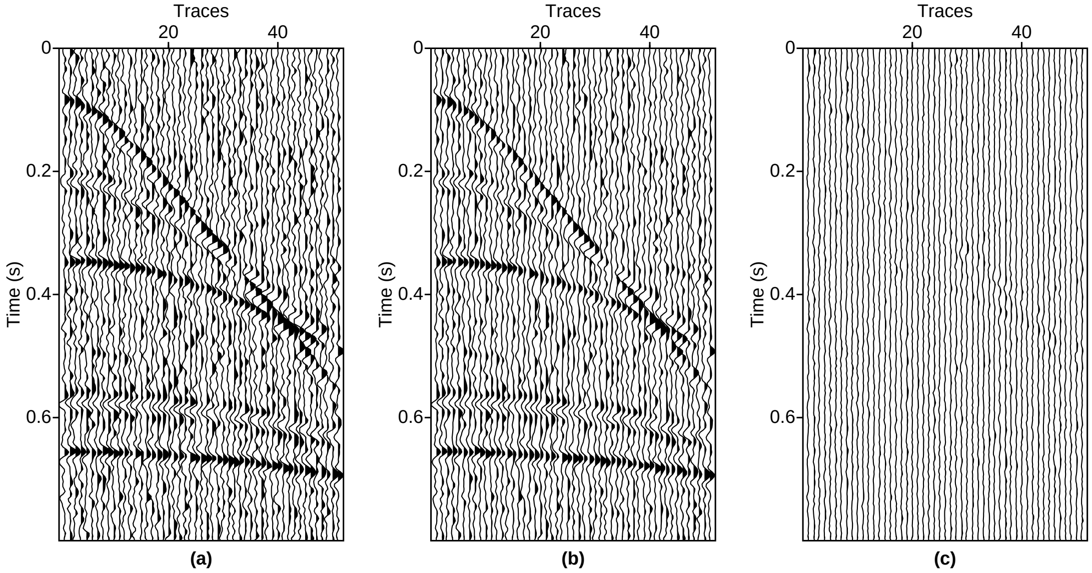

Figure 4.11: Same as the data in Figure 4.6. (a) Original data. (b) Prediction using the third order Volterra series with parameters $p = 5$, $q = 5$, and $r = 5$. (c) Error between original data and predicted data.

4.5 Synthetic and Real Data Examples

I examined the performance of the Volterra expansion with 2-D synthetic and real data examples consisting of linear and hyperbolic events. The examples are used to show that curved events can be predicted using nonlinear filtering techniques. The problem of random noise attenuation is not considered and I use these examples solely to validate our discussion about nonlinear predictions in the $f - x$ domain. Therefore, all examples presented in this section are chosen from noise free data ($\mathbf{D}_c$).

In Figures 4.12(a), 4.13(a), and 4.14(a) I portray synthetic data that consists of two linear events. I use these data to compare predictions from the Volterra series with linear prediction theory. Figures 4.12(b) and 4.14(b) correspond to the prediction of data using a third-order Volterra series with parameters $p = 3$, $q = 3$, and $r = 3$. Figure 4.12(c) shows the error between the observed data and predicted data via a third-order Volterra series. Figures 4.12(d) and 4.13(b) depict predictions of the data using linear prediction theory with parameter $p = 3$. In Figures 4.12(e) and 4.13(c) I portray the difference between original data and predicted data using the linear prediction method. Figure 4.13(d) illustrates the prediction of data using the cubic part of a Volterra series with parameter $r = 3$. Figure 4.13(e) shows the error between original data and predicted data using only the cubic part of a Volterra series. The cubic part of a Volterra series yields a prediction similar to the linear prediction theory. In Figure 4.14 I portray the individual contributions of a Volterra series to the prediction of data. The contribution due to the linear terms of the Volterra series to the predicted data in Figure 4.14(b) is shown in Figure 4.14(c). It can be seen that the data are predicted mostly with linear terms. Figures 4.14(d) and 4.14(e) show the parts of the prediction associated with the quadratic ($q = 3$) and cubic ($r = 3$) terms in a third-order Volterra series. The contribution of the quadratic part is negligible to the prediction of the data; the contribution of the cubic part is relatively better than the quadratic part. Figure 4.14(f) shows the contribution of nonlinear terms associated with quadratic and cubic terms ($q = 3$ and $r = 3$). These examples confirm that waveforms with linear moveouts can be predicted using linear prediction theory and cubic Volterra prediction.

Figures 4.15(a) and 4.16(a) show a 2-D synthetic data example with hyperbolic events. These events have been synthesized using a forward apex-shifted hyperbolic Radon transform (Hargreaves et al., 2003; Trad, 2003). In Figure 4.15 a comparison between linear prediction theory and the third-order Volterra series is illustrated.

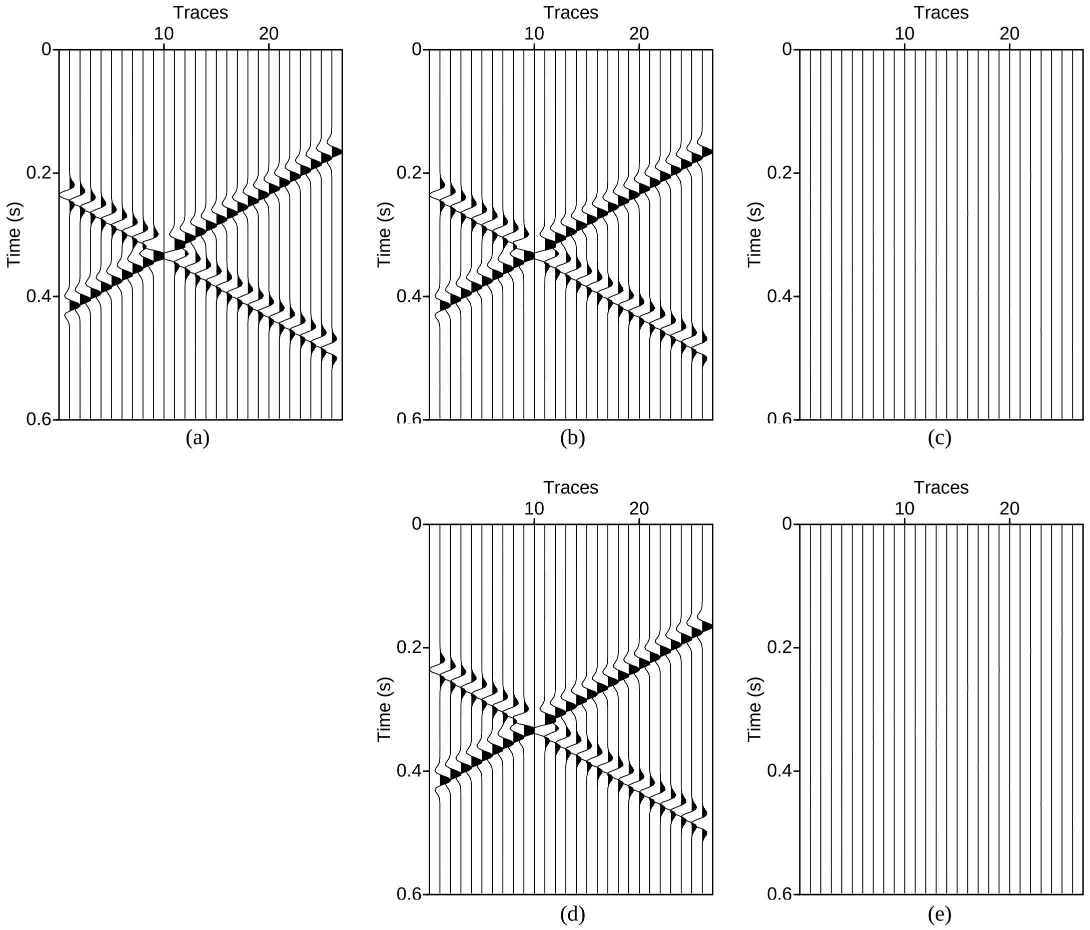

Figure 4.12: 2-D synthetic data for comparison of prediction between linear prediction theory and third order Volterra series. (a) Original data. (b) Prediction using the third-order Volterra series with parameters $p = 3$, $q = 3$, and $r = 3$. (c) Error between original data and predicted via the third-order Volterra series. (d) Prediction using linear prediction theory with parameter $p = 3$. (e) Error between original data and predicted data via linear prediction theory.

In Figure 4.16 I portray the predictions associated with linear, quadratic, and cubic terms.

Figures 4.15(b) and 4.15(d) show data prediction using a third-order Volterra series with parameters $p = 6$, $q = 6$, and $r = 6$ and using linear prediction theory

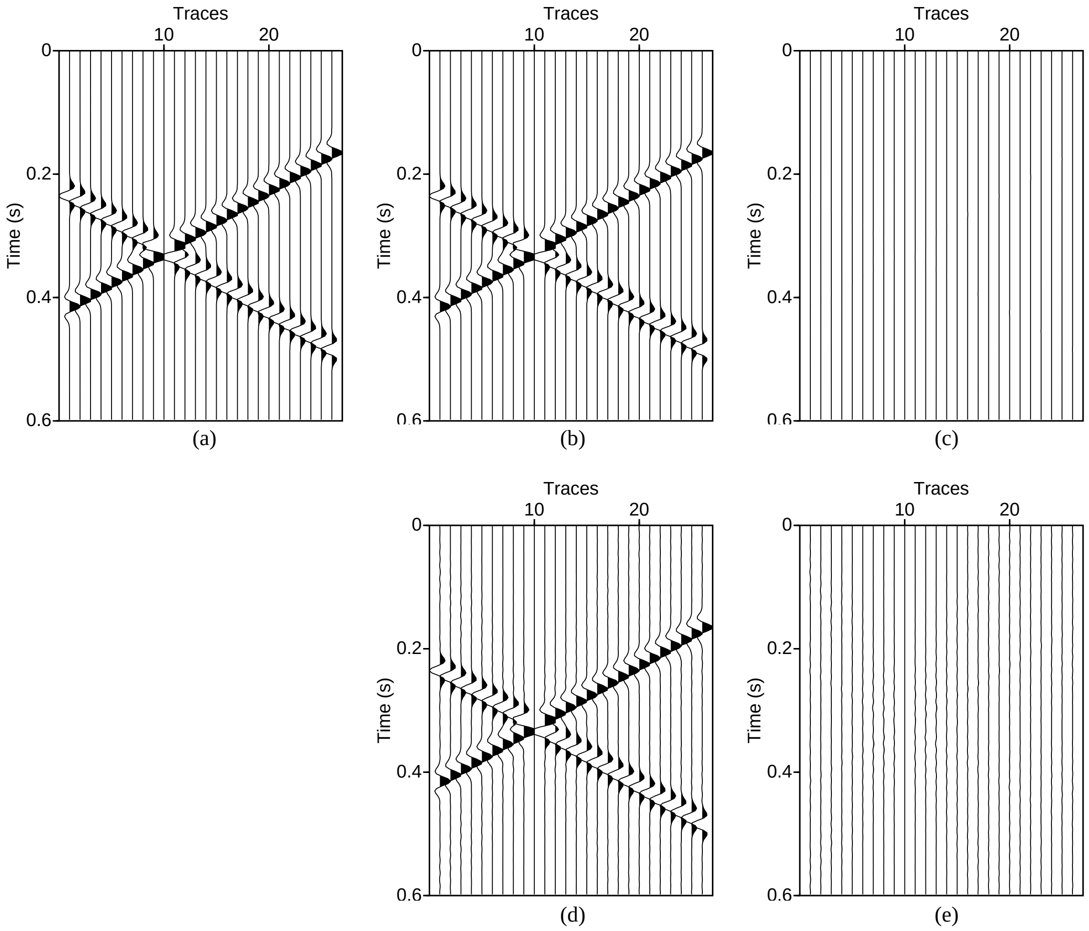

Figure 4.13: 2-D synthetic data for comparison of prediction between linear prediction theory and the cubic part of a Volterra series. (a) Original data. (b) Prediction using linear prediction theory with parameter $p = 3$. (c) Error between original data and predicted data via linear prediction theory. (d) Prediction using the cubic part of a Volterra series with parameter $r = 3$. (e) Error between original data and predicted data via the cubic part of a Volterra series.

with parameter $p = 6$, respectively. The prediction is good and the data are properly modeled but there is a small amount of coherent energy in the noise panel (Figure 4.15(c)). The prediction of the linear prediction method is very poor, the signals are leaking into the error panel (Figure 4.15(e)).

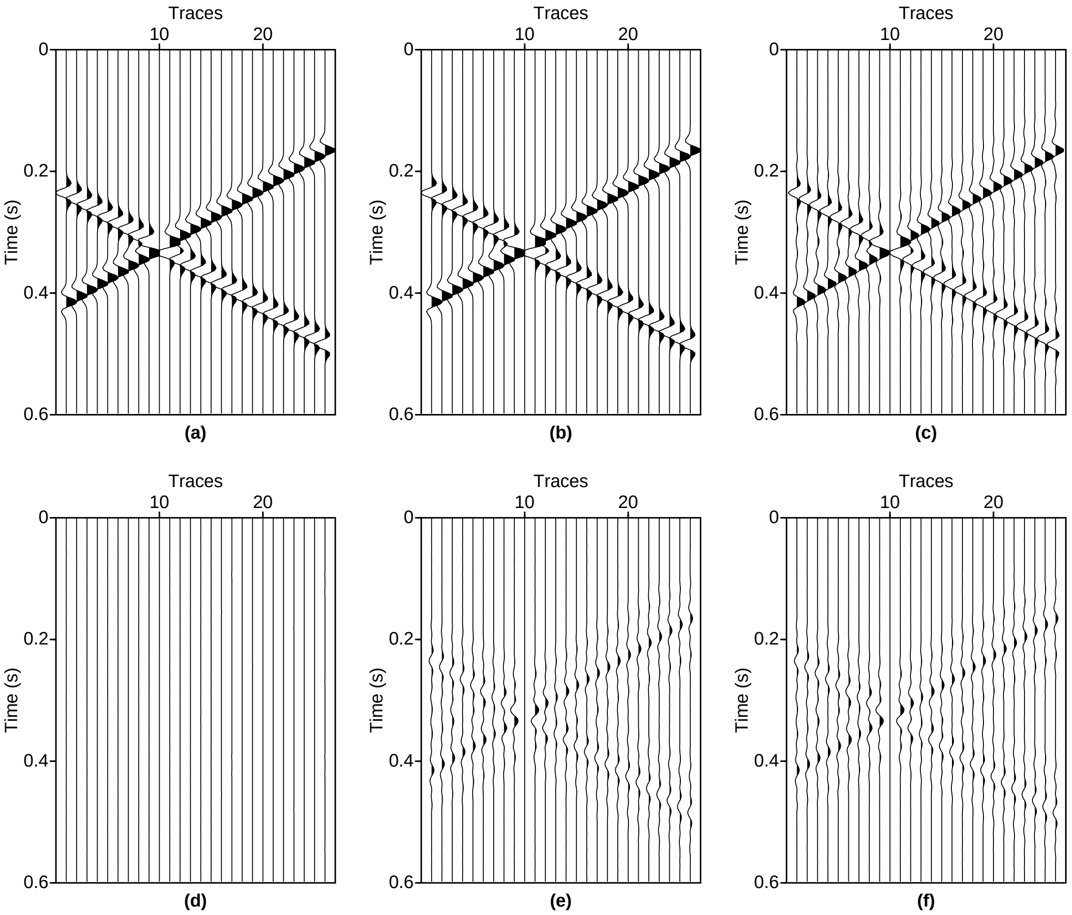

Figure 4.14: 2-D synthetic data. (a) Original data. (b) Prediction using a third-order Volterra series with parameters $p = 3$, $q = 3$, and $r = 3$. (c) Contribution from the linear part. (d) Contribution from the quadratic part. (e) Contribution from the cubic part. (f) Contribution from both quadratic and cubic parts ($q = 3$, and $r = 3$).

In Figure 4.16(c) I portray the contribution due to the linear terms of a third-order Volterra series to predicted data in Figure 4.16(b) due to the linear terms of third order Volterra series. Note that this contribution is negligible. Figures 4.16(d) and 4.16(e) illustrate the parts of the prediction associated with quadratic ($q = 6$) and cubic ($r = 6$) terms in the third-order Volterra series. Figure 4.16(f) shows the contribution of nonlinear terms associated with quadratic and cubic terms ($q = 6$ and $r = 6$). It is clear that nonlinear quadratic and cubic terms are required to properly model the full aperture. . In summary, I are able to model the data by adding quadratic and cubic terms to improve the predictability of the model.

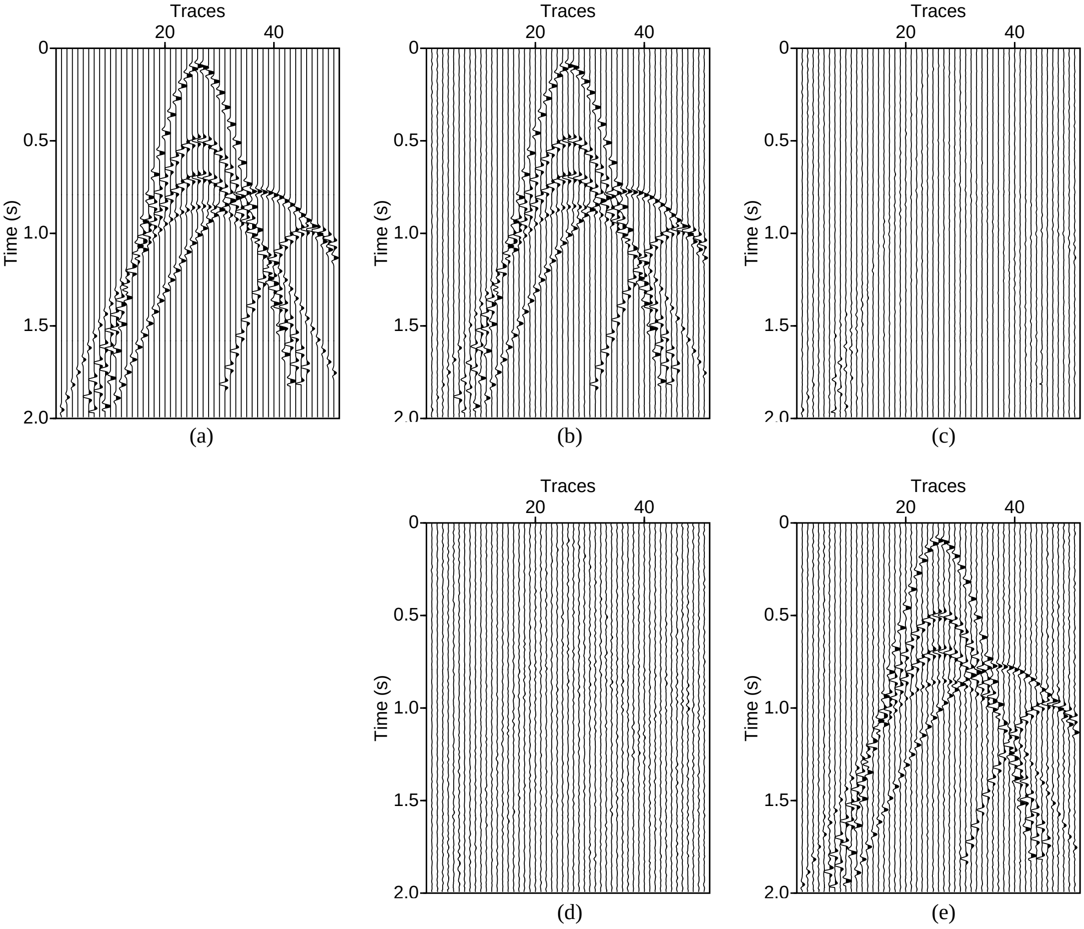

Figure 4.15: 2-D synthetic data for comparison of prediction between linear prediction theory and a third-order Volterra series. (a) Original data. (b) Prediction using a third order Volterra series with parameters $p = 6$, $q = 6$, and $r = 6$. (c) Error between original data and predicted data via a third-order Volterra series. (d) Prediction using linear prediction theory with parameter $p = 6$. (e) Error between original data and predicted data via linear prediction theory.

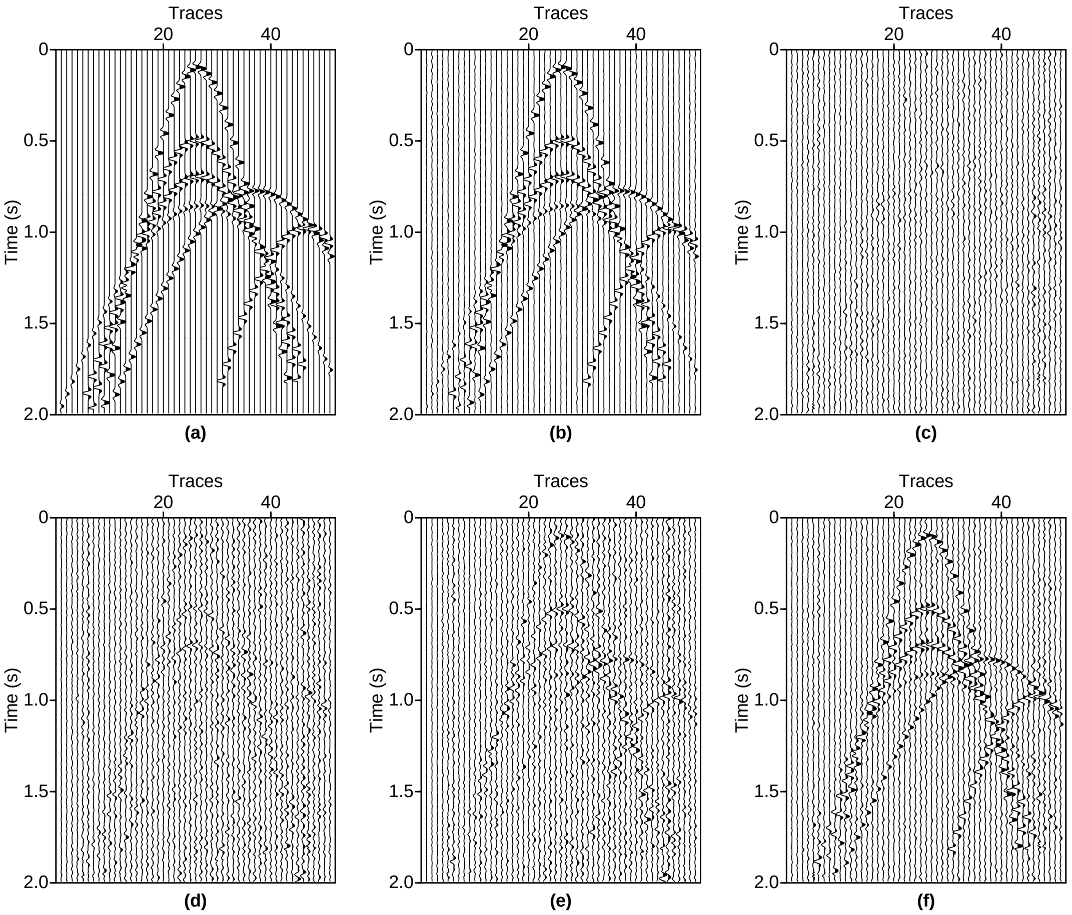

Figure 4.16: 2-D synthetic data. (a) Original data. (b) Prediction using a third-order Volterra series with parameters $p = 6$, $q = 6$, and $r = 6$. (c) Contribution from the linear part. (d) Contribution from the quadratic part. (e) Contribution from the cubic part. (f) Contribution from both quadratic and cubic parts ($q = 6$, and $r = 6$).

Figure 4.17(a) shows a 2-D synthetic example with hyperbolic events and a linear event. These events have been synthesized using a forward apex-shifted hyperbolic Radon transform (Trad, 2003).

I present this example to demonstrate that linear moveouts can be mostly predicted with linear terms in the Volterra series and nonlinear moveouts can be mostly predicted with quadratic and cubic nonlinear terms. Figure 4.17(b) is the prediction using a third-order Volterra series with $p = 7$, $q = 7$, and $r = 7$. In Figure 4.17(c) I portray the part of the prediction attributed to the linear kernel ($p = 7$). Figures 4.17(d) and 4.17(e) show the parts of the prediction associated with quadratic ($q = 7$) and cubic ($r = 7$) terms in the third-order Volterra series. In addition, Figure 4.17(f) illustrates the part of the prediction associated to both quadratic and cubic terms ($q = 7$ and $r = 7$). These terms give good predictions, especially for the apexes of events where linear terms cannot predict the data.

Finally, I test the performance of the Volterra series with a marine data set. The data consist of 60 traces extracted from a marine common offset section acquired over a salt body in the Gulf of Mexico (Figures 4.18(a) and 4.19(a)). The real data set has a combination of diffractions, roughly linear events, and hyperbolic events. I used filters with order $p = 9$, $q = 9$, and $r = 9$ for a third-order Volterra prediction (Figures 4.18(b) and 4.19(b)). In this case the data is properly modeled. I also compute the linear prediction filter with parameter $p = 9$ and attempt to model the data (Figure 4.18(d)). The prediction error between the original data and the prediction with a third-order Volterra series is small (Figure 4.18(c)), whereas the difference between the original data and the predicted data via a linear prediction method is large. In particular, the diffractions are leaking to the error panel as a consequence of improper modeling (Figure 4.18(e)). It is clear that the linear prediction was not able to properly model the data.

In Figures 4.19(c), 4.19(d) and 4.19(e) I examine the individual contributions of the linear and nonlinear parts of the Volterra series to the prediction. Linear terms mostly predict linear moveouts with parameter $p = 9$; nonlinear terms model

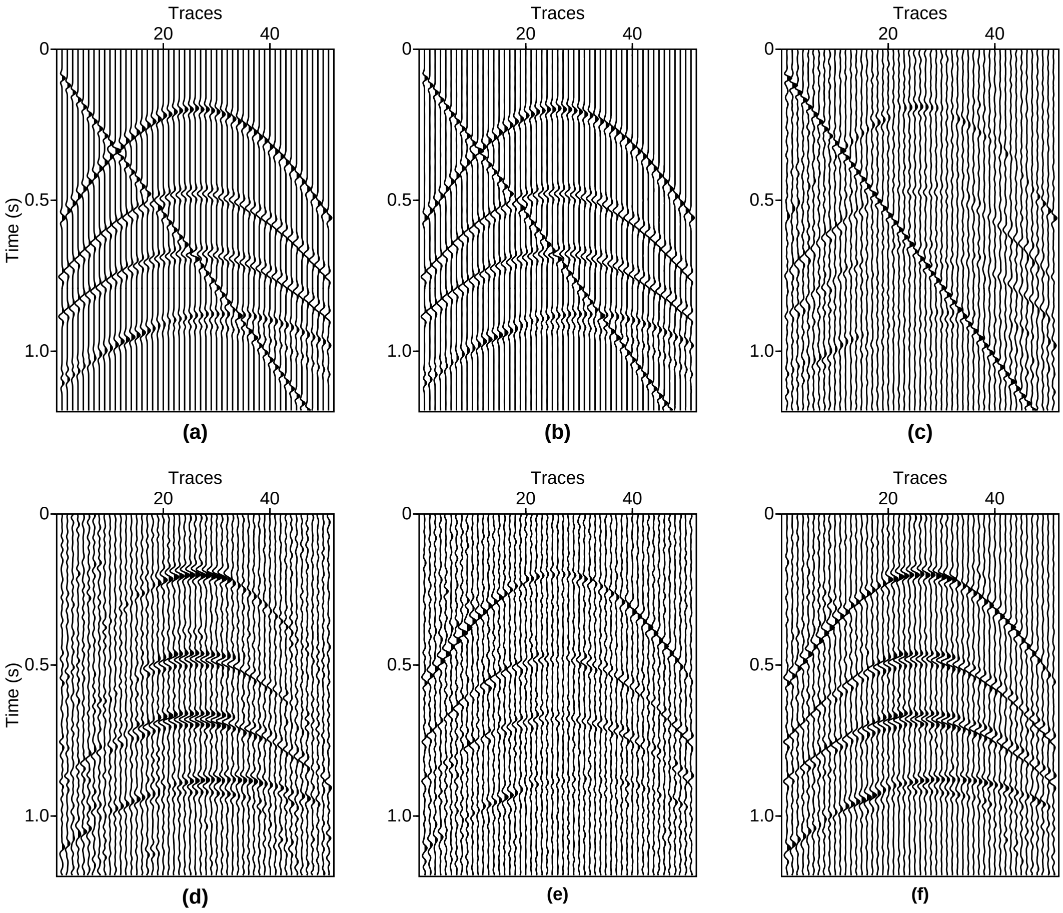

Figure 4.17: 2-D synthetic data. (a) Original data. (b) Prediction using a third-order Volterra series with parameters $p = 7$, $q = 7$, and $r = 7$. (c) Contribution from the linear part. (d) Contribution from the quadratic part. (e) Contribution from the cubic part. (f) Contribution from both quadratic and cubic parts ($q = 7$, and $r = 7$).

diffractions and apexes of events properly with parameters $q = 9$ and $r = 9$ (Figure 4.19(f)). It is clear that this particular data set requires both linear and nonlinear components to properly model the complex waveforms.

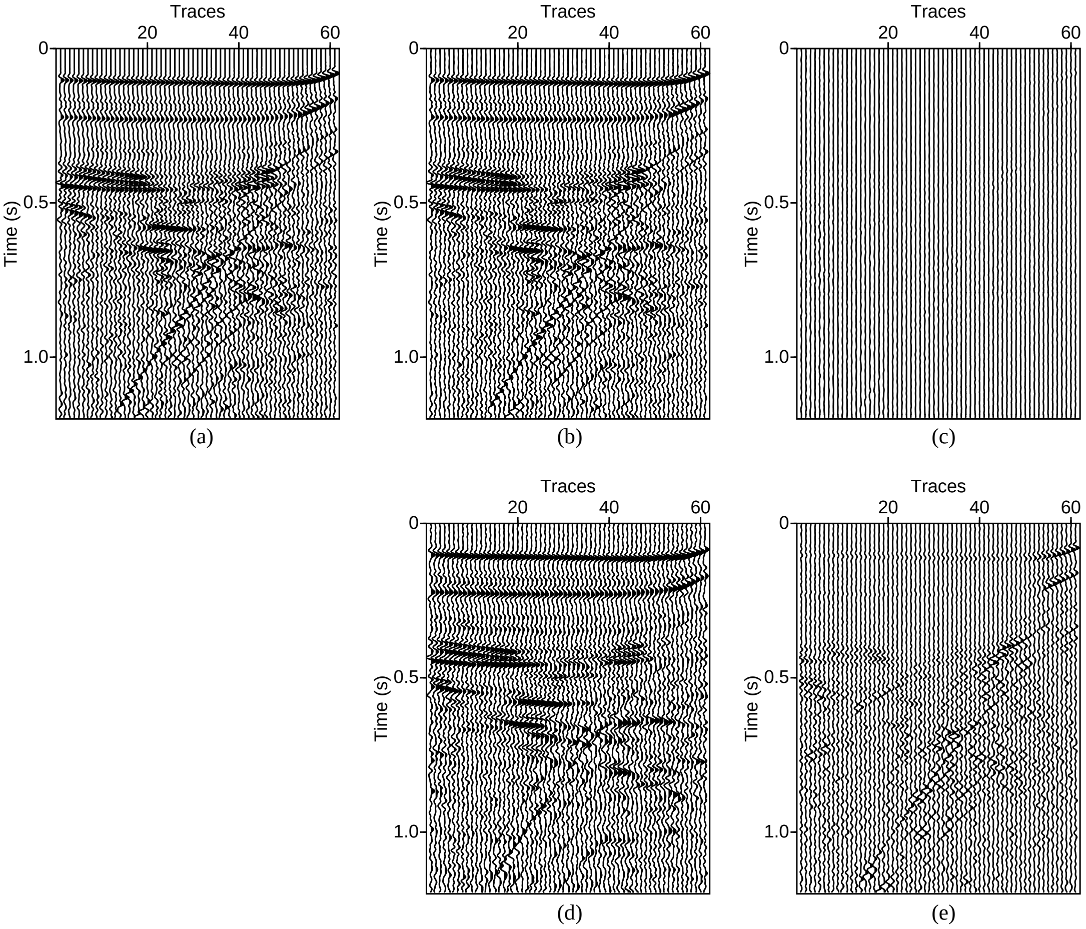

Figure 4.18: 2-D real data for comparison of prediction between linear prediction theory and a third-order Volterra series. (a) Original data. (b) Prediction using a third order Volterra series with parameters $p = 9$, $q = 9$, and $r = 9$. (c) Error between the original data and predicted data via a third-order Volterra series. (d) Prediction using linear prediction theory with parameter $p = 9$. (e) Error between original data and predicted data via linear prediction theory.

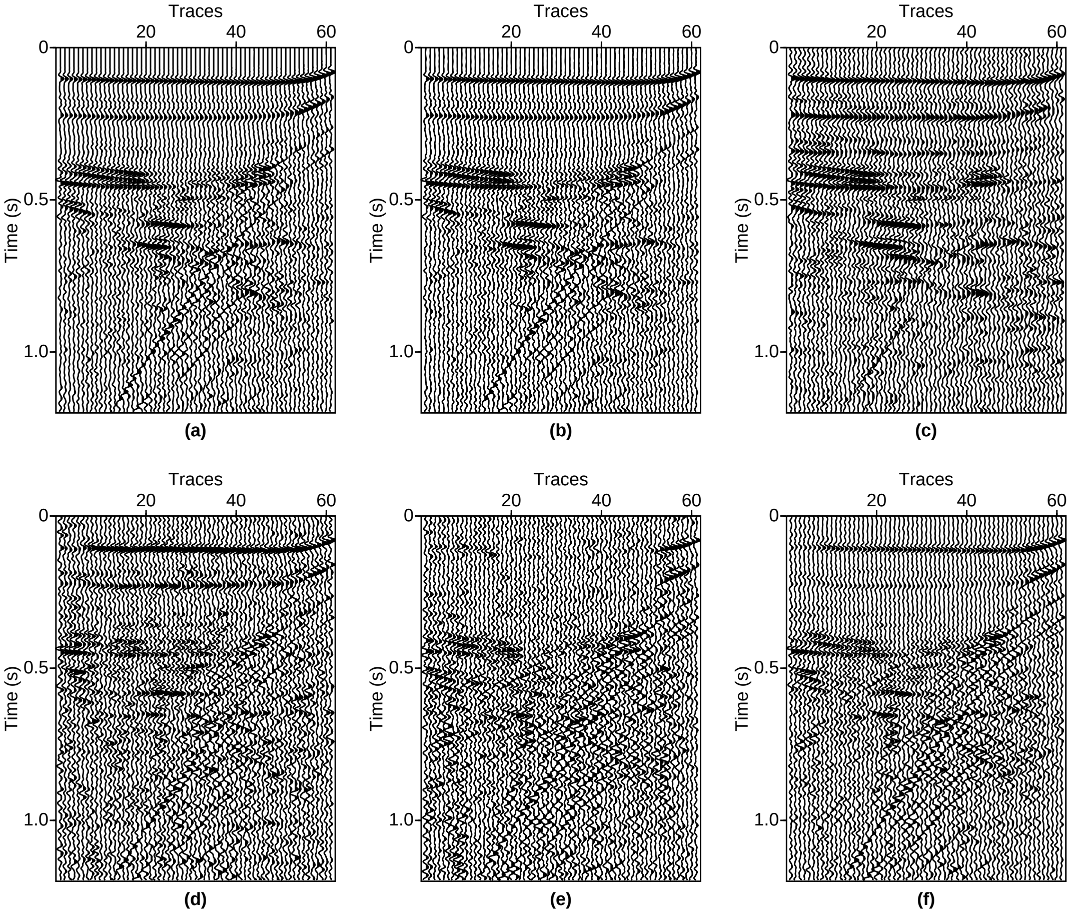

Figure 4.19: 2-D real data. (a) Original data. (b) Prediction using a third-order Volterra series with parameters $p = 9$, $q = 9$, and $r = 9$. (c) Contribution from the linear part. (d) Contribution from the quadratic part. (e) Contribution form the cubic part. (f) Contribution from both quadratic and cubic parts.($q = 9$, and $r = 9$)

4.6 Summary

In this chapter, I surveyed modeling methods in the $f - x$ domain. Canales (1984) method was reviewed and extensions of this method to nonlinear problems were explored.

Events with nonlinear moveouts can be modeled using nonlinear terms of a Volterra series. Events with complex waveforms need additional prediction coefficients in order to properly model the data.

It is clear that linear prediction fails to model data sets with curvature; nonlinear predictions can accurately model these data. I cannot claim, however, that the linear part of a Volterra series models the linear events and that the nonlinear kernels are modeling the hyperbolic events.

Chapter 5

Adaptive Subtraction of Multiples

5.1 Introduction

Noise is an inevitable problem in seismic data processing. All unwanted events that distort the signal are considered noise. I mentioned that random noise could be removed via Canales' method in Chapter 4.

Multiples in seismic data are examples of coherent noise. Multiples can be sorted according to their arrival times (Figure 5.1): short-path multiples that turn back soon after primaries and long- path multiples that turn back as distinct event (Sheriff, 2006).

The reflected data contains both the primaries and the multiples. Energy of primaries have been reflected from source to receiver, while multiples have been reflected two or more times. Also multiples tend to obscure the primaries. The removal of multiples is a complicated problem and partially solved in seismic exploration. There are many methods for elimination of multiples and they are successful when their conditions are fulfilled (Weglein, 1999). Therefore, elimination of multiples remains as a problem in exploration seismology. In this Chapter, I present the adaptive subtraction methods based on Volterra series that can be used to attenuate the multiples.

In the adaptive multiple subtraction problem, one has access to a distorted version of the multiples and the goal is to find an operator that eliminates the distortion before subtracting them from the data. There are many methods to annihilate multiple reflections such as inverse scattering and surface-related multiple attenuation (SRME) (autoconvolution in space) (Weglein et al., 1992; Verschuur et al., 1992; Berkhout and Verschuur, 1997; Spitz, 1999; Abma et al., 2005). I am not going to explain here how a multiple model can be produced; this subject is well understood and published in many studies (Verschuur et al., 1992; Weglein et al., 1992; Berkhout and Verschuur, 1997; Verschuur and Berkhout, 1997; Verschuur and Prein, 1999; Weglein, 1999). These methods are used to construct multiple models, which are subsequently utilized by adaptive subtraction algorithms.

In this thesis I introduce a method for adaptive subtraction of multiples, using a $f-x$ linear prediction filter. A nonlinear prediction filter based on a Volterra series for the removal of multiples from recorded data sets. The problem can also be tackled using linear prediction error filters in the $f-x$ domain as suggested by Spitz (1999). The idea is to compute a $f-x$ prediction error operator from the estimate of the multiples. Then, the estimated prediction error filter is applied to the data containing both primaries and multiples to annihilate the multiples. The procedure is equivalent to finding a notch filter with the notch positioned at the wave number representing the multiple. The procedure of Spitz (1999) can fail when the multiple events in the window of analysis cannot be modeled as linear

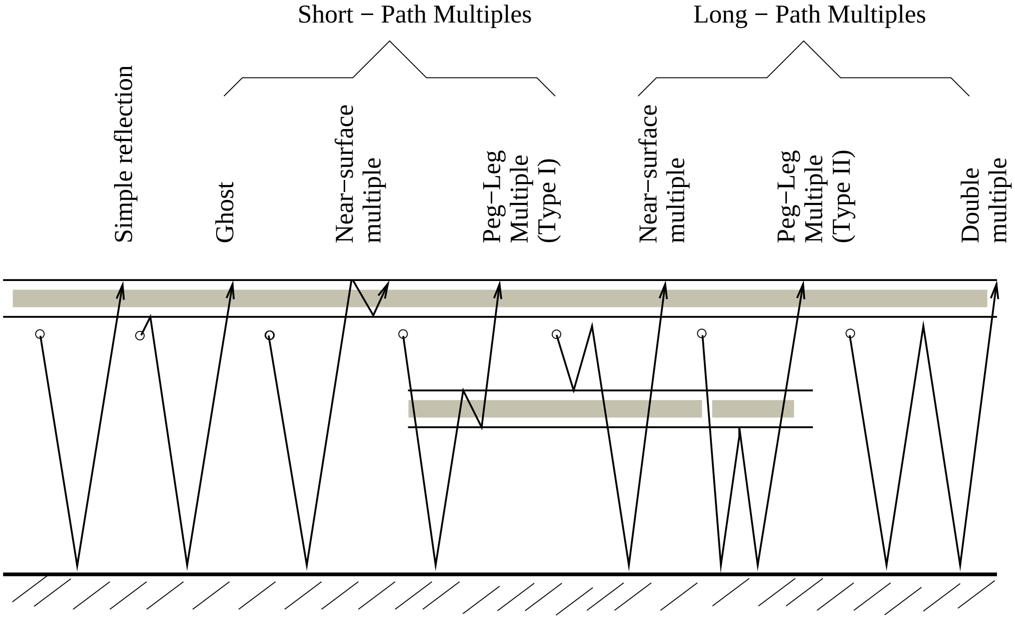

Figure 5.1: Multiple types. After Sheriff (2006) .

events. In this case, the assumption of predictability of waveforms in the $f - x$ domain is not valid and consequently the algorithm fails to attenuate the multiples (Abma et al., 2005). Our extension of Spitz (1999) method to use the Volterra series aims to solve this problem. I assume that the predicted multiples differ from the true multiples mainly in the wavelet and a possible time shift with respect to the multiples in the data panel.

5.2 Prediction Error Operator

From the multiple panel one can compute prediction error filters of the form

$$\mathbf{f} = \begin{bmatrix} 1 \\ -\mathbf{m} \end{bmatrix} \quad . \tag{6.1}$$

Notice that the new operator corresponds to the filter that predicts the noise or error and not to the signal (equation (3.14)). In addition, equation (6.1) is valid for both linear prediction and nonlinear prediction. The operator **f** is then applied to the data (primaries plus multiples). The concept is illustrated with examples.

5.3 Synthetic Data Examples

Figure 5.2(a) represents three linear events with one primary (horizontal event) and two multiples (dipping events). The data are similar to the examples provided in Abma et al. (2005). Figure 5.2(b) is an estimation of the multiples and Figure 5.2(c) is the primary estimation via the $f-x$ linear prediction filter (Spitz, 1999). The removal of multiples and the preservation of the primary reflection is fairly good.

In Figure 5.3(a) I portray one primary (horizontal event) and two multiples that cannot be approximated by linear events. Figure 5.3(b) shows the distorted estimation of multiples. Figures 5.3(c) and 5.3(d) are the solution via the classical linear prediction and the solution with the third- order Volterra series, respectively. Linear prediction algorithm can not removed any multiples, the nonlinear prediction error filter, on the other hand, was able to model the curved multiples and attenuate multiples. Both methods were able to preserve the primary event without affecting its amplitude response.

In Figure 5.4(a) I portray a similar synthetic data example with one primary and two multiples. Figure 5.4(d) shows the distorted estimation of multiples. Figures 5.4(b) and 5.4(c) are the solution via classical linear prediction and the solution with the third- order Volterra series, respectively. Nonlinear prediction error filter subtracts more multiples than linear one. In this example, I illustrate the removed multiples from linear and nonlinear prediction error filters in Figures 5.4(e) and 5.4(f), respectively. Linear prediction error filter eliminated less multiples than the nonlinear prediction error filter. However, the nonlinear prediction filter removes more multiples, but it also affects the amplitude response of the primaries.

In Figure 5.5(a) I present three primaries (linear events) and two multiples (two multiples with sharp apexes) that cannot be properly modeled by linear events. Figure 5.5(b) shows the distorted estimation of the multiples and Figure 5.5(c) is the solution via classical linear prediction. Again, the method can not separate multiples from primary and also distorts primaries. Figure 5.5(d) is the solution with the third order Volterra series. However, the prediction error filter was able to model the curved multiples and produce an acceptable result when applied to the data panel.

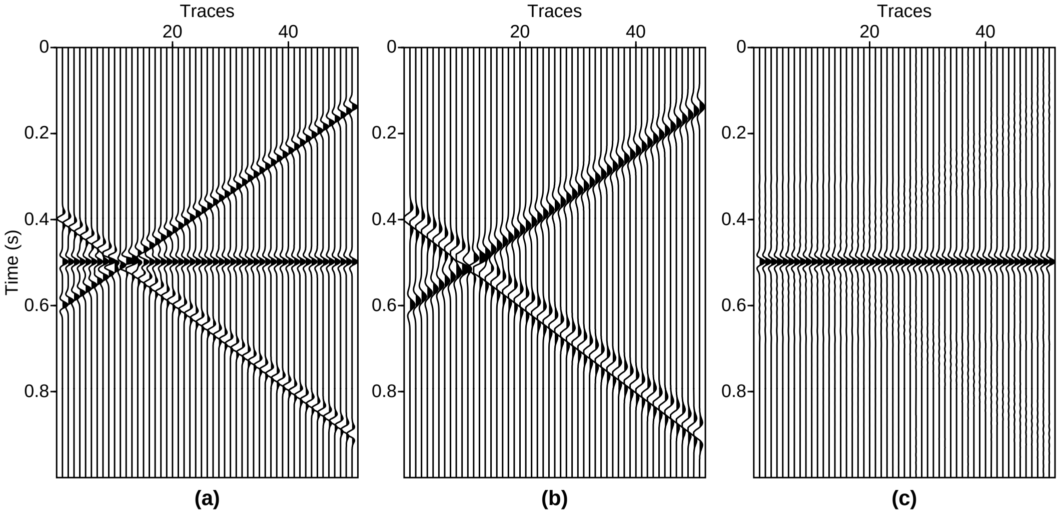

Figure 5.2: Synthetic data example. (a) Original data. (b) Multiples. (c) Adaptive multiple attenuation using $f-x$ linear prediction error operators computed from (b).

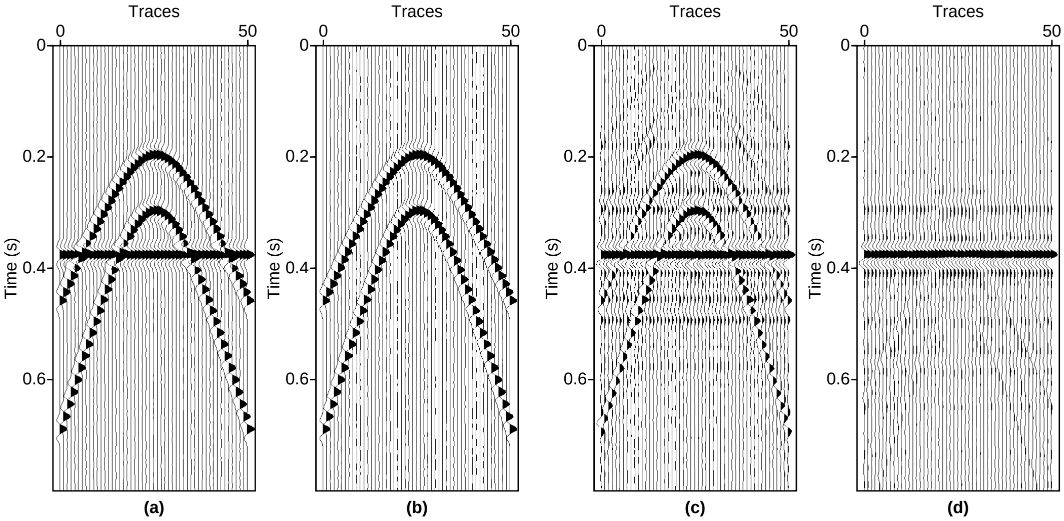

Figure 5.3: Synthetic data example with two multiples and one primary. (a) Original data. (b) Prediction of multiples. (c) Adaptive multiple attenuation via $f-x$ linear prediction error filtering. (d) Adaptive multiple attenuation obtained with a $f-x$ nonlinear prediction error operator (third order Volterra series).

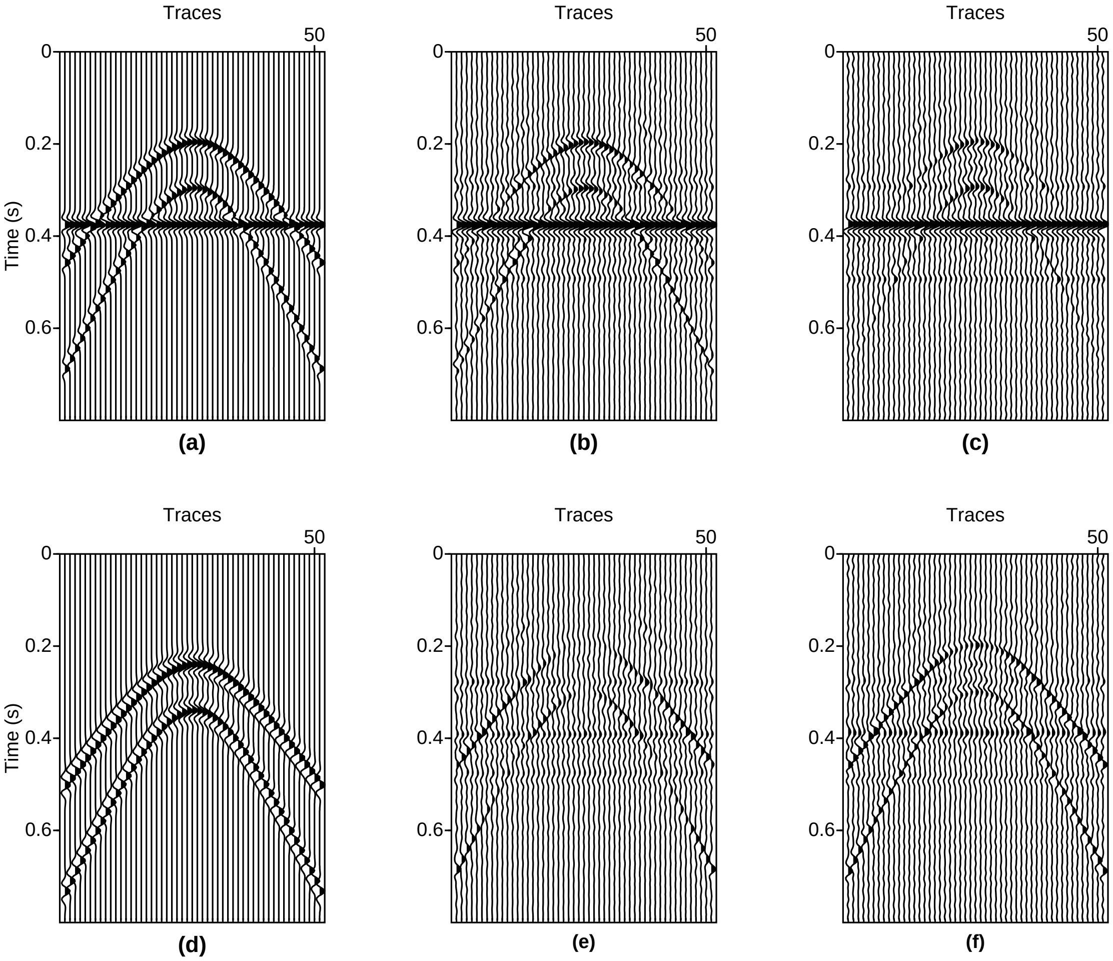

Figure 5.4: Synthetic data example with two multiples and one primary. (a) Original data. (b) Adaptive multiple attenuation via $f{-}x$ linear prediction error filtering. (c) Adaptive multiple attenuation obtained with a $f-x$ nonlinear prediction error operator (third order Volterra series). (d) Prediction of multiples. (e) Removed multiples via $f-x$ linear PEF. (f) Removed multiples with a third- order Volterra PEF.

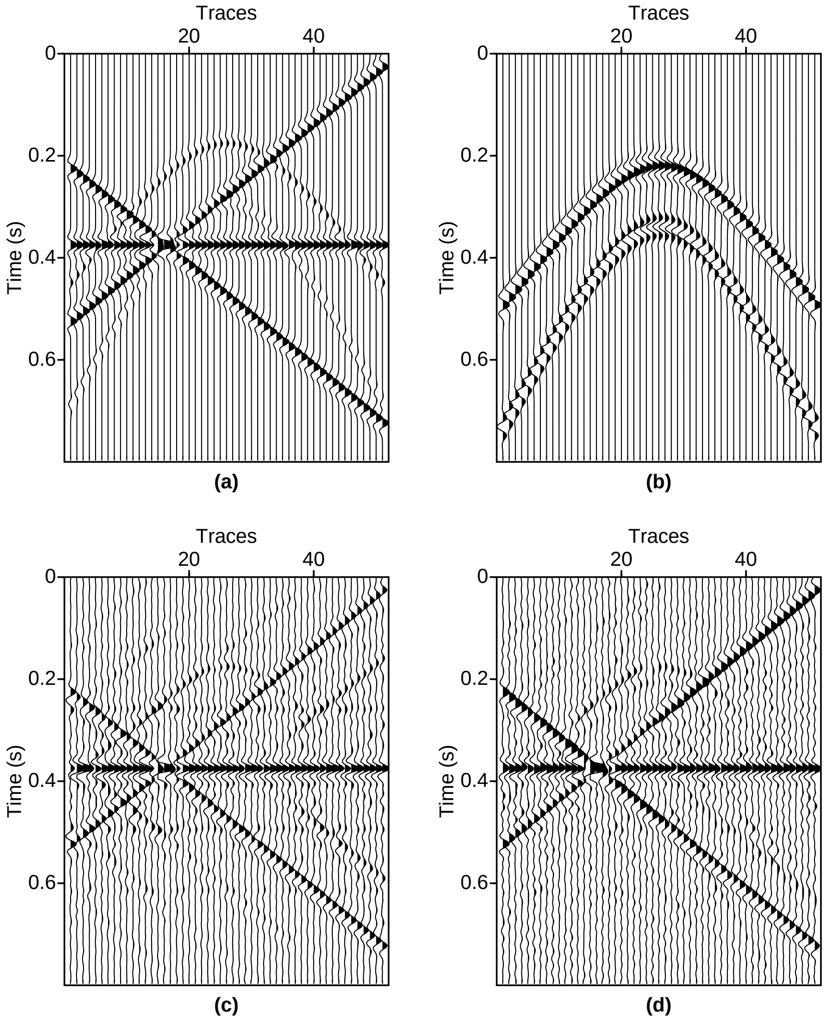

Figure 5.5: Synthetic data example with two multiples and three primaries. (a) Original data. (b) Prediction of multiples. (c) Adaptive multiple attenuation via $f-x$ linear prediction error filtering. (d) Adaptive multiple attenuation obtained with a $f-x$ nonlinear prediction error operator (third order Volterra series).

5.4 Real Data Examples

In this section I will present a real data set from Gulf of Mexico subsalt data set. These data was made publically available to several research groups in order to test methods for multiple attenuation (Verschuur and Prein, 1999). The major problem in this data set is that the primaries are weaker than the multiples. The common offset and shot gather data sets are recorded with 4 ms time sampling interval in 7 seconds; the number of samples per traces are 1751 samples. I extracted 35 traces from a common offset data set and 38 traces for shot gather data set to test our algorithm.

In Figure 5.6(a) I portray a common offset data set that contains multiples and primaries. Figure 5.6(b) shows the estimation of a multiple model via SRME method (Verschuur et al., 1992). Figure 5.6(c) is the solution obtained via classical linear prediction method. Linear prediction filter attenuates all multiples from the original data. Figure 5.6(d) is the solution with a third order Volterra series. Nonlinear prediction error filter provides a similar result to the linear one. Both methods were able to remove the multiples; particularly after 6 seconds in travel time, the primaries can be seen in Figures 5.6(c) and (d).

Figure 5.6(a) portrays the data. I can focus on an analysis interval between 3.5 and 4.5 seconds for a better representation (in Figure 5.7(a)). Figure 5.7(c) shows that the linear prediction error filter works quite well, but it also distorts some events that are not present in the multiple model. Red solid arrow indicates a roughly linear event (Figure 5.7)(c), and blue dashed arrow and green dotted-dashed arrows (Figure 5.7(d)) could be obtained via a nonlinear prediction error filter. These same events are not recoverable via a linear prediction error filter.

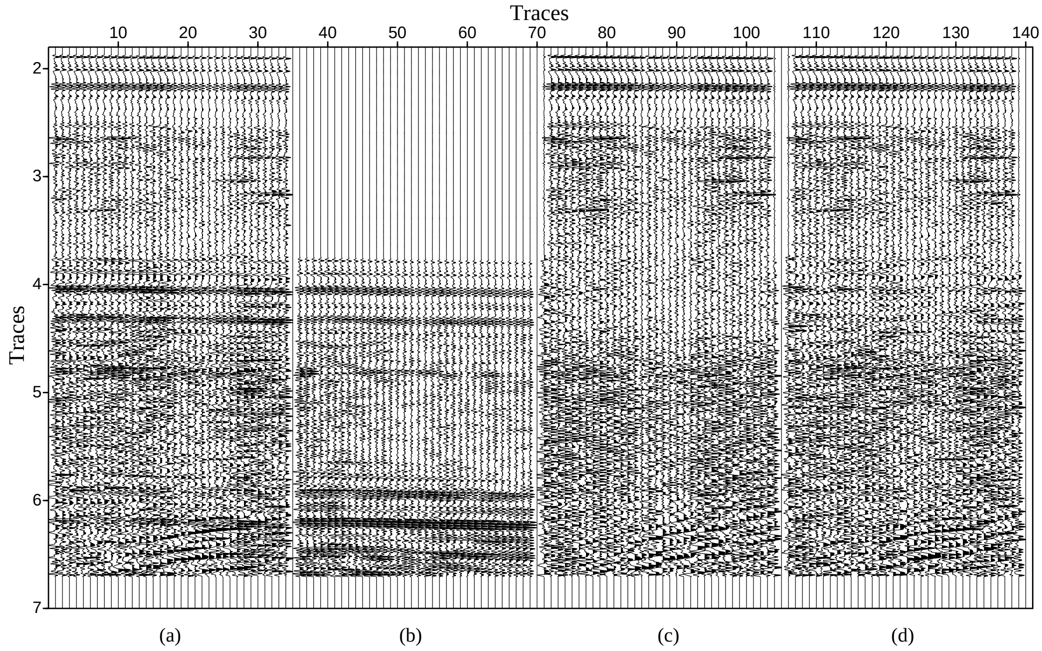

Figure 5.6: Real Data example common shot offset. (a) Original data. (b) Prediction of multiples. (c) Adaptive multiple attenuation via $f - x$ linear prediction error filtering. (d) Adaptive multiple attenuation obtained with a $f - x$ nonlinear prediction error operator (third order Volterra series).

I illustrate Figure 5.8 to compare amount of removed multiples with linear and nonlinear prediction filters. Figures 5.8(c) and 5.8(d) shows that similar amount of multiples has been removed.

Figure 5.9(a) illustrates a shot gather data set that contains slightly curved events. Figure 5.9(b) shows estimation of multiple model reconstructed with FSME. Figure 5.9(c) is the solution via linear prediction error filtering model. Figure 5.9(d) is the solution with a third order Volterra series. Linear and nonlinear prediction error filters works similar and there are more residuals than the common offset data set. A possible primary can be seen after 6 seconds in Figures 5.9(c) and (d).

Shot gather data can be examined by focusing on time interval between 3.75

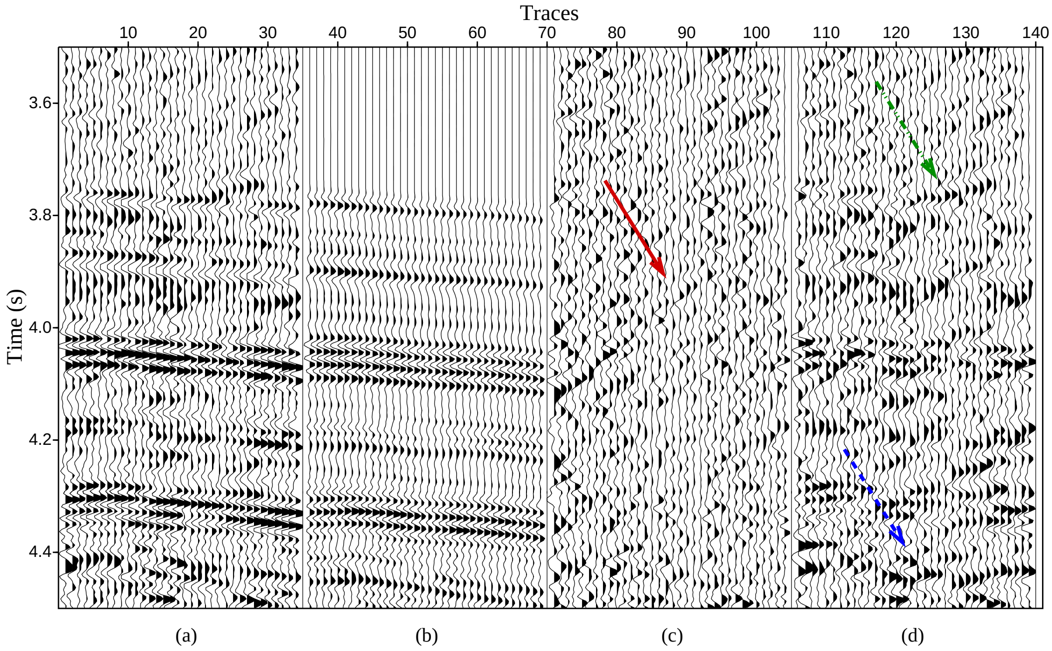

Figure 5.7: Real Data example common offset gather (closer look). (a) Original data. (b) Prediction of multiples. (c) Adaptive multiple attenuation via $f - x$ linear prediction error filtering. (d) Adaptive multiple attenuation obtained with a $f - x$ nonlinear prediction error operator (third order Volterra series).

and 5 seconds (in Figure 5.10(a)). Figures 5.10(c) and (d) provides similar results. I can see removed multiples in Figures 5.11(c) and (d); both methods remove some portion of multiples.

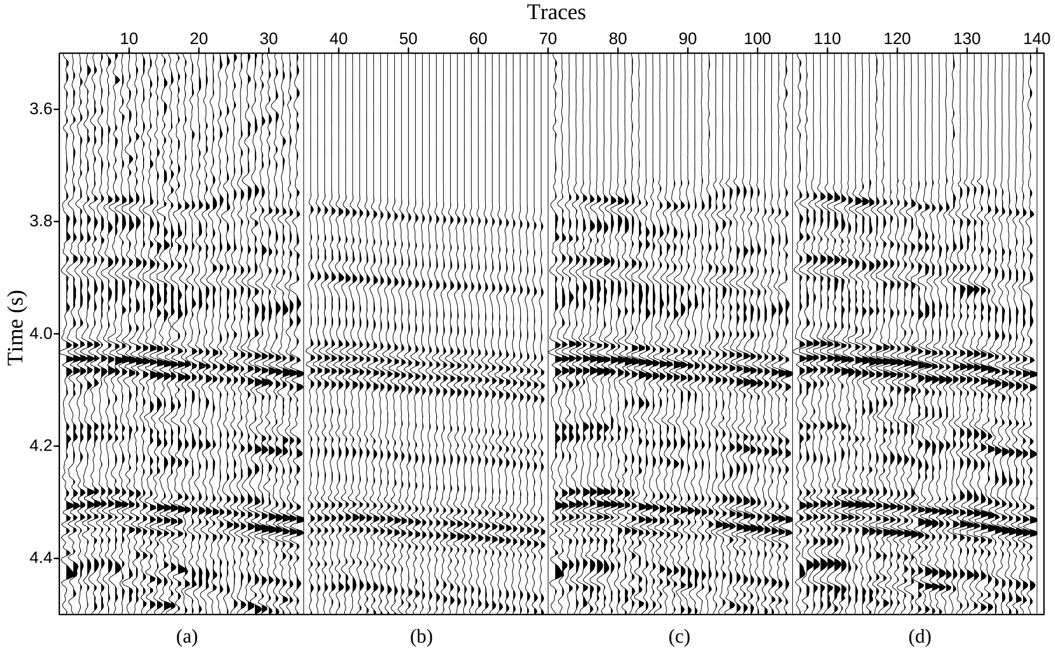

Figure 5.8: Real Data example common shot offset (closer look removed multiples). (a) Original data. (b) Prediction of multiples. (c) Removed multiples via $f - x$ linear prediction error filtering. (d) Removed multiples with a $f - x$ nonlinear prediction error operator (third order Volterra series).

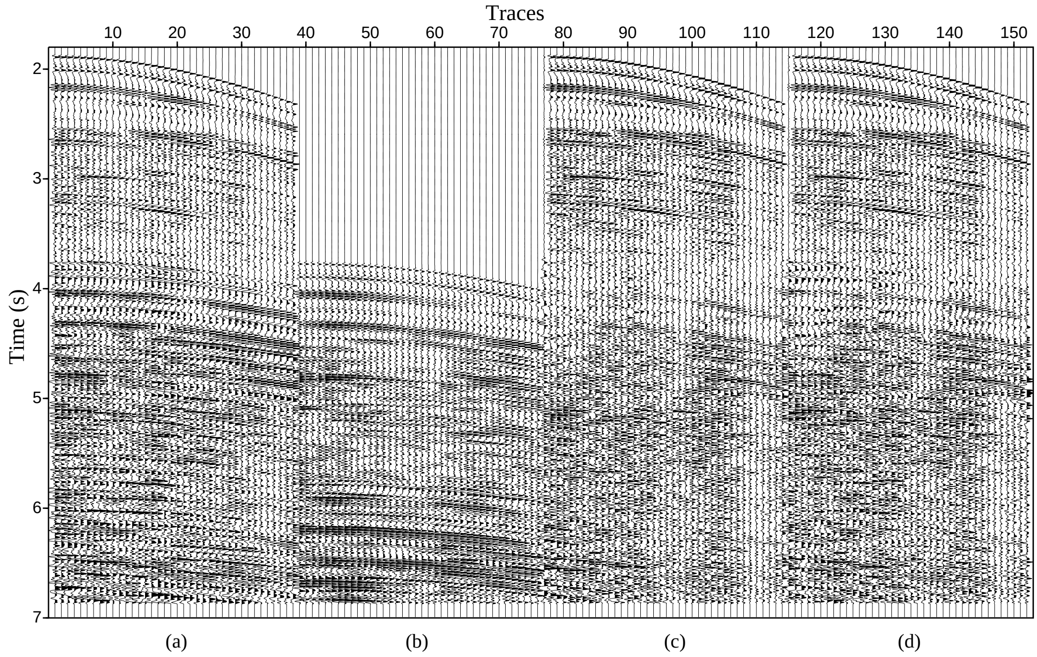

Figure 5.9: Real Data example shot gather. (a) Original data. (b) Prediction of multiples. (c) Adaptive multiple attenuation via $f - x$ linear prediction error filtering. (d) Adaptive multiple attenuation obtained with a $f - x$ nonlinear prediction error operator (third order Volterra series).

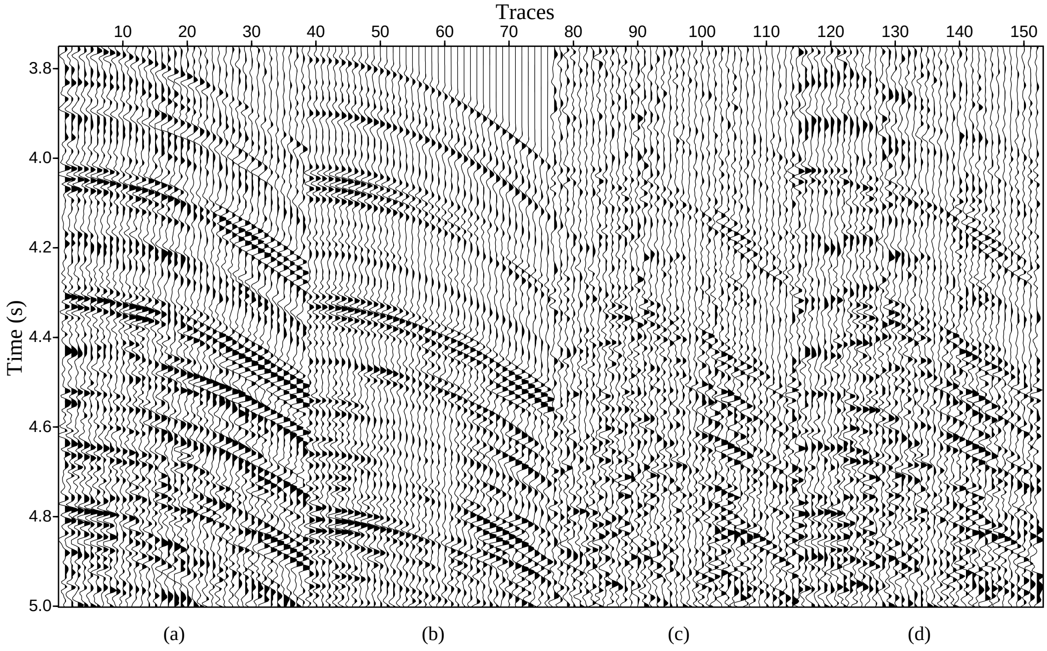

Figure 5.10: Real Data example shot gather (a closer look). (a) Original data. (b) Prediction of multiples. (c) Adaptive multiple attenuation via $f-x$ linear prediction error filtering. (d) Adaptive multiple attenuation obtained with a $f-x$ nonlinear prediction error operator (third order Volterra series).

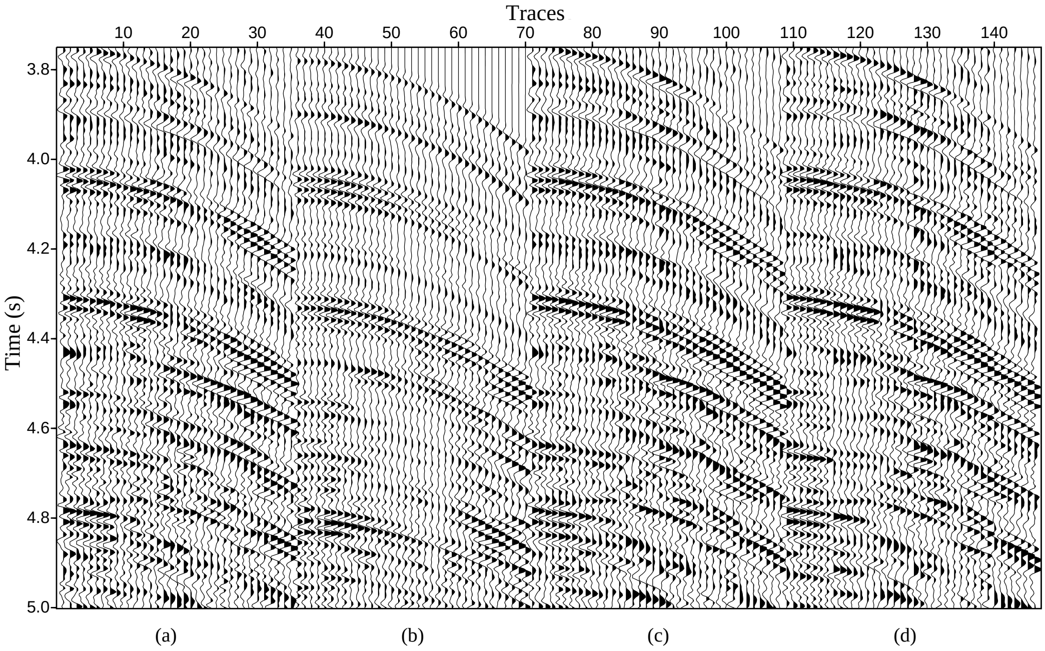

Figure 5.11: Real Data example shot gather (closer look of removed multiples). (a) Original data. (b) Prediction of multiples. (c) Removed multiples via $f - x$ linear prediction error filtering. (d) Removed multiples with a $f - x$ nonlinear prediction error operator (third order Volterra series).

5.5 Summary

In this Chapter I have investigated the possibility of using nonlinear prediction for adaptive attenuation of multiples, so far with encouraging results. Adaptive subtraction of curved events with synthetic data examples has showed that Volterra series contributes removal of curved events. On the other hand, real data examples indicates that this method is not better than usual adaptive subtraction models based on linear prediction error filtering methods when modeling with small data windows in the f - x domain. More research, however, is needed to understand the role played by the nonlinear kernels of the Volterra expansion. In other words, the linear part has a very well understood action when written in prediction error form. It represents a notch filter that attenuates one or more linear events. The second and third-order kernels must contribute as some sort of special "notch" operators that can model a continuous distribution of wave numbers (variable dip in the $t - x$ domain).

Chapter 6

Conclusions and Future Directions

This thesis presented a method of signal modeling based on Volterra series. Specifically, I have developed an autoregressive method based on a nonlinear prediction algorithm. The prediction coefficients of the first-, second-, and third-order Volterra model are obtained with a least squares solution. I also bring a new approach to solving two fundamental problems in seismic exploration: modeling of complex waveforms in the $f-x$ domain and adaptive subtraction of multiples.

Primary issues and available methods in modeling time series are examined in Chapters 1 and 2. First, the nonlinear system is introduced as a higher-order extension of the usual linear convolution method. Former linear prediction methods are also investigated to obtain prediction coefficients. Comparisons between a first-order Volterra series and Yule-Walker equations and Burg's algorithm have shown that linear prediction methods model the data quite similarly. On the contrary, all linear methods can not properly model complex data sets (waveforms) unless a large number of coefficients is used.

In Chapter 3 a Volterra series and its properties are analyzed. The Volterra series in this thesis has been truncated to a third-order Volterra series. Prediction coefficients are found via least squares solutions. The real and synthetic data examples showed that Volterra series modeling modeling is more versatile than linear prediction at the time of representing signals such as those arising in the context of exploration seismology.

Events that exhibit strong curvature cannot be modeled via linear $f-x$ prediction filters. I can model these events by implementing the Volterra algorithm. The optimum filter length of a linear model of order (p) or a nonlinear Volterra model of order (p, q, r) can be computed via inspection of the residuals. A more practical method is desirable but at this stage looking for evidence of under-over fitting using sophisticated statistical analysis method was beyond the scope of this study.

Different synthetic and real data examples showed that the modeling of events with linear moveout can be predicted mainly via linear prediction coefficients and cubic prediction coefficients. The contribution of quadratic parts to events with linear moveouts is negligible. In addition to this, events that exhibit curvature can be modeled properly with nonlinear terms of a Volterra series. Finally, it is important to stress that waveforms that exhibit linear moveout can be predicted with a linear system. Conversely, when waveforms exhibit nonlinear moveout, which translates in the $f-x$ domain as non stationary signals, the nonlinear part of the Volterra system helps in predicting the signal.

Another application of the nonlinear Volterra model in this thesis is adaptive subtraction of multiples. Elimination of multiples is a common problem in exploration seismology. In this case the prediction coefficients have been designed as prediction error filters (PEF). I used this algorithm to annihilate multiples from seismic sections. I have satisfactory results for adaptive subtraction.

Limitations

All methods have limitations. Volterra series produces many coefficients and calculation of these coefficients is computationally expensive. Also, a large number of coefficients causes overfitting of the signal and this leads to modeling both signal and noise.

6.1 Future Directions

Volterra series have been implemented in nonlinear systems in a range of problems from biomedical research to communication applications. Adaptations of these methods to seismic exploration problems are presented in Chapter 4 and Chapter 5 of this thesis. The Volterra filter I developed is time-invariant; the signal is stationary. However, in the future the Volterra filter presented here could be designed as a time-variant filter.

The reconstruction of monochromatic complex sinusoids is a well studied problem in seismic exploration in the case of regularly spaced traces in seismic sections. However, there are various interpolation methods in $f-x$ and $f-k$ domains; these methods can not reconstruct missing traces with events that exhibit curvature. There are examples of Volterra series implementation in the reconstruction of nonlinear image interpolation. For instance, Collis et al. (1997) presented an application of pixel interpolation in television images based on a nonlinear Volterra filter. Particularly, a third-order Volterra filter will be useful to interpolate missing data in seismic traces.

Noise reduction is not possible with the Volterra method because nonlinear filtering requires many coefficients for the quadratic and cubic parts of the operator resulting in signal and noise being simultaneously modeled (overfitting). The next step is to study a new inversion scheme to estimate Volterra kernels and to smooth the data to prevent overfitting the noise.

Bibliography

Abma, R., N. Kabir, K. H. Matson, S. Michell, S. A. Shaw, and B. McLain (2005). Comparisons of adaptive subtraction methods for multiple attenuation. *The Leading Edge 03*, 277–280.

Akaike, H. (1969). Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics 21*(2), 243–247.

Benedetto, S. and E. Biglieri (1983). Nonlinear equalization of digital satellite channels. *IEEE Journal on Selected Areas in Communications SAC-1*(1), 57–62.

Berkhout, A. J. and D. J. Verschuur (1997). Estimation of multiple scattering by iterative inversion, part i: Theoretical considerations. *Geophysics 62*(5), 1586–1595.

Box, G. E. and G. M. Jenkins (1970). *Time Series Analysis Forecasting and Control.* Holden-Day, San Francisco, CA.

Bracalari, M. and E. Salusti (1994). On a nonlinear autoregressive method applied to geophysical signal. *Geophysics 59*(9), 1270–1277.

Burg, J. P. (1975). *Maximum entropy spectral estimation.* Ph. D. thesis, Dept. of Geophysics, Standford Univ., Standford, CA.

Canales, L. L. (1984). Random noise reduction. *Expanded Abstract, 54th Annual SEG Meeting, Atlanta*, 525–527.

Chatfield, C. (1989). *The analysis of time series an introduction.* Chapman and Hall New York.

Cherry, J. A. (1994). *Distortion Analysis of Weakly Nonlinear Filters Using Volterra Series.* M.Eng. thesis, Carleton Univ., Ottawa.

Coker, M. J. and D. N. Simkins (1980). A nonlinear adaptive noise canceller. *IEEE Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '80 5*, 470–473.

Collis, W. B., M. Weston, and P. R. White (1997). The application of non-linear type filters to television images. *Higher-Order Statistics: Proceedings of the IEEE Signal Processing Workshop on 21-23 July 1997*, 39–42.

Fang, Y., L. Jiao, and J. Pan (2000). Mimo volterra filter equalization using $p^{th}-$order inverse approach. *IEEE Acoust., Speech, Signal Processing 1*, 177–180.

Flioriani, E., T. D. de Wit, and P. LeGal (2000). Nonlinear interactions in a rotating disk flow: From a volterra model to the ginzburg-landau equation. *American Institute of Physics, Chaos 10*(4), 834–847.

Giani, A., P. R. White, W. B. Collis, and M. Weston (2000). Nonlinear techniques for video de-interlacing. *Image Processing, 2000. Proceedings. 2000 International Conference on 10-13 Sept. 2000 2*, 684–687.

Gulunay, N. (1986). Fxdecon and complex wiener prediction filter. *Expanded Abstract, 56th Annual SEG Meeting, Houston*, 279–281.

Gurley, K. R., A. Kareem, and M. A. Tognarelli (1996). Simulation of a class of non-normal random processes. *International Journal of Non-linear Mechanics 31*(5), 601–617.

Hargreaves, N., B. VerWest, R. Wombell, and D. Trad (2003). Multiple attenuation using an apex-shifted radon transform. *Expanded Abstract, 67th Ann. Internat Mtg. SEG*, 1929–1932.

Kellman, P., P. vanGelderen, J. A[], de Zwart, and J. H. Duynb (2003). Method for functional mri mapping of nonlinear response. *NeuroImage 19*, 190–199.

Kim, K., S. B. Kim, E. J. Powers, R. W. Miksad, and F. J. Fischer (1994). Adaptive second-order volterra filtering and its application to second-order drift phenomena. *IEEE Journal of Oceanic Engineering 19*(2), 183–192.

Koh, T. and E. J. Powers (1985). Second-order volterra filtering and its application to nonlinear system identification. *IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-33*(6), 1445–1455.

LeCaillec, J. M. and R. Garello (2004). Comparison of statistical indices using third order statistics for nonlinearity detection. *Signal Processing 84*, 499–525.

Lines, L. R. and S. Treitel (1984). Tutorial: a review of least-squares inversion and its application to geophysical problems. *Geophysical Prospecting 32*(2), 156–186.

Makhoul, J. (1975). Linear prediction: A tutorial review. *Proceedings of the IEEE 63*(4), 561–580.

Marple, S. L. (1980). A new autoregressive spectrum analysis algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-28*(4), 441–454.

Marple, S. L. (1987). *Digital Spectral Analysis with Applications.* Englewood Cliffs, NJ: Prentice-Hall, Inc.

Menke, W. (1989). *Geophysical Data Analysis: Discrete Inverse Theory.* Academic Press, Inc., San Diego, CA.

Naghizadeh, M. and M. D. Sacchi (2007). Multistep autoregressive reconstruction of seismic records. *Geophysics 72*(6), V111–V118.

Nam, S. W. and E. J. Powers (1994). Application of higher order spectral analysis to cubically nonlinear system identification. *IEEE Transactions on Signal Processing 42*(7), 1746–1765.

Nikias, C. L. and J. M. Mendel (1993). Signal processing with higher-order spectra. *IEEE Signal Processing Magazine 07*, 10–37.

Oppenheim, A. V. and R. W. Schafer (1989). *Discrete-Time Signal Processing.* Prentice-Hall, Upper Saddle River-NJ.

Powers, E. J., S. Nam, and S. B. Kim (1990). Adaptive algorithms for the frequency-domain identification of a second-order volterra system with random input. *IEEE Spectrum Estimation and Modeling, Fifth ASSP Workshop on 10-12 Oct 1990 ASSP-5*, 25–29.

Powers, E. J., I. S. Park, S. Mehta, and E. J. Yi (1997). Higher-order statistics and extreme waves. *Higher-Order Statistics, 1997. Proceedings of the IEEE Signal Processing Workshop*, 98–102.

Robinson, E. A. (1954). *Predictive decomposition of time series with applications to seismic exploration.* Ph. D. thesis, MIT, Cambridge, MA.

Robinson, E. A. and S. Treitel (1980). *Geophysical Signal Analysis.* Prentice-Hall.

Rugh, W. J. (1981). *Nonlinear Systems Theory, The Volterra-Wiener Approach.* The Johns Hopkins University Press.

Sacchi, M. D. and H. Kuehl (2001). Arma formulation of fx prediction error filters and projection filters. *Journal of Seismic Exploration* (9), 185–197.

Schetzen, M. (1980). *The Volterra and Wiener Theories of Nonlinear Systems.* A Wiley-Interscience Publication, NY.

Schetzen, M. (2006). *The Volterra and Wiener Theories of Nonlinear Systems (Reprinted edition 2006 with additional material).* Krieger Publising Company, Melbourne, FL.

Sheriff, R. E. (2006). *Encyclopedic Dictionary of Applied Geophysics (4th ed. Geophysical references v.13).* Society of Exploration Geophysicists, Tulsa, OK.

Spitz, S. (1991). Seismic trace interpolation in the $f-x$ domain. *Geophysics 56*(6), 785–796.

Spitz, S. (1999). Pattern recognition, spatial predictability, and subtraction of multiple events. *The Leading Edge 01*, 55–58.

Thomson, D. W. J. (2004). Arctic oscillation (ao) time series, 1899 - june 2002. *www.jisao.washington.edu/ao/*.

Trad, D. (2003). Interpolation and multiple attenuation with migration operators. *Geophysics 68*(6), 2043–2054.

Ulrych, T. J. and M. D. Sacchi (2005). *Information-Based Inversion and Processing with Applications*. Elsevier.

Verschuur, D. J. and A. J. Berkhout (1997). Estimation of multiple scattering by iterative inversion, part ii: Practical aspects and examples. *Geophysics 62*(5), 1596–1611.

Verschuur, D. J., A. J. Berkhout, and C. P. A. Wapenaar (1992). Adaptive surface-related multiple elimination. *Geophysics 57*(9), 1166–1177.

Verschuur, D. J. and R. J. Prein (1999). Multiple removal results from delft university. *The Leading Edge 01*, 86–91.

Wang, R. J. and S. Treitel (1973). The determination of digital wiener filters by means of gradient methods. *Geophysics 38*(2), 310–326.

Weglein, A. B. (1999). Multiple attenuation: an overview of recent advances and the road ahead (1999). *The Leading Edge 01*, 40–44.

Weglein, A. B., F. A. Gasparotto, P. M. Carvalho, and R. H. Stolt (1992). An inverse-scattering series method for attenuating multiples in seismic reflection data. *Geophysics 62*(6), 1975–1989.

Wiener, N. (1942). *Response of a nonlinear device to noise*. MIT Radiation Laboratory Report No:165.

Wiener, N. (1958). *Nonlinear Problems in Random Theory.* MIT Press, Cambridge MA.

Wold, H. (1965). *Bibliography on Time Series and Stochastic Processes.* Oliver and Boyd, London.

Yule, G. U. (1927). On a method of investigating periodicities in disturbed series, with special reference to wolfer's sunspot numbers. *Philosophical Trans. Royal Soc. 226*(A), 267–298.

Zhang, Q., B. S. D. T. Westwick, and K. R. Lutchen (1998). Factors affecting volterra kernel estimation: Emphasis on lung tissue viscoelasticity. *Annals of Biomedical Engineering 26*, 103–116.

Zhong, Y., K.-M. Jan, K. H. Ju, and K. H. Chon (2006). Quantifying cardiac sympathetic and parasympathetic nervous activities using principal dynamic modes analysis of heart rate variability. *American Physiological Society; Am J Physiol Heart Circ Physiol 291*, H1475H1483.