University of Alberta

Parametric reconstruction of multidimensional seismic records

by

Mostafa Naghizadeh

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

Doctor of Philosophy
in
Geophysics

Department of Physics

Fall 2009
Edmonton, Alberta

Examining Committee

Mauricio D. Sacchi, Physics

Doug Schmitt, Physics

Mohsen Razavy, Physics

Ivan Mizera, Mathematical and Statistical Sciences

Felix Herrmann, Earth and Ocean Sciences, UBC

To

all of my teachers,
especially my parents

Abstract

Logistic and economic constraints often dictate the spatial sampling of a seismic survey. The process of acquisition records a finite number of spatial samples of the continuous wave field. The latter leads to a regular or irregular distribution of seismograms. Seismic reconstruction methods are used to recover non-acquired data and to synthesize a dense distribution of sources and receivers that mimics a properly sampled survey. This dissertation examines the seismic sampling problem and proposes algorithms for efficient multidimensional seismic data reconstruction.

In particular, I address the problem of reconstructing irregularly sampled data using multi-dimensional linear prediction filters. The methodology entails a strategy that consists of two steps. First, the unaliased part of the wave field is reconstructed via Fourier reconstruction (Minimum Weighted Norm Interpolation). Then, prediction filters for all the frequencies are extracted from the reconstructed low frequencies. The latter permits the the recovery of aliased data with Multi-Step Auto-Regressive (MSAR) algorithm. The recovered prediction filters are used to reconstruct the complete data in either the f-x domain (MSAR-X) or the f-k domain (MSAR-K).

The thesis also presents the use of Exponentially Weighted Recursive Least Squares (EWRLS) to estimate adaptive prediction filters for f-x seismic interpolation. Adaptive prediction filters are able to model signals where the dominant wavenumbers are varying in space. This concept leads to a f-x interpolation method that does not require windowing strategies for optimal results. In other words, adaptive prediction filters can be used to interpolate waveforms with spatially variant dips.

Acknowledgements

First of all I would like to record my gratitude to Dr. Mauricio D. Sacchi for his supervision, advice, and guidance from the very early stages of this research. Above all and the most needed, he provided me unyielding encouragement and support in various ways. His scientist intuition has made him a constant oasis of ideas and passions in geophysics, which exceptionally inspire and enrich my growth as a student, a researcher and a future geophysicist. I am indebted to him more than he knows.

I appreciate my PhD committee members for their constructive comments on this thesis. I am thankful that in the midst of all their activities, they accepted to be members of the committee.

I gratefully acknowledge Dr. Ulrych Theune for his exceptional C programming course as it provided me with the tools to pursue my research and enhanced the development of my programming skills. I know I will continue to use and value these skills throughout my professional career.

My special thanks go to Dr. Daniel Trad, for being such a great mentor during my summer internship in the CGGVeritas Calgary office and giving me a jump start on dealing with massive amounts of real seismic data.

Collective and individual acknowledgments are also owed to my colleagues at the Signal Analysis and Imaging Group (SAIG). Many thanks go in particular (and in alphabetic order) to Abdolnaser Yousefzadeh, Cristina Moldoveanu-Constantinescu, Juefu Wang, Sam Kaplan, Sharareh Karmand, Soner Bekleric, and Somanath Misra for scientific discussions and prominent friendships. It was my pleasure working with them since my arrival at the University of Alberta. I also convey special acknowledgement to Rhonda Kajner for her support and help with the editing of my publications and this dissertation.

My parents, Marziyeh and Abdollah, deserve special mention for their inseparable support and care. They were my very first teachers and the most important of them all. I am not sure if words are capable of expressing my gratefulness to them. I also like to thank my siblings Sadeq, Mohammad, Roghayyeh, and Reyhaneh.

Finally, I would like to thank everybody who was important to the successful completion of this thesis, as well as expressing my apology that I could not mention their names one by one.

Contents

Appendices

List of Tables

List of Figures

List of symbols and abbreviations

Variables

$\mathbf{G}$ Data matrix

$\mathbf{I}$ Identity matrix

$\mathbf{Q}$ Scaling function (spectrum of sampling function)

$\mathbf{R}$ Correlation matrix

$\hat{\mathbf{R}}$ Estimate of correlation matrix

$\hat{\mathbf{R}}^{AC}$ Estimate of correlation matrix using autocorrelation method

$\hat{\mathbf{R}}^{COV}$ Estimate of correlation matrix using covariance method

$\hat{\mathbf{R}}^{FB}$ Estimate of correlation matrix using modified covariance method (forward-backward)

$\mathbf{a}$ AR model vector

$\mathbf{p}$ Prediction filter coefficients

$\mathbf{r}$ Correlation vector

$\mathcal{E}$ Expectation value

$E(z)$ Innovation process

$G(z)$ Transfer function

J_b Cost function for backward prediction

J_f Cost function for forward prediction

$S_{xx}(f)$ Power Spectral Density (PSD)

$\hat{S}_{xx}^{PER}(f)$ PSD estimated by periodogram

$\hat{S}_{xx}^{BT}(f)$ PSD estimated by Blackman-Tukey method

$\hat{S}_{xx}^{AR}(f)$ PSD estimated by AR modeling

$X(f)$ Discrete Fourier Transform of signal

$X(z)$ Z-transform of discrete signal

Γ_l Box car function with length l

Υ Bandlimiting function

Υ_{pf} Multi-band-pass filter built using AR spectrum of data

a_k Autoregressive model parameters

k Wavenumber

p Ray parameter

$\hat{r}_{xx}$ Estimated ACF

$x(n)$ Discrete signal

λ Forgetting factor

μ Trade-offset parameter for inversion

$\hat{\mu}$ Estimated mean value of a process

ρ Cost function

σ^2 Variance of the innovation process (noise)

ω Temporal frequency

Abbreviations

1D One-dimensional

2D Two-dimensional

3D Three-dimensional

ACF Auto-Correlation Function

AR Autoregressive

DFT Discrete Fourier Transform

EWRLS Exponentially Weighted Recursive Least Squares

FFT Fast Fourier Transform

IFFT Inverse Fast Fourier Transform

MSAR MultiStep AutoRegressive

MSAR-K MSAR reconstruction in the *f-k* domain

MSAR-X MSAR reconstruction in the *f-x* domain

MWNI Minimum Weighted Norm Interpolation

NMO Normal Move-Out

NUFFT Non-Uniform Fast Fourier Transform

PSD Power Spectral Density

RC Reflection Coefficient (for Levinson-Durbin algorithm)

RLS Recursive Least Squares

f-k frequency-wavenumber

f-x frequency-space

t-x time-space

CHAPTER 1

Introduction

The goal of the seismic method is to gather, process and invert seismic wave fields to obtain an interpretable image of the subsurface. The seismic method can be summarized as follows: a controlled pulse of energy provokes a disturbance in the surface of the earth that propagates into the subsurface and reflections on geological boundaries propagate back to the surface of the earth. This information is recorded by an array of receivers, saved, processed for SNR enhancement, and finally, used to produce an image of the subsurface via inversion methods. The final image is used as a starting point for geophysical interpretation toward the understanding of geological structures, sedimentological environments and processes that, in the case of applied seismology, serve to find, delineate and exploit accumulation of hydrocarbons.

In 2D seismology, sources of energy and receivers are deployed along a line (seismic line). The latter produces an experiment where the recorded wave field depends on 3 variables: time, source and receiver positions along the acquisition line. In 3D seismology one works with an areal distribution of sources and receivers given rise to a 5 dimensional data volume: time plus 2 coordinates (x, y or North - East) for each source and receiver. Figures 1.1a and 1.1b show a schematic representation of the geometry used for 2D and 3D seismic surveys, respectively. Each seismogram is defined by a time series with a unique spatial label given by its associated source and receiver coordinates. The time axis is discretized via the acquisition system that can easily meet adequate sampling criteria to preserve information content. This thesis will focus on the study of sampling criteria for the spatial variables that define a seismic survey. In particular, I am interested in algorithms designed to reconstruct improperly sampled wave fields.

It is often the case that logistic and economic constraints dictate the spatial sampling of

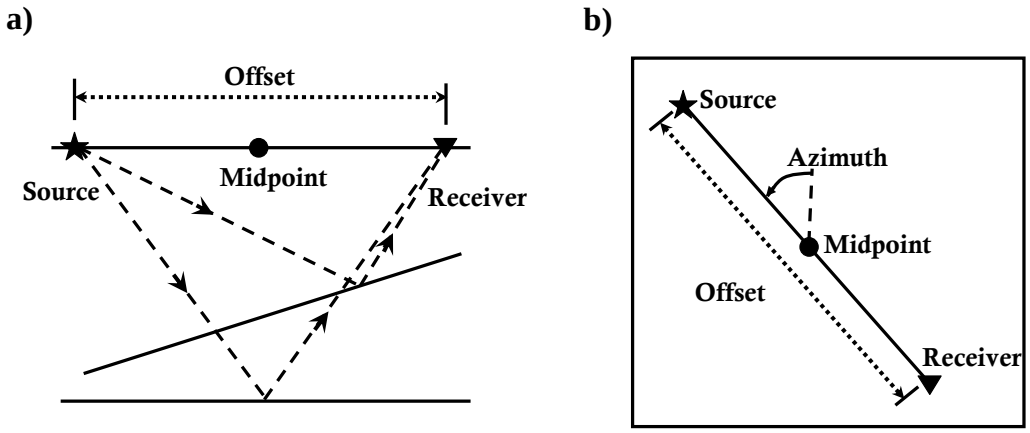

Figure 1.1: Schematic representation of geometries for 2D (a) and 3D (b) seismic surveys.

seismic data. Wave fields are continuous, in other words, energy reaches the surface of the earth everywhere in our area of study. The process of acquisition records a finite number of spatial samples of the continuous wave field leading to a regular or irregular distribution of sources and receivers.

Many important techniques for removing coherent noise and imaging the earth interior have stringent sampling requirements which are often not met in real surveys. In order to avoid loss of information content one should sample according to the Nyquist criterion. When this criterion is not honored, reconstruction can be used to recover the data to a denser distribution of sources and receivers to mimic a properly sampled survey. The final result of the reconstruction stage has a significant impact on subsequent seismic processing steps such as resolution enhancement, noise removal and imaging or inversion.

1.1 Review of seismic reconstruction methods

One of the essential points that separates seismic data interpolation from ordinary interpolation techniques is the need to honor the spatial continuity of seismic wave fields. Methods for reconstruction of seismic wave fields can be classified into two main categories: Wave equation based methods and Signal processing approaches.

The wave-equation methods utilize the physics of wave propagation to reconstruct seismic data. In general, the idea can be summarized as follows. An operator that maps seismic wave fields to a physical domain is used to set a transformation from data space to model space. The data are transformed back to a new geometry via inverting the operator. It

is basically a regression approach where the regressors are built based on wave equation principles (in general, approximations to kinematic ray theoretical solutions of the wave equations). The methods proposed by Ronen (1987), Fomel (2003), Malcolm et al. (2005), and Stolt (2002) fall under this category. These methods require the knowledge of some sort of velocity distribution in the earth's interior (migration velocities, rms velocity, stacking velocities). While wave equation based methods are very important, this thesis will not investigate these methods and will focus only on signal processing approaches.

Reconstruction of seismic data using signal processing approaches is an ongoing research topic in exploration seismology. While they are based on statistical estimation theory, they also utilize information from the physics of wave propagation by taking into account proper apriori information and assumptions. They do not require information of the underlying distribution of velocities and in general are characterized by their computational efficiency and robustness in situations where the optimality conditions on which they were built are not satisfied.

The signal processing methods for seismic data reconstruction often rely on transforming the data to other domains. The most commonly used transformations are Fourier (Duijndam et al., 1999; Liu and Sacchi, 2004; Zwartjes and Gisolf, 2006), Radon (Trad et al., 2002), and Curvelet (Hennenfent and Herrmann, 2008; Herrmann and Hennenfent, 2008) transforms. In Fourier inversion methods, one attempts to invert numerically the Fourier complex spectra of the data and use the inverted results to perform reconstruction. Fourier reconstruction of seismic data is an ill-posed problem and the solution needs to be extracted with the aid of regularization techniques (Dahlquist and Bjorck, 2006; Saad, 2004). Fourier reconstruction schemes that use sparsity constraints have been proposed by Sacchi et al. (1998) and Zwartjes and Gisolf (2006). One can also consider a spectral norm that permits to find solutions that are sparse or smooth according to the behavior of the power spectral density of the data. The latter is the core of the so called Minimum Weighed Norm interpolation (MWNI) method (Liu and Sacchi, 2004; Sacchi and Liu, 2005).

The main goal of Fourier reconstruction methods are to recover the spectrum of the original data. Estimating the spectrum of data using Fourier transform is considered as a non-parametric approach. The alternative for non-parametric spectrum estimation is the parametric spectrum estimation. Autoregressive modeling (prediction filtering) leads to parametric spectral estimators. Spitz (1991) and Porsani (1999) introduced seismic trace interpolation methods using prediction filters. These methods operate in the frequency-space (f-x) domain. In both cases low frequency data components in a regular spatial grid are used to estimate the prediction filters which are then used to interpolate high frequency data components. This is the typical interpolation method used for increasing sampling density of regularly sampled aliased data. The equivalent of these interpolation methods in

the frequency-wavenumber f-k domain was introduced by Gulunay (2003) and often referred as f-k interpolation.

1.2 Challenges pertaining seismic data reconstruction

In this section I discuss the factors which determine the success or failure of seismic data reconstruction methods. One, or a combination, of these factors might result in failure of a given reconstruction method. These factors are:

1. **Sampling**: In this dissertation I consider the sampling operator as a function which is applied to the original well-sampled discrete signal and replaces some of its samples with zero values. Such a sampling operator can be a regular sampling function which decimates data samples regularly or an irregular one. The later eliminates data samples randomly or in an irregular fashion. The signal processing community has historically preferred regular over irregular sampling schemes. The latter, however, is being revised under new paradigms provided by the theory of *compressed sensing* (Donoho, 2006; Tsaig and Donoho, 2006; Herrmann and Hennenfent, 2008; Herrmann et al., 2008). Compressed sensing establishes the recovery conditions for irregularly sampled signals that can be expressed as superposition of a few basis functions. It also utilizes early ideas of sparse inversion that have been used by the geophysical community for data reconstruction and interpolation (Sacchi et al., 1998). It is important to stress that the current impact of compressed sensing resides on the fact that theorems for stable recovery of under-sampled signals have been provided. In other words, it reveals conditions under which one can recover a signal from an irregular distribution of samples.

2. **Alias**: The extent of spatial alias in seismic data is a crucial factor for the performance of reconstruction algorithms. The lower the amount of aliased energy in the data the easier to interpolate or reconstruct the missing samples. The combination of severe alias and regular sampling can be a detrimental mixture for sparse Fourier reconstruction algorithms (Naghizadeh and Sacchi, 2008). The cure for these situations is utilizing methods that are based on autoregressive modeling (Spitz, 1991). We explore this idea in details in Chapter 4.

3. **Curvature of events**: Linear events in the time-space (t-x) domain have sparse representation in the frequency-wavenumber (f-k) domain. This helps sparse Fourier reconstruction methods to recover missing samples. On the other hand, curved events in the t-x domain do not have a sparse representation in the f-k domain and therefore, they cannot be reconstructed using sparse Fourier reconstruction algorithms. The

most commonly used strategy to overcome this problem is spatial windowing. The latter is needed to approximate curved event by local linear events. Another option is to use an adaptive interpolation method (Naghizadeh and Sacchi, 2009a).

4. **Noise**: From a broader prospective a reconstruction method can be considered a noise elimination method. Based on this definition, the sampling operator, alias, and curvature of data create artifacts that can be treated as noise and need to be removed effectively by the reconstruction algorithm. Therefore, the presence of random or coherent additive noise in the original data can further complicate the ability of the reconstruction method to recover missing samples. This is discussed in Chapter 6.

5. **Dimensionality of data**: It has been discussed that improved reconstruction results can be obtained via multi-dimensional reconstruction methods (Liu et al., 2004; Trad, 2008). For instance, if one spatial dimension is poorly sampled (aliased), reconstruction algorithms can rely on properly sampled extra dimensions to properly reconstruct missing samples. This topic is analyzed in Chapter 3.

6. **Percentage of available samples**: The minimum percentage of available samples required for an optimal reconstruction depends on all of the above mentioned factors, the type of reconstruction algorithm and data complexity. Severely decimated surveys can be reconstructed provided that there is low geological complexity (Liu et al., 2004).

1.3 Motivations

The main motivation for this thesis is to introduce methodologies for the reconstruction of seismic wave fields. The goal is to provide reconstruction methods that can

- handle aliased energy,
- work with regular and irregular sampling operators,
- provide robust reconstruction results under different sampling scenarios,
- utilize the multidimensional nature of data,
- be applied to data with locally variant dip information.

1.4 Contributions

The contribution of this thesis can be summarized as follows

- Investigating the relationship between sampling operators and Fourier reconstruction methods.
- Introducing the multistep autoregressive (MSAR) method for reconstruction of irregularly sampled aliased seismic data.
- Extending the MSAR reconstruction method to the multidimensional case.
- Introducing *f-x* adaptive seismic trace interpolation using the Exponentially Weighted Recursive Least Squares (EWRLS) in order to interpolate regularly sampled data with dip-variant events.

1.5 Organization of this thesis

This dissertation is organized as follows:

Chapter 2 deals with basic definitions pertaining spectrum estimation methods. Various algorithms for non-parametric (Fourier) and parametric (autoregressive) spectrum estimation are provided. Synthetic examples are used to show the benefits of autoregressive spectrum estimation methods, which are the cornerstone of the reconstruction algorithms introduced in this thesis.

In **Chapter 3** the relationship between the sampling function and the performance of Fourier reconstruction methods is analyzed. Synthetic and real data examples are used to show the performance of Fourier reconstruction for various sampling functions.

Chapter 4 introduces the multistep autoregressive (MSAR) reconstruction method. The MSAR algorithm is a combination of a Fourier based method and auto-regressive reconstruction to reconstruct seismic data in the case that the acquisition is irregular and that the average sampling rate is below the Nyquist criterion (Naghizadeh and Sacchi, 2007). This is a novel way of utilizing low frequency information in order to reconstruct high frequency components. Synthetic and real data examples are provided to examine the performance of the MSAR algorithm and compare it to classical Fourier reconstruction methods.

In **Chapter 5** the MSAR algorithm is extended to the multidimensional case. In addition to the theoretical expansion, practical algorithms are provided to effectively deal with multidimensional data. Several synthetic and real data examples are provided to analyze the performance of the multidimensional MSAR algorithm.

In **Chapter 6** an adaptive *f-x* seismic trace interpolation technique is introduced. This method is suitable for interpolation of curved aliased seismic events as well as seismic records with variable lateral dip. Exponentially Weighted Recursive Least Squares (EWRLS) is

used to compute local prediction filters that are used for data interpolation. The proposed method of interpolation is examined with synthetic and real data examples and the results are compared to traditional f-x interpolation method.

Chapter 7 concludes the thesis with a summary of results. In addition, I provide a discussion about the advantages and shortcomings of the proposed reconstruction algorithms.

CHAPTER 2

Spectrum estimation

2.1 Introduction

The reconstruction methods discussed and developed in this dissertation are directly related to the problem of estimating the spectrum of spatial data. Spectral estimation methods are divided into parametric and non-parametric methods (McClellan, 1982). Fourier reconstruction methods use non-parametric spectral estimation methods. Meanwhile, the reconstruction methods which use prediction filters or autoregressive (AR) modeling can be categorized as parametric spectral estimation techniques (Naghizadeh and Sacchi, 2009b). In this chapter, I will review the problem of spectral estimation. Special emphasis is placed on AR spectral estimation methods. The latter are the cornerstone of the new reconstruction methods introduced in this thesis.

2.2 Fourier Transform

Consider a discrete-time (or discrete-space) process $x(n)$ given at integer time/space samples n over the domain $(-\infty, \infty)$. The signal $x(n)$ is considered a uniformly sampled version of a continuous function $x(t)$. The z-transform of $x(n)$ is given by

$$X(z) = \sum_{n=-\infty}^{\infty} x(k) z^{-k}, \tag{2.1}$$

and its Discrete-Time Fourier Transform (DTFT) is obtained by replacing $z = e^{j2\pi f}$

$$X(f) = \sum_{k=-\infty}^{\infty} x(k)e^{-j2\pi fk}, \tag{2.2}$$

where, $j = \sqrt{-1}$. If $x(n)$ is defined only on a confined interval, e.g. $[0, N-1]$, then the Discrete Fourier Transform (DFT) of $x(n)$ is obtained by the following formula

$$X(n) = \sum_{k=0}^{N-1} x(k)e^{-j2\pi nk/N}, \quad n = 0, \ldots, N-1. \tag{2.3}$$

Notice that in DTFT, the Fourier domain is a continuous function, while in DFT both time/space and frequency are discrete. For the DFT it was assumed that the signal $x(n)$ is periodically extended outside the interval $[0, N-1]$ (Oppenheim and Schafer, 1974).

2.3 Stationary Process

To fully characterize a discrete-time process such as $x(n)$, one needs to define the joint Probability Density Function (PDF) of every sample in the process. This requires a massive amount of information to be dealt with. An alternative, which is both practical and closely related to linear modeling, is to use a partial characterization based on second-order statistics. This is carried out by defining the mean of the process

$$\mu(n) \triangleq \mathcal{E}[x(n)], \tag{2.4}$$

and the Auto-Correlation Function (ACF)

$$r_{xx}(n, m) \triangleq \mathcal{E}[x(n)x^*(m)]. \tag{2.5}$$

If a signal can be fully defined by the expressions (2.4) and (2.5), it must be a stationary signal. Determining the exact values of the mean and the ACF of a signal requires an infinite number of signal samples. However, we often have only a finite numbers of samples. Therefore, these parameters have to be estimated from a finite number of samples. To solve the problem we need to assume that one realization of the process is representative of the entire ensemble (ergodicity property) (Papoulis, 1991). Given N observations $x(0), x(1), \ldots, x(N-1)$ of the process $x(n)$, the sample mean is estimated by:

$$\hat{\mu} = \frac{1}{N} \sum_{n=0}^{N-1} x(n), \tag{2.6}$$

and the unbiased estimate of the ACF is given by

$$\hat{r}_{xx}(k) = \frac{1}{N-k} \sum_{n=0}^{N-k-1} x^*(n)x(n+k), \quad k = 0, 1, \ldots, N-1, \tag{2.7}$$

which is consistent but has high variance for large lags k. A biased estimate of the ACF with lower variance is obtained via

$$\hat{r}_{xx}(k) = \frac{1}{N} \sum_{n=0}^{N-k-1} x^*(n)x(n+k), \quad k = 0, 1, \ldots, N-1. \tag{2.8}$$

2.4 Power Spectral Density

A stationary process can be alternatively characterized using its Power Spectral Density (PSD)

$$S_{xx}(f) \triangleq \sum_{k=-\infty}^{\infty} r_{xx}(k)e^{-j2\pi fk}. \tag{2.9}$$

The PSD is real, nonnegative, and it is the Fourier transform of the ACF. For discrete signals with a finite number of samples, the PSD has to be estimated like the mean and the ACF. Spectrum estimation of a signal can be achieved using both non-parametric and parametric methods. I will focus on parametric estimation methods which are of primary importance to this thesis. A brief explanation of non-parametric estimation methods, however, is also provided.

2.4.1 Non-parametric Spectrum Estimation

Non-parametric PSD estimation methods are distinguished by their improvements in variance and resolution. The variance of the estimator can be reduced by trading off frequency resolution (Thomson, 1982). The most commonly used method for non-parametric PSD estimation is the periodogram,

$$\hat{S}_{xx}^{PER}(f) \triangleq \frac{1}{N}|X(f)|^2 = \frac{1}{N}\left|\sum_{n=0}^{N-1} x(n)e^{-j2\pi fn}\right|^2. \tag{2.10}$$

It can be easily proved that the periodogram is the Fourier transform of the biased ACF estimator (Narasimhan and Veena, 2005; Marple, 1987)

$$\hat{S}_{xx}^{PER}(f) = \sum_{k=-(N-1)}^{N-1} \hat{r}_{xx}(k)e^{-j2\pi fk}. \tag{2.11}$$

The bias of the periodogram estimators is given by

$$\varepsilon[\hat{S}_{xx}^{PER}(f)] = W_B(f) * S_{xx}(f), \tag{2.12}$$

where $W_B(f) \propto \mathbf{sinc}^2(f)$ is the Fourier transform of the Bartlet (triangular) window (the convolution of two box-car windows). The variance of the periodogram can be reduced using the average of periodograms extracted from adjacent data windows. In other words, one can subdivide the data into K non-overlapping windows and after computing the periodogram for each segment, use the average of the K periodograms to find the optimal periodogram (Bartlett method). It can also be implemented in an overlapping windows fashion, provided that the condition of the partition of unity is satisfied for the chosen windows (Welch method) (Welch, 1967).

The other alternative to obtain a reliable periodogram, known as the Blackman-Tukey estimate, is to suppress the high lags of the ACF estimate $\hat{r}_{xx}(k)$. This is necessary since at high lags less data points were used for computing $\hat{r}_{xx}(k)$. Hence, the spectrum is given by

$$\hat{S}_{xx}^{BT}(f) = \sum_{k=-M}^{M} w(k)\hat{r}_{xx}(k)e^{-j2\pi fk}, \tag{2.13}$$

where, $0 \leq w(k) \leq w(0) = 1$, $w(-k) = w(k)$, and $w(k) = 0$ for $|k| > M$.

Reliable non-parametric estimates of the spectrum can also be estimated via the multi-taper method (Slepian, 1976; Thomson, 1982). This method utilizes windows defined by a set of special sequences known as Slepian sequences or discrete prolate spheroidal sequences, which are optimally compact in both time and frequency domains.

2.4.2 Parametric Spectrum Estimation

Linear Models

Any stationary random process can be separated into a linearly unpredictable (random) part and a linearly predictable (deterministic) part (Priestly, 1988). A general linear model is represented as filtered white noise

$$x(n) = \sum_{k=0}^{\infty} g_k \varepsilon(n-k) \tag{2.14}$$

where $\varepsilon(n)$ is known as the innovation process. This model is impractical due to the infinite number of parameters g_k. By transforming equation (2.14) to the z-domain we obtain:

$$X(z) = G(z)E(z). \tag{2.15}$$

Assuming that the transfer function $G(z)$ is rational, i.e.:

$$G(z) = B(z)/A(z). \tag{2.16}$$

we can obtain a finite length parameter model

$$x(n) + \sum_{k=1}^{M} a_k x(n-k) = \varepsilon(n) + \sum_{k=1}^{L} b_k x(n-k). \tag{2.17}$$

Equation (2.17) can lead to three well-known models (Narasimhan and Veena, 2005):

1. The autoregressive (AR) model. In this case $b_k = 0$ and therefore, $G(z) = 1/A(z)$.
2. The Moving Average (MA) model. In this case $a_k = 0$ leading to $G(z) = B(z)$.
3. The Autoregressive Moving Average (ARMA). This is the general form containing both AR and MA components.

Autoregressive processes

This thesis focuses on AR Models for data reconstruction. An AR process of order M will satisfy the difference equation

$$\sum_{k=0}^{M} a_k x(n-k) = \varepsilon(n) \tag{2.18}$$

where $a_0 = 1$, a_k are AR parameters and $\varepsilon(n)$ is a white noise process. It is useful to provide an interpretive explanation for the filtering aspects of the AR process. In the z-domain, letting $A(z) = \sum_{k=0}^{M} a_k z^{-k}$, $X(z) = \sum_{k=0}^{\infty} x(k) z^{-k}$ and $E(z) = \sum_{k=0}^{\infty} \varepsilon(k) z^{-k}$, we have

1. The analysis (whitening) filter $A(z)$ turns the process into white noise

$$A(z)X(z) = E(z). \tag{2.19}$$

which is a stable filter.

2. The generator filter $1/A(z)$ turns white noise into the process $x(n)$. This can be easily shown by considering the the following expression

$$X(z) = \frac{1}{A(z)} E(z). \tag{2.20}$$

The operation is stable if all the poles are located inside the unit circle (Takalo et al., 2005; Marple, 1980).

In the next section we will look into methods of AR parameter estimation and its relation to linear prediction methods. For the moment, I assume known AR parameters. The AR spectral estimator is defined as

$$\hat{S}_{xx}^{AR}(f) = \frac{\sigma^2}{|A(e^{j2\pi f})|^2} \tag{2.21}$$

where σ^2 is the variance of the innovation process.

2.5 AR and Linear Prediction

2.5.1 Yule-Walker Formula

Yule-Walker equations summarize the optimal solution for an AR process. Starting from equation (2.18) and multiplying both sides by $x^*(n-l)$ we have

$$\mathcal{E}\left[\sum_{k=0}^{M} a_k x(n-k) x^*(n-l) \right] = \mathcal{E}[\varepsilon(n) x^*(n-l)] \tag{2.22}$$

Now, for $l > 0$, the noise $\varepsilon(n)$ is uncorrelated with past output samples $x(n-l)$, which leads to Yule-Walker equations

$$\sum_{k=0}^{M} a_k r_{xx}(l-k) = 0, \quad l > 0, \tag{2.23}$$

or

$$r_{xx}(l) = -a_1 r_{xx}(l-1) - a_2 r_{xx}(l-2) - \ldots - a_M r_{xx}(l-M), \quad l > 0, \tag{2.24}$$

Equation (2.24) can be written in matrix formulation for $l = 1, 2, \ldots, M$ as follows (Narasimhan and Veena, 2005)

$$\begin{bmatrix} r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(M-1) \\ r_{xx}^*(1) & r_{xx}(0) & \cdots & r_{xx}(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}^*(M-1) & r_{xx}^*(M-2) & \cdots & r_{xx}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{bmatrix} = - \begin{bmatrix} r_{xx}^*(1) \\ r_{xx}^*(2) \\ \vdots \\ r_{xx}^*(M) \end{bmatrix}, \tag{2.25}$$

or

$$\mathbf{Ra} = -\mathbf{r}. \tag{2.26}$$

Therefore, the AR parameters are derived from the coefficients of the ACF via

$$\mathbf{a} = -\mathbf{R}^{-1}\mathbf{r}, \tag{2.27}$$

where $\mathbf{R}$ is always invertible due to its Toeplitz-Hermitian form (Marple, 1987). For $l = 0$ Yule-Walker equation becomes:

$$\sum_{k=0}^{M} a_k r_{xx}(-k) = \mathcal{E}[\varepsilon(n)x^*(n)] = \mathcal{E}[\varepsilon(n)\varepsilon^*(n)] = \sigma^2. \tag{2.28}$$

By adding equation (2.28) to equation (2.25), an augmented version of the Yule-Walker equation is obtained

$$\begin{bmatrix} r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(M) \\ r^*_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(M-1) \\ \vdots & \vdots & \ddots & \vdots \\ r^*_{xx}(M) & r^*_{xx}(M-1) & \cdots & r_{xx}(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_M \end{bmatrix} = - \begin{bmatrix} \sigma^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{2.29}$$

or

$$\begin{bmatrix} r_{xx}(0) & \mathbf{r}^H \\ \mathbf{r} & \mathbf{R} \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \sigma^2 \\ \mathbf{0} \end{bmatrix} \tag{2.30}$$

Equation (2.29) shows that there is a one-to-one correspondence between the parameters of the AR process combined with the innovation variance $[\sigma^2, a_1, a_2, \ldots, a_M]$, and the first $M + 1$ ACF values $[r_{xx}(0), r_{xx}(1), \ldots, r_{xx}(M)]$.

2.5.2 Linear Prediction

Linear prediction deals with the problem of predicting the present value $x(n)$ of a random process via a linear combination of its closest M values (one-step forward linear prediction). The predicted value is given by

$$\hat{x}(n) = \sum_{k=1}^{M} p_k x(n-k) = \mathbf{p}^H \mathbf{x}(n-1), \tag{2.31}$$

where $\mathbf{x}(n) \triangleq [x(n), x(n-1), \ldots, x(n-M+1)]$. The goal is to find the optimal prediction filter coefficients $\mathbf{p}_o$ which minimize the mean-squared prediction error (i.e. power of error):

$$\begin{aligned}
\rho &= \mathcal{E}\left[\ |e(n)|^2\ \right] \\
&= \mathcal{E}\left[\ |x(n) - \hat{x(n)}|^2\ \right] \\
&= \mathcal{E}\left[\ |(x(n) - \mathbf{p}^H\mathbf{x}(n-1))(x(n) - \mathbf{p}^H\mathbf{x}(n-1))^*|^2\ \right] \\
&= \mathcal{E}[x(n)x^*(n)] - \mathbf{p}^H\mathcal{E}[\mathbf{x}(n-1)x^*(n)] \\
&\quad -\mathcal{E}[x(n)\mathbf{x}^H(n-1)]\mathbf{p} + \mathbf{p}^H\mathcal{E}[\mathbf{x}(n-1)\mathbf{x}^H(n-1)]\mathbf{p} \\
&= r_{xx}(0) - \mathbf{p}^H\mathbf{r} - \mathbf{r}^H\mathbf{p} + \mathbf{p}^H\mathbf{R}\mathbf{p} \qquad (2.32)
\end{aligned}$$

To minimize $\rho(\mathbf{P})$ with respect to $\mathbf{P}$, we set the derivatives equal to zero

$$\begin{aligned}
\frac{\partial\rho(\mathbf{p})}{\partial\mathbf{p}} &= \frac{\partial\mathbf{p}^H\mathbf{R}\mathbf{p}}{\partial\mathbf{p}} - \frac{\partial(\mathbf{p}^H\mathbf{r} + \mathbf{r}^H\mathbf{p})}{\partial\mathbf{p}} + \frac{\partial r_{xx}(0)}{\partial\mathbf{p}} \\
&= 2\mathbf{R}\mathbf{p} - 2\mathbf{r} + 0 = 0 \qquad (2.33)
\end{aligned}$$

therefore, the optimal prediction filter is given by

$$\mathbf{p}_o = \mathbf{R}^{-1}\mathbf{r}. \qquad (2.34)$$

A close comparison between the solution of the AR parameters, equation (2.27), and the solution for the linear prediction operator in equation (2.34) reveals that

$$p_k = -a_k, \quad K = 1, 2, \ldots, M. \qquad (2.35)$$

Combining equation (2.18) and (2.31) using equality (2.35) leads to

$$x(n) = -\sum_{k=1}^{M} a_k x(n-k) + \varepsilon(n) = \hat{x}(n) + \varepsilon(n), \qquad (2.36)$$

which means the white noise that drives the filter is the unpredictable portion of each sample. Overall, it can be concluded that linear prediction and AR modeling are equivalent concepts.

2.5.3 Levinson-Durbin Algorithm

The solution of Yule-Walker equation (2.27) involves the inversion of the $M \times M$ matrix. Due to the structure of the matrix $\mathbf{R}$ (Toeplitz, Hermitian and positive semidefinite), the

inversion can be carried out in a very efficient manner that requires $O(M^2)$ operations via the Levinson-Durbin algorithm (Vaidyanathan, 2008; Narasimhan and Veena, 2005).

The basic idea of the Levinson-Durbin algorithm is to start solving a first-order prediction problem and to increase the order up to the desired order M. The Levinson-Durbin algorithm can be summarized as follows

Algorithm 2.1: Levinson-Durbin algorithm for AR parameter estimation

Input:
Order of AR process: M
ACF of the signal: $r_{xx}(k) \quad k = 0, 1, \ldots, M$
Output:
Estimated AR parameters: $a_k(i) \quad k = 0, 1, \ldots, M, \quad i = 0, 1, \ldots, k.$

begin
  Initialization:
  $a_i(0) = 1 \quad i = 0, 1, \ldots, M$
  $E_0 = r_{xx}(0), \quad k = 1$
  while $k \leq M$ **do**
    Compute RC: $a_k(k) = k_k = -\frac{\sum_{i=0}^{k-1} a_{k-1}(i) r_{xx}(i-k)}{E_{k-1}}$
    for $i = 1$ **to** $k - 1$ **do**
      Compute filter: $a_k(i) = a_{k-1}(i) + a_k(k) a^*_{k-1}(k - i)$
    end
    Compute error power: $E_k = (1 - |a_k(k)|^2) E_{k-1}$
    k=k+1
  end
end

The coefficients $|k_k|$ are called reflection coefficients. They are an indicator of how helpful the sample $x(n - k)$ is in predicting $x(n)$ from the $(k - 1)$st order prediction. The value $|k_k| = 0$ means that the added sample did not contribute to the prediction process and $|k_k| = 1$ means that the desired prediction was achieved (Markel and Gray, 1976).

2.5.4 AR Properties

The AR process has several interesting properties which makes it very useful for signal processing. Here we briefly list some of its properties:

1. The prediction error filter $A(z)$ is a minimum-phase filter. In other words

(a) if $\mathbf{R}$ is positive-definite, all zeros of $A(z)$ lie inside the unit circle. This results in a stable generator filter $1/A(z)$.

(b) If $\mathbf{R}$ is singular (i.e. if $x(n)$ consists of M sinusoids), all zeros of $A(z)$ lie on the unit circle (at the frequencies of the sinusoids). Also in this case, the prediction error power would be zero and the process is perfectly predictable (Narasimhan and Veena, 2005).

2. The prediction error power E_k decreases monotonically from $E_0 = r_{xx}(0)$ and reaches a plateau after the true order M. This makes the power of the prediction error a useful indicator to determine the AR order M.

2.6 Practical AR Estimation

Equation (2.27) shows that the problem of the AR parameter estimation depends on the estimator of ACF. When the data consist of a few samples (short time/space series) the procedure utilized to estimate the ACF becomes important. One can also bypass the estimation of the ACF and estimate the AR parameters by operating directly on data. A matrix representation will serve to clarify these ideas and define the type of algorithms that are used in this thesis for AR parameter estimation (Narasimhan and Veena, 2005).

Given N observed data points $\{x(0), x(1), \ldots, x(N-1)\}$, the data matrix formulation estimates $\mathbf{R}$ directly from data instead of using the ACF. Consider the autocorrelation matrix written in the following form

$$\hat{\mathbf{R}} = \frac{1}{C}\mathbf{G}^H\mathbf{G}, \tag{2.37}$$

where $\mathbf{G}$ is the data matrix formed by concatenation of adjacent observation vectors and C is an appropriate constant. Different AR estimators can be obtained by the way $\mathbf{G}$, is defined

$$\mathbf{G} = [\ldots, \mathbf{x}(n-1), \mathbf{x}(n), \mathbf{x}(n+1), \ldots]^H \tag{2.38}$$

where $\mathbf{x}(n) \triangleq [x(n), x(n-1), \ldots, x(n-M)]$. Therefore, the AR spectrum estimation is reduced to building the data matrix $\mathbf{G}$. I will discuss three different methods that exploit the aforementioned idea.

2.6.1 Autocorrelation Method

Autocorrelation method solves the augmented Yule-Walker equation by replacing the true ACF with the biased ACF estimator:

$$\hat{r}_{xx}(k) = \frac{1}{N} \sum_{n=0}^{N-k-1} x^*(n)x(n+k), \quad k = 0, 1, \ldots, M, \tag{2.39}$$

The correlation matrix of the augmented Yule-Walker equation is estimated as

$$\hat{\mathbf{R}}^{AC} = \begin{bmatrix} r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(M) \\ r^*_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(M-1) \\ \vdots & \vdots & \ddots & \vdots \\ r^*_{xx}(M) & r^*_{xx}(M-1) & \cdots & r_{xx}(0) \end{bmatrix} = \frac{1}{N}\mathbf{G}^H\mathbf{G} \tag{2.40}$$

where $(N+M) \times (M+1)$ data matrix $\mathbf{G}$ is given by:

$$\mathbf{G}^{AC} = \begin{bmatrix} x(0) & x(1) & \cdots & x(M) & x(M+1) & \cdots & x(N-1) & 0 & \cdots & 0 \\ 0 & x(0) & \cdots & x(M-1) & x(M) & \cdots & x(N-2) & x(N-1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x(0) & x(1) & \cdots & x(N-M-1) & x(N-M) & \cdots & x(N-1) \end{bmatrix}^H,$$

where it is assumed that the data before $n = 0$ and after $n = N-1$ are zero.

The correlation matrix $\hat{\mathbf{R}}^{AC}$ is Toeplitz, Hermitian and positive semi-definite and therefore the filters are ensured to be stable. The Toeplitz structure of the correlation matrix allows for the implementation of the fast and efficient Levinson-Durbin algorithm. Due to the unreasonable assumption made by replacing unknown data samples by zeros, however, the results might not be so accurate (Robinson and Treitel, 1980). This is particular true when working with short time/space series.

2.6.2 Covariance Method

The Covariance method solves (Ulrych and Clayton, 1976):

$$\begin{bmatrix} c_{xx}(0,0) & c_{xx}(0,1) & \cdots & c_{xx}(0,M) \\ c_{xx}(1,0) & c_{xx}(1,1) & \cdots & c_{xx}(1,M) \\ \vdots & \vdots & \ddots & \vdots \\ c_{xx}(M,0) & c_{xx}(M,1) & \cdots & c_{xx}(M,M) \end{bmatrix} \begin{bmatrix} 1 \\ \hat{a}(1) \\ \vdots \\ \hat{a}(M) \end{bmatrix} = \begin{bmatrix} \hat{\sigma}^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.41}$$

where the ACF is estimated by the correlation matrix defined by

$$r_{xx}(j-k) \approx c_{xx}(j,k) \triangleq \frac{1}{N-M} \sum_{n=M}^{N-1} x^*(n-j)x(n-k). \tag{2.42}$$

In this method the correlation matrix is estimated via $\hat{\mathbf{R}}^{COV} = \frac{1}{N-M}\mathbf{G}^H\mathbf{G}$, where the $(N-M)\times(M+1)$ data matrix $\mathbf{G}$ is given by

$$\mathbf{G}^{COV} = \begin{bmatrix} x(M) & x(M+1) & \cdots & x(N-1) \\ x(M-1) & x(M) & \cdots & x(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ x(0) & x(1) & \cdots & x(N-M-1) \end{bmatrix}^H. \tag{2.43}$$

Notice that no assumption was made about the data before $n=0$ and after $n=N-1$. The correlation matrix $\hat{\mathbf{R}}^{COV}$ is Hermitian and positive semi-definite, but it is not necessarily Toeplitz, and therefore it could lead to an unstable generator filter.

2.6.3 Modified Covariance Method

The modified covariance method minimizes the average of forward and backward prediction error powers, thus also known as the Forward-Backward (FB) algorithm. It has the same structure as the covariance method but the ACF estimate $c_{xx}(j,k)$ is changed to

$$c_{xx}(j,k) \triangleq \frac{1}{2(N-M)} \left(\sum_{n=M}^{N-1} x^*(n-j)x(n-k) + \sum_{n=0}^{N-1-M} x(n+j)x^*(n+k) \right). \tag{2.44}$$

For this method the correlation matrix is estimated via $\hat{\mathbf{R}}^{BF} = \frac{1}{2(N-M)}\mathbf{G}^H\mathbf{G}$, with the $2(N-M)\times(M+1)$ data matrix $\mathbf{G}$ given by

$$\mathbf{G}^{BF} = \begin{bmatrix} x(M) & x(M+1) & \cdots & x(N-1) & x^*(0) & x^*(1) & \cdots & x^*(N-M-1) \\ x(M-1) & x(M) & \cdots & x(N-2) & x^*(1) & x^*(2) & \cdots & x^*(N-M) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x(0) & x(1) & \cdots & x(N-M-1) & x^*(M) & x^*(M+1) & \cdots & x^*(N-1) \end{bmatrix}^H.$$

Again no assumption was made about the data before $n=0$ and after $n=N-1$. The modified covariance method has better resolution than the covariance method (Robinson and Treitel, 1980; Narasimhan and Veena, 2005).

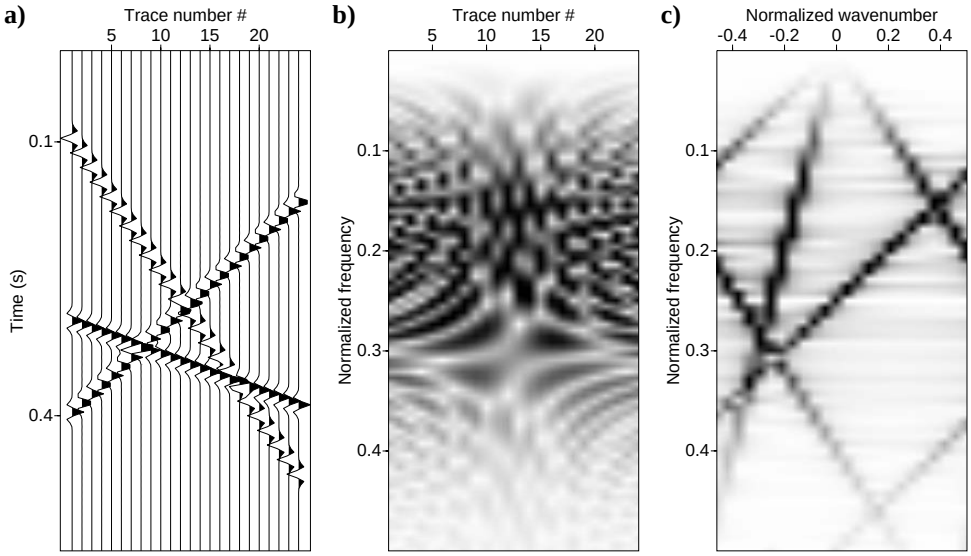

Figure 2.1: A Synthetic 2D seismic section. a) Data in the t-x domain. b) Data in the f-x domain. c) Data in the f-k domain.

2.7 Comparison of parametric and non-parametric spectrum estimations

In this section I compare the performances of the Levinson-Durbin and the modified covariance methods (parametric spectrum estimation methods) with the the classical non-parametric spectrum estimation method, the periodogram, using synthetic examples. Figure 2.1a shows a synthetic seismic record composed of 3 linear events in the time-space (t-x) domain. By applying Fourier transform with respect to time the data are transformed to the frequency-space (f-x) domain (Figure 2.1b). In the f-x domain the spatial samples of each frequency are considered a complex harmonic signal. By performing Fourier transform on the spatial samples of each frequency, the data are transformed to the frequency-wavenumber domain (Figure 2.1c). In the f-k, for a given frequency, only a limited number of wavenumbers should be present.

In order to examine the performance of spectral estimation methods, the spatial samples of the normalized frequency [1] 0.1 in the f-x domain is chosen. Figure 2.2a shows the real (solid line) and imaginary (dashed line) parts of the selected signal. Figure 2.2b shows the normalized frequency 0.1 in the f-k domain (Fourier transform of 2.2a). Figure 2.2b

[1] Normalized frequency and wavenumber axes are obtained by considering $\Delta t = 1$ and $\Delta x = 1$, respectively. This means that in order to obtain frequency axes in Hz and Cycles/m one must divide the normalized ones by Δt and Δx, respectively.

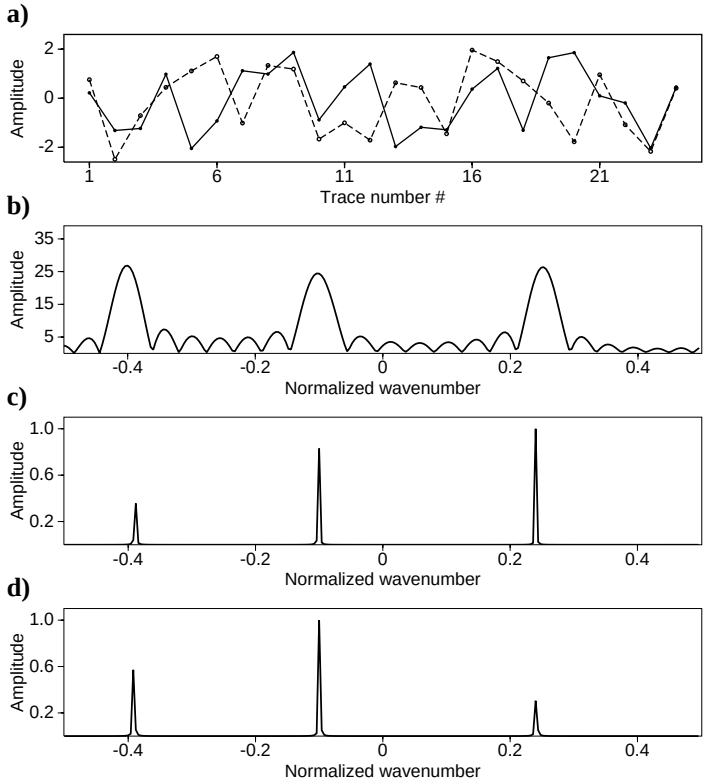

Figure 2.2: a) Real (solid line) and imaginary (dashed line) parts of *f-x* data (Figure 2.1b) at normalized frequency 0.1. b) Non-parametric (periodogram) spectrum estimator of the data in (a). c) Parametric spectrum of (a) computed via Levinson-Durbin method. d) Parametric spectrum of (a) computed via the modified covariance method.

portrays the non-parametric spectrum estimator of Figure 2.2a. Figures 2.2c and 2.2d show the parametric spectrum estimator of Figure 2.2a using Levinson-Durbin and Modified covariance methods, respectively. For both methods the length of prediction filter was equal to $M = 3$. It is clear that parametric estimation methods yield high resolution wavenumber estimators.

To further investigate the performance of parametric spectrum estimation methods, I chose the spatial samples of the normalized frequency 0.3 in the *f-x* domain (Figure 2.3a). The desired wavenumbers at normalized frequency 0.3 (Figure 2.3b) are located very close to each other in contrast to the well-separated wavenumbers in the normalized frequency 0.1 (Figure 2.2b). Figures 2.3c and 2.3d represent the parametric spectrum estimator of Figure 2.3a using Levinson-Durbin and modified covariance methods, respectively. The Levinson-Durbin and modified covariance methods identify two and three closely located wavenumbers, re-

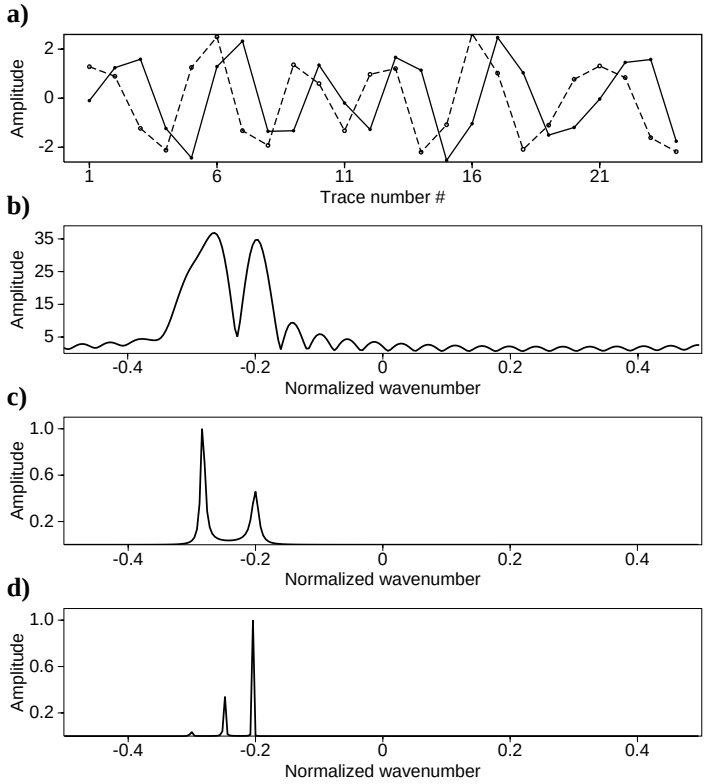

Figure 2.3: a) Real (solid line) and imaginary (dashed line) parts of *f-x* data (Figure 2.1b) at normalized frequency 0.3. b) Non-parametric (periodogram) spectrum estimation of data in (a). c) Parametric spectrum of (a) computed via Levinson-Durbin method. d) Parametric spectrum of (a) computed via modified covariance method.

spectively. This means that the modified covariance method yields results with a higher resolution than the Levinson-Durbin method.

The main benefits of parametric spectrum estimation are obtained when working with data sets composed of a small number of samples. Figure 2.4a shows the first 5 traces of Figure 2.1a. The *f-x* and *f-k* representations of Figure 2.4a are depicted in Figures 2.4b and 2.4c, respectively. Notice that, due to the small number of traces, the signals are almost undetectable in both *f-x* and *f-k* domains. However, since Figure 2.4a is a portion of the data in Figure 2.1a, one should expect to obtain the similar spectral estimators for each frequency.

Figure 2.5a shows the real (solid line) and imaginary (dashed line) parts of the spatial samples in the normalized frequency 0.1 in Figure 2.4. In addition, Figure 2.5b shows the normalized frequency 0.1 in the *f-k* domain (Fourier transform of 2.5a). The non-parametric

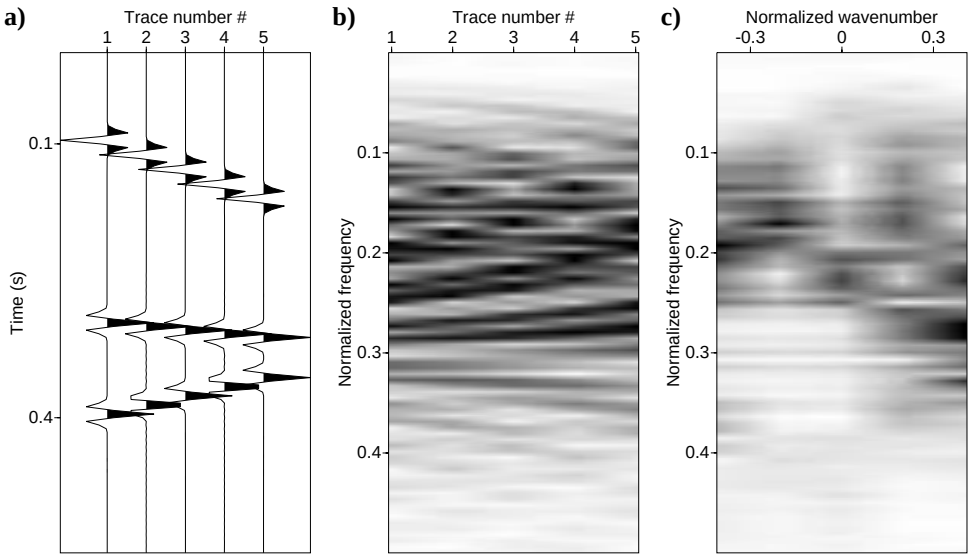

Figure 2.4: A small window of data in Figure 2.1a (First 5 traces). a) Data in the t-x domain. b) Data in the f-x domain. c) Data in the f-k domain.

spectrum estimation method (Figure 2.5b) was unable to identify the desired wavenumber samples. Notice that even applying Fourier transform with padded zeros would only result in a smooth spectrum rather than a high resolution spectrum. Figures 2.5c and 2.5d represent the parametric spectrum estimation of Figure 2.5a using Levinson-Durbin and modified covariance methods, respectively. One by one comparisons of Figures 2.5c and 2.5d and Figures 2.2c and 2.2d shows that even with very small number of data samples parametric spectrum estimation methods provide high resolution estimates of the spectrum of the data.

Figures 2.6a and 2.6b show the spatial samples and wavenumber samples of the signal at the normalized frequency 0.3 in Figure 2.4, respectively. Figures 2.6c and 2.6d represent the parametric spectrum estimation of Figure 2.6a using Levinson-Durbin and modified covariance methods, respectively. The results indicate that, even for closely located spectral values and small number of data samples, parametric spectral analysis gives a reliable estimate of the data spectrum. It can be concluded that the Modified covariance method gives a slightly better estimation of the spectrum of data compared to the Levinson-Durbin method.

Finally, we examine the spectrum estimation methods for the case of noisy data. Figure 2.7a shows the data in Figure 2.1a which is corrupted by additive noise. Figures 2.7b and

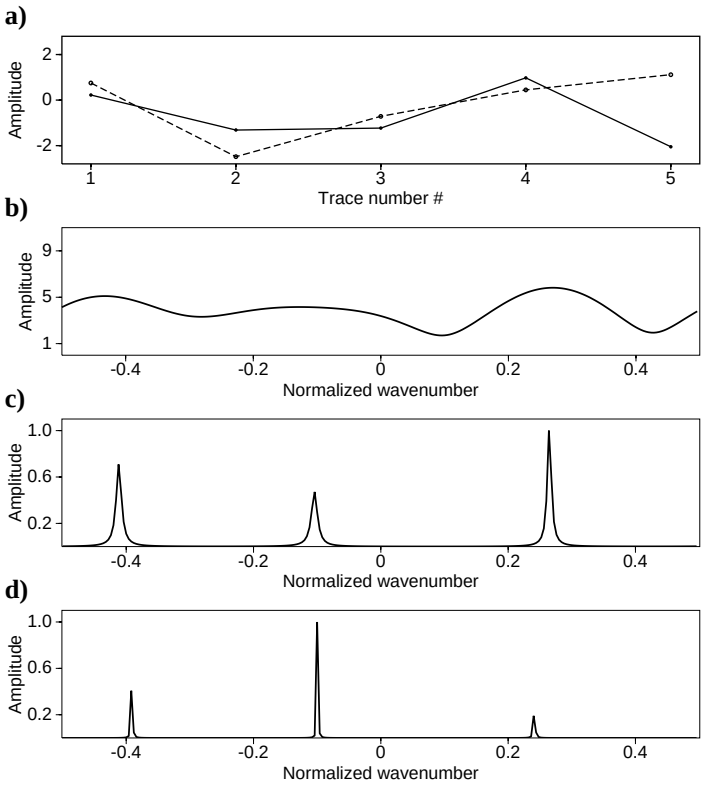

Figure 2.5: a) Real (solid line) and imaginary (dashed line) parts of f-x data (Figure 2.4b) at normalized frequency 0.1. b) Non-parametric (periodogram) spectrum estimator of the data in (a). c) Parametric spectrum of (a) computed via the Levinson-Durbin method. d) Parametric spectrum of (a) computed via the modified covariance method.

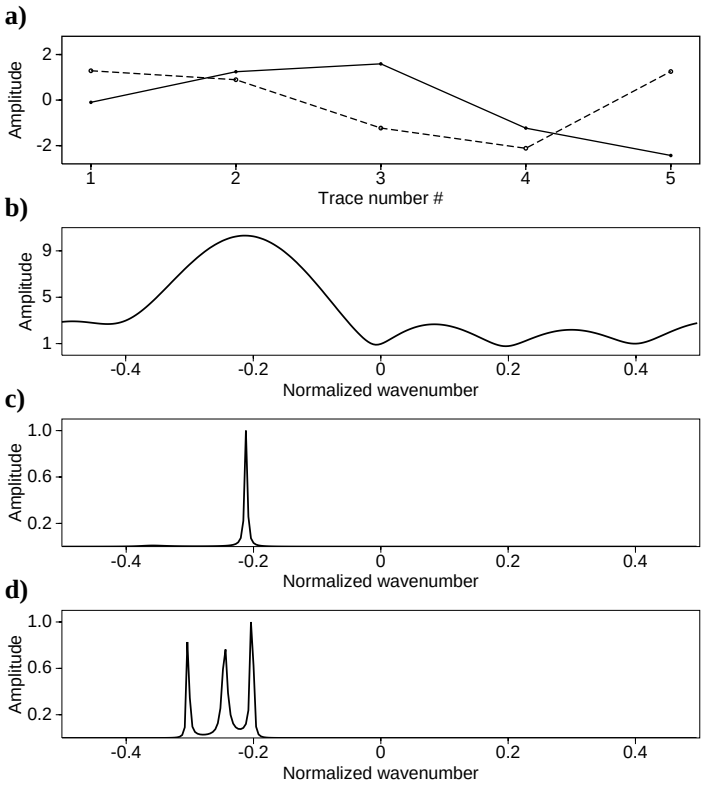

Figure 2.6: a) Real (solid line) and imaginary (dashed line) parts of *f-x* data (Figure 2.4b) at normalized frequency 0.3. b) Non-parametric (periodogram) spectrum of the data in (a). c) Parametric spectrum of (a) computed via the Levinson-Durbin method. d) Parametric spectrum of (a) computed via the modified covariance method.

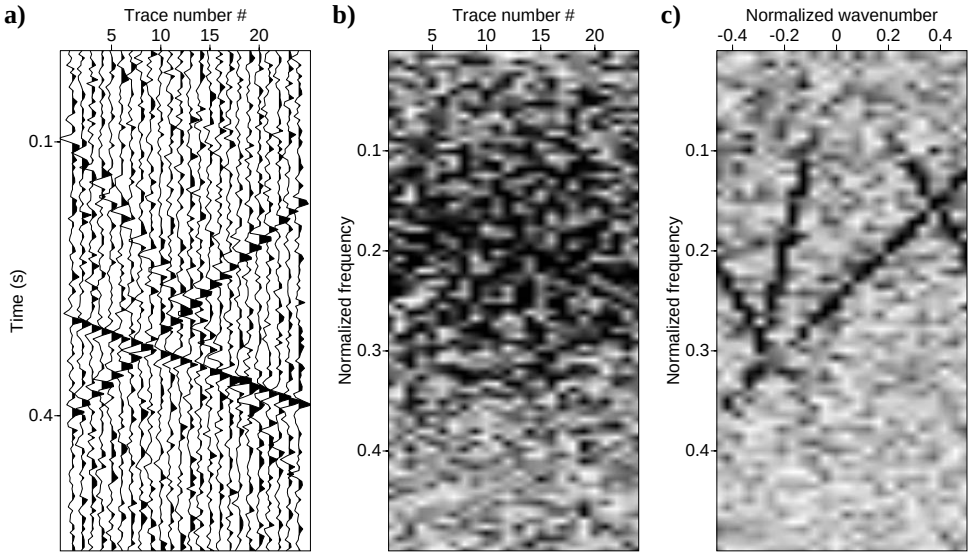

Figure 2.7: Data in Figure 2.1a corrupted by random noise. a) Data in the t-x domain. b) Data in the f-x domain. c) Data in the f-k domain.

2.7c depict the f-x and f-k panels of the data in Figure 2.7a.

Figures 2.8a and 2.9a show the signals in Figure 2.7b at the normalized frequencies 0.1 and 0.3, respectively (the solid line represents the real parts and the dashed line represents the imaginary parts of the data). The amplitudes of the Fourier transforms of Figures 2.8a and 2.9a are shown in Figures 2.8b and 2.9b. Figures 2.8c and 2.9c show the result of spectrum estimation using the Levinson-Durbin method for the data in Figures 2.8a and 2.9a. Figures 2.8d and 2.9d represent the resulting spectra obtained via the modified covariance method for the data in Figures 2.8a and 2.9a. The existence of noise in the data samples results in a wider and smoother spectral estimation in the parametric spectrum estimator (Figures 2.8c, 2.9c, 2.8d and 2.9d). However, the location of peaks in the estimated spectra are found very close to the expected locations.

In the following chapters, the modified covariance method is adopted to estimate the AR coefficients required by the seismic data reconstruction algorithms proposed in this thesis. Since in theory, the AR model parameters are equal to the negative of the prediction filter components, in this thesis, these two concepts are utilized interchangeably as if they are the same operations. This means any reference to the prediction filter estimation as well as the AR model parameter estimation in the text should be translated as applying the modified covariance method to the data.

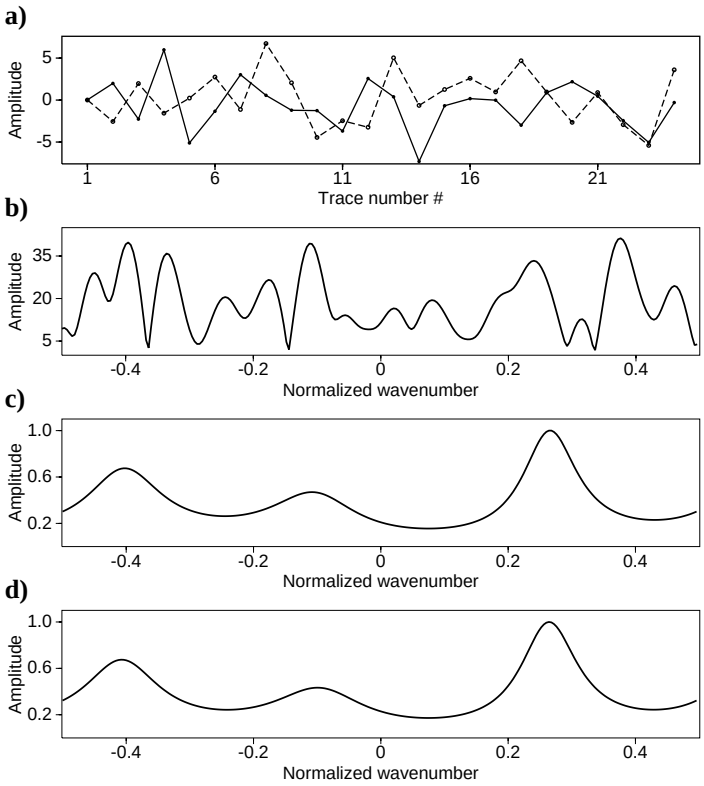

Figure 2.8: a) Real (solid line) and imaginary (dashed line) parts of *f-x* data (Figure 2.7b) at normalized frequency 0.1. b) Non-parametric (periodogram) spectrum of the data in (a). c) Parametric spectrum of (a) computed via the Levinson-Durbin method. d) Parametric spectrum of (a) computed via the modified covariance method.

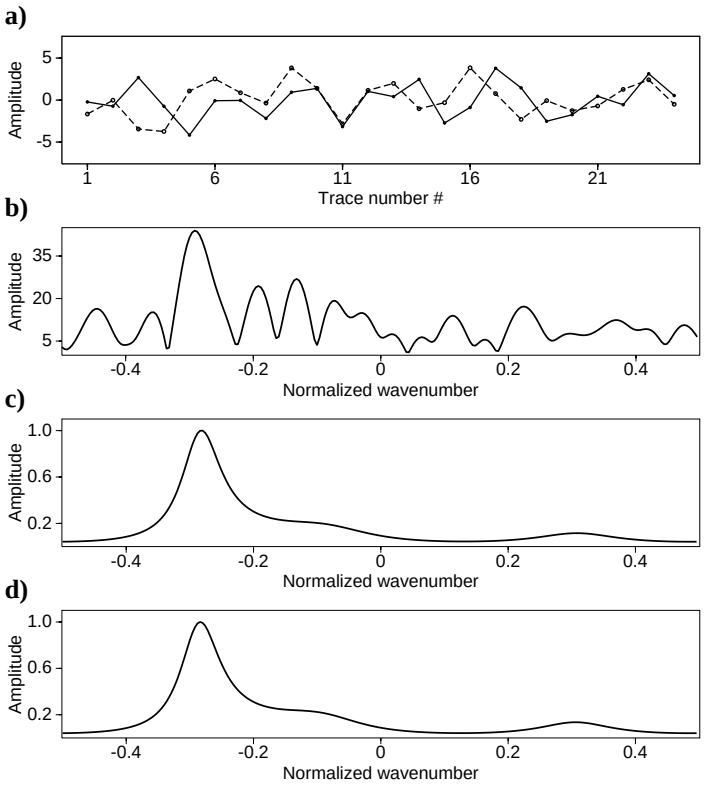

Figure 2.9: a) Real (solid line) and imaginary (dashed line) parts of *f-x* data (Figure 2.7b) at normalized frequency 0.3. b) Non-parametric (periodogram) spectrum of the data in (a). c) Parametric spectrum of (a) computed via the Levinson-Durbin method. d) Parametric spectrum of (a) computed via the modified covariance method.

2.8 Conclusions

In this chapter I have reviewed basic principles of spectrum estimation using non-parametric (periodogram) and parametric (autoregressive modeling) methods. Efficient and practical algorithms are provided for autoregressive parameter estimation. Using synthetic examples, it was shown that AR parameter estimation methods are more accurate in resolving the spectrum of short data data sets than the non-parametric spectrum analysis. The AR parameter estimation methods introduced in this chapter will be used in Chapters 4 and 6 for seismic data reconstruction purposes. It is important to stress that the AR models were not only used to model time dependant processes but spatial processes as well (Haining, 2003; Ma, 2004).

CHAPTER 3

Sampling Functions and Fourier Reconstruction

3.1 Introduction

The sampling theorem is an interesting topic of chief importance in the physical sciences and in engineering. Sampling methods can be classified into two groups, uniform and nonuniform sampling methods. In the uniform sampling method, a signal is sampled periodically at constant intervals. This leads to recovery conditions based on the well-known Nyquist sampling theorem. An overview of uniform sampling and its properties is given by Unser (2000). For nonuniform sampling methods, a distinction could be made between methods using known sampling position, and those where the positions at which observations are taken are not precisely known (Vandewalle et al., 2007). However, in this thesis I will only consider the methods with known sample locations. Another important group of sampling methods are multi-channel sampling methods. These methods are used for super-resolution reconstruction by the signal processing community (Sroubek et al., 2007). In geophysical applications, Ronen (1987) used a multi-channel sampling approach to reconstruct an alias-free zero-offset section from aliased common offset sections.

Regardless of the adopted sampling method, the truly important step is to reconstruct the original signal from sampled data. For discrete signals, the reconstructed target is a regularly sampled signal with a desired sampling rate. In general, reconstruction algorithms must be designed in accordance with the type of sampling. It is possible, however, to make reconstruction methods independent of the sampling if the original signal honors specific

assumptions. The most commonly used assumption is band-limitation. In other words, band-limited signals can be completely recovered from a wide range of sampling methods (Feuer and Goodwin, 2005; Strohmer, 1997; Eldar, 2006). Duijndam et al. (1999) utilized band-limiting assumptions in order to reconstruct the irregularly sampled seismic records in the spatial directions. Another important assumption is sparseness. These methods, commonly known as compressed sensing in signal processing, assume that the signal has a sparse representation in the Fourier domain (or any other domain)(Donoho, 2006; Tsaig and Donoho, 2006; Hennenfent and Herrmann, 2007). These reconstruction methods retrieve sparse solution via introducing a sparsity norm in the formulation of the problem (Sacchi and UIrych, 1996; Hennenfent and Herrmann, 2006). In geophysical applications, Liu and Sacchi (2004) and Liu (2004) used iteratively updated weighting functions of the data spectrum as a mean to introduce sparsity in the Fourier spectrum. Similar line of research was initiated by Zwartjes and Gisolf (2006) who compared the performance of several sparsity norms on seismic data reconstruction methods.

In this chapter I will investigate the effects of various sampling schemes on the performance of Fourier reconstruction algorithms that utilize band-limitation and/or sparsity assumptions. I will inquire how the spectrum of data changes when a subset of the original samples are replaced by zeros. The latter provides important information to design strategies to eliminate the resulting artifacts which is the essence of every reconstruction method. First, I develop the mathematical framework for describing sampling and its response in the Fourier domain.

3.2 Sampling Theory

3.2.1 One-dimensional sampling functions

We consider a discrete signal $\mathbf{x}$ of length N represented by

$$\mathbf{x} = \{x(0), x(1), x(2), \ldots, x(N-1)\}. \tag{3.1}$$

Let us replace $N - M$ samples of $\mathbf{x}$ by zeros. Indicating the location of available samples by $\mathcal{H} = \{h(0), h(1), h(2), \ldots, h(M-1)\}$, the signal with missing samples, $\mathbf{x}_s$ is represented via the following expression

$$x_s(n) = \begin{cases} x(n) & n \in \mathcal{H} \\ 0 & n \notin \mathcal{H} \end{cases},$$

The Discrete Fourier Transform (DFT) of the original data, $\mathbf{X}$, is given by

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-i2\pi \frac{nk}{N}} \quad k = 0, 1, 2, \ldots, N-1. \tag{3.2}$$

The DFT of signal with missing samples, $\mathbf{X}_s$ can be written as

$$X_s(k) = \sum_{n=0}^{M-1} x(h(n)) e^{-i2\pi \frac{h(n)k}{N}} \quad k = 0, 1, 2, \ldots, N-1, \tag{3.3}$$

where we have considered that only the available samples contribute to the summation.

The function $\mathbf{Q}$

$$Q(k) = \frac{1}{N} \sum_{m=0}^{M-1} e^{-i2\pi \frac{h(m)k}{N}} \quad k = 0, 1, 2, \ldots, N-1. \tag{3.4}$$

is called the scaling function or the frequency response of the sampling operator. It can be proved that $\mathbf{X}_s = \mathbf{X} * \mathbf{Q}$, where $*$ represents the convolution operator. Writing the convolution operator explicitly, we have:

$$\begin{aligned} X_s(k) &= \sum_{j=0}^{N-1} X(j) Q(k-j) \\ &= \sum_{j=0}^{N-1} [\sum_{n=0}^{N-1} x(n) e^{-i2\pi \frac{nj}{N}}][\frac{1}{N} \sum_{m=0}^{M-1} e^{-i2\pi \frac{h(m)(k-j)}{N}}] \\ &= \sum_{j=0}^{N-1} [\sum_{n=0}^{N-1} x(n) e^{-i2\pi \frac{nj}{N}}][\frac{1}{N} \sum_{m=0}^{M-1} e^{-i2\pi \frac{h(m)k}{N}} e^{i2\pi \frac{h(m)j}{N}}] \\ &= \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x(n) e^{-i2\pi \frac{h(m)k}{N}} [\frac{1}{N} \sum_{j=0}^{N-1} e^{-i2\pi \frac{nj}{N}} e^{i2\pi \frac{h(m)j}{N}}]. \end{aligned} \tag{3.5}$$

Now due to the orthogonality of Fourier series, the expression inside the bracket in the last line of (3.5) is nonzero if $n = h(m)$. This means that the expression (3.5) can be reduced to:

$$\begin{aligned} X_s(k) &= \sum_{m=0}^{M-1} x(h(m)) e^{-i2\pi \frac{h(m)k}{N}} [\frac{1}{N}.N] \\ &= \sum_{m=0}^{M-1} x(h(m)) e^{-i2\pi \frac{h(m)k}{N}} \qquad k = 0, 1, 2, \ldots, N-1, \end{aligned} \tag{3.6}$$

which is the same result as equation (3.3). Therefore, the scaling function (3.4) convolves the spectrum of the original signal to yield the spectrum of the signal with missing samples.

The scaling function is a complex valued vector but for simplicity only its absolute value will be depicted.

Figure 3.1 shows the different sampling functions (left side) and their associated scaling functions (right side) for regular decimation operators. The decimation factors are $1, 2, 3, 4$ and 5 from top to bottom, respectively. For the decimation factor equal to 1 (no decimation), the scaling function has only one impulse at normalized frequency 0.0. This means that the spectrum of the data in this case would remain untouched. The decimation with factor 2 replaces every other sample of the data with zero and its scaling function contains two impulses. The scaling function is responsible for the replication of the original spectrum. Increasing the decimation factor results in more equally-spaced replications of the original spectrum. Also, as the figures show, the replicated spectra have all the same amplitude.

Figure 3.2 depicts 5 different random sampling operators (left side) and their scaling function in the Fourier domain (right side). In this examples 50% of the samples are missing. The scaling function of a nonuniform sampling operator has a different structure than the uniform one. It has nonzero values for all of its components inside the fundamental interval of the spectrum. Interestingly, it has only one large value which appears at the exact location of the uniform sampling case with no decimation. The rest of its components are small. As the number of missing samples increases, more components of the scaling function become close to one. Figure 3.3 shows 5 random sampling operators with 85% missing samples. As the number of missing samples increases, the amplitude of the impulses in the scaling function increases.

Figure 3.4 shows the scaling functions (right side) of 5 different sampling operators (left side) which are combinations of regular and random sampling operators. Each sampling function is composed of a regular decimation with factor 2 followed by 50% random elimination of the remaining samples. The scaling functions contain two high amplitude impulses (response of regular decimation) while the rest of the impulses have low amplitudes (due to random sampling).

Figure 3.5 shows the scaling functions (right side) for different sizes of gaps (left side) inside a regularly sampled data set. The gap leads to replications of the spectrum of the original signal in its neighbor samples. The amplitudes of the replicated spectra increases with the size of the gap. Figure 3.6 depicts the scaling functions (right side) for the extrapolation of regularly sampled data (left side). The scaling functions for the extrapolation problem (Figure 3.6) is very similar to the one for the gap problem (Figure 3.5).

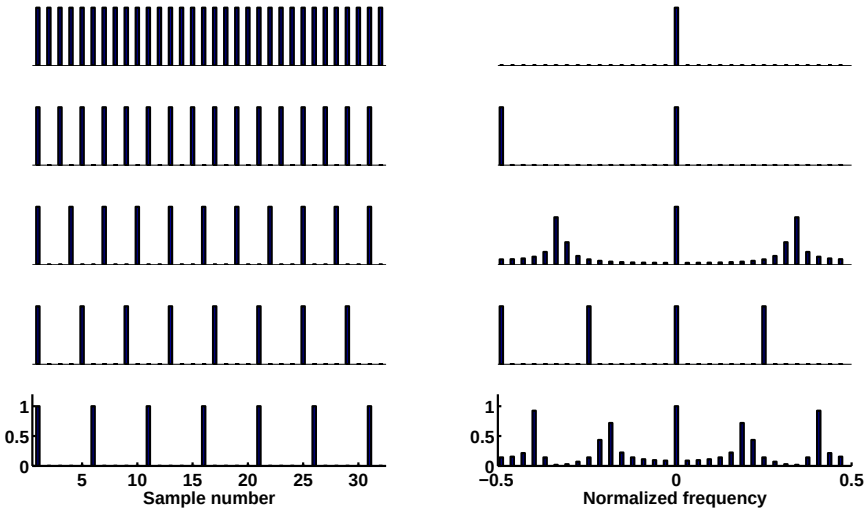

Figure 3.1: Sampling operators (left) and their correspondent scaling functions (right). The decimation factors are $1, 2, 3, 4$ and 5 from top to bottom.

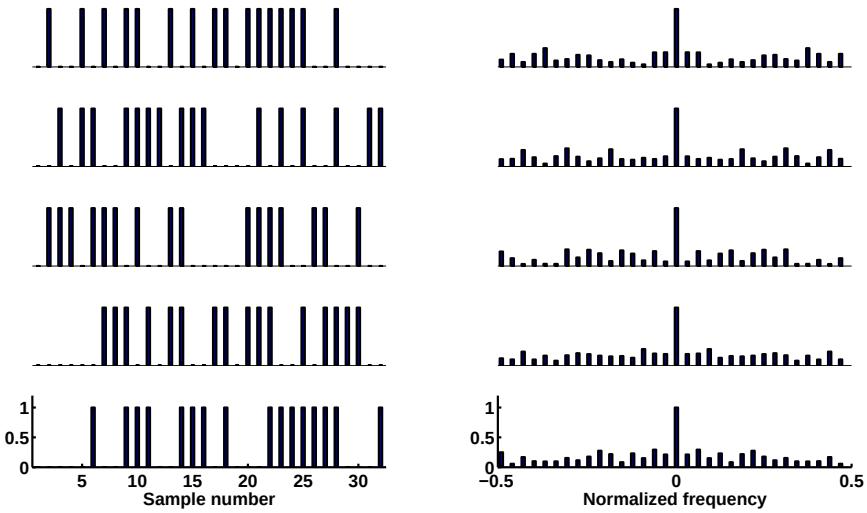

Figure 3.2: Sampling operators (left) and their associated scaling functions (right). For each example 50% of data were randomly removed.

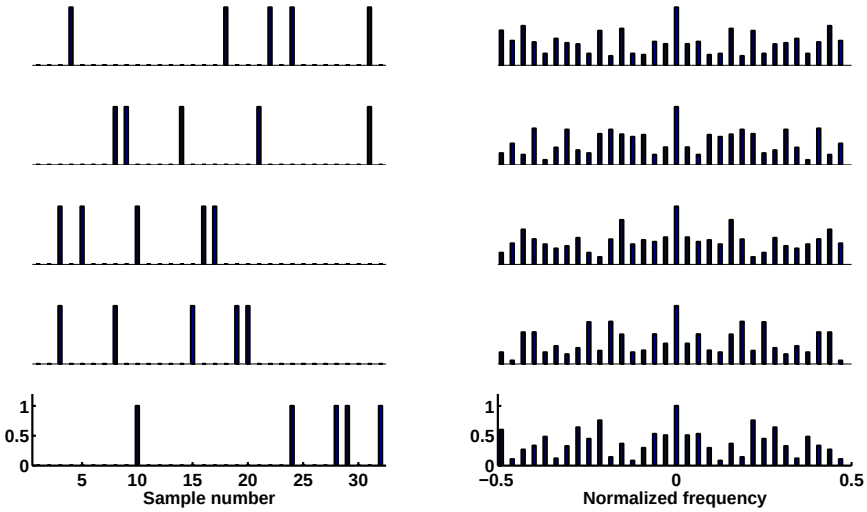

Figure 3.3: Sampling operators (left) and their associated scaling functions (right). For each example 85% of data were randomly eliminated.

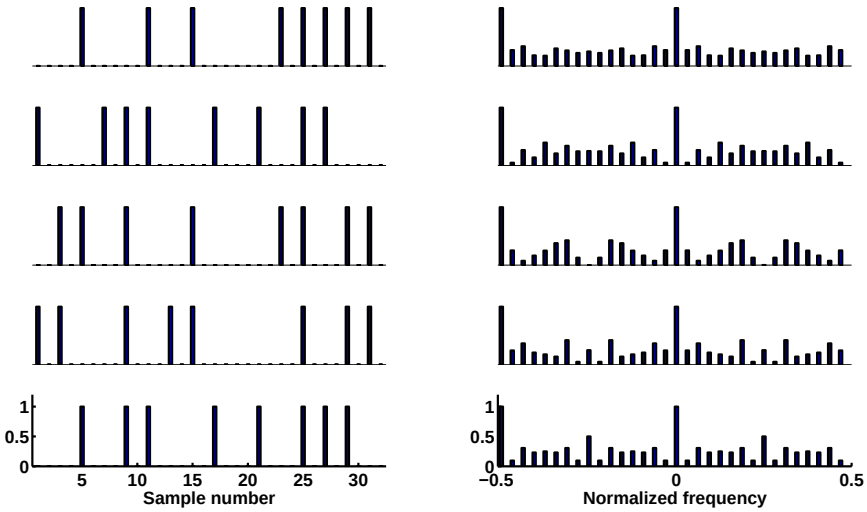

Figure 3.4: Sampling operators (left) and their associated scaling functions (right). Each example consists of a regular decimation with factor 2 followed by 50% random elimination of the remaining samples.

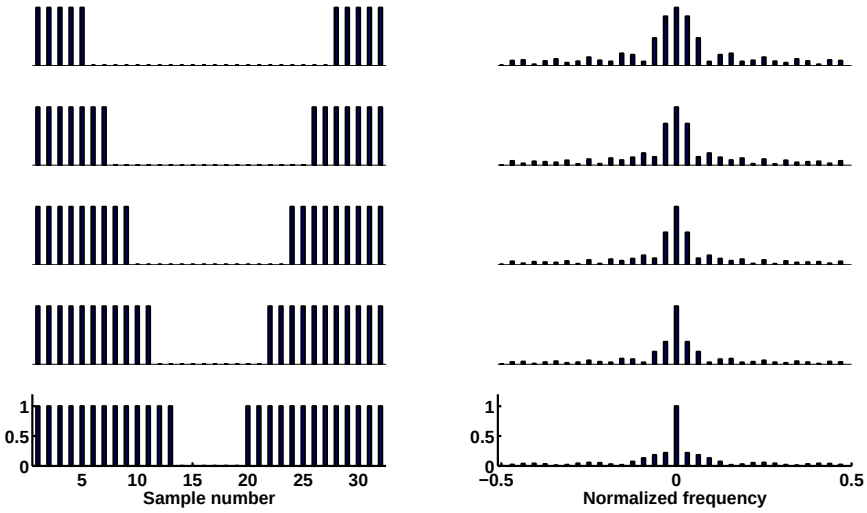

Figure 3.5: Sampling operators (left) and their associated scaling functions (right) for a gap inside the data. The size of the gap decreases from top to bottom.

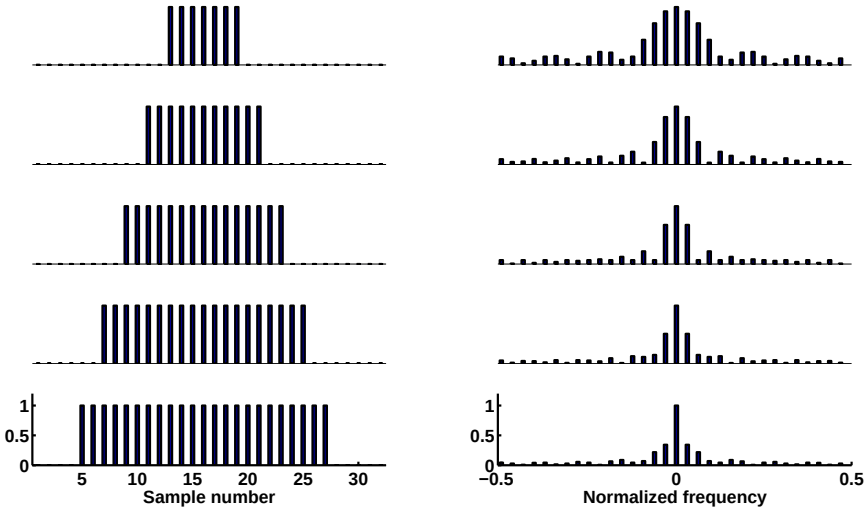

Figure 3.6: Sampling operators (left) and their associated scaling functions (right) for the extrapolation case. The number of missing samples is decreasing from top to bottom.

3.2.2 Multidimensional sampling functions

The formulas introduced in the previous section can be easily generalized to multidimensional cases. Suppose we have a 2D signal, $\mathbf{u}$, with size $N_x \times N_y$ represented by

$$u(n_x, n_y) \quad \begin{cases} n_x \in 0 : N_x - 1 \\ n_y \in 0 : N_y - 1 \end{cases}.$$

Assume that we arbitrarily reserve M samples of data and set the rest to zero to obtain $\mathbf{u}_s$, the signal with missing samples. Let us indicate the location of the available samples by

$$\mathcal{H} = \{(h_x(m), h_y(m)) \mid m \in 0 : M - 1\},$$

where, h_x and h_y indicate the location of the available samples on the axis X and Y, respectively. The DFT of the original data, $\mathbf{U}$, is given by:

$$U(k_x, k_y) = \frac{1}{N_y} \sum_{n_y=0}^{N_y-1} \left(\frac{1}{N_x} \sum_{n_x=0}^{N_x-1} u(n_x, n_y) e^{-i2\pi \frac{k_x n_x}{N_x}} \right) e^{-i2\pi \frac{k_y n_y}{N_y}}.$$

Now, the DFT of the data with missing samples, $\mathbf{U}_s$, is obtained by

$$\mathbf{U}_s = \mathbf{U} \circledast \mathbf{Q},$$

where, $\circledast$ represents the 2D convolution operator and $\mathbf{Q}$ is the 2D scaling function given by

$$Q(k_x, k_y) = \frac{1}{N_x . N_y} \sum_{m=0}^{M-1} e^{-i2\pi \frac{k_x h_x(m)}{N_x}} e^{-i2\pi \frac{k_y h_y(m)}{N_y}}. \tag{3.7}$$

Generalizing the scaling function for more dimensions is straightforward.

The absolute value (or amplitude) of the 2D scaling function, $\mathbf{Q}$, is plotted for various 2D sampling operators. Figure 3.7b shows the scaling function of a discrete data with no missing samples (Figure 3.7a). The scaling function is a single impulse in the middle of the 2D spectrum. Figure 3.7c shows a 2D sampling operator that eliminates every other slice of data in the Y direction. Filled circles represent the available samples and crosses show the missing samples. Figure 3.7d shows the scaling function of Figure 3.7c. The scaling function has two impulses located at normalized wavenumbers $(k_x, ky) = \{(0, 0), (0, -0.5)\}$. Figure 3.7e shows another 2D decimation function that samples the data in a chessboard pattern. Figure 3.7f shows the scaling function of Figure 3.7e. In comparison to Figure 3.7d, the impulse at the normalized wavenumber $(0, -0.5)$ is moved to $(-0.5, -0.5)$.

This is an interesting phenomenon and can be efficiently utilized for data reconstruction

purposes. Assume that the original 2D signal in Figure 3.7b is band-limited in the k_x axis in the interval $[-0.2, 0.2]$ and full band in the k_y axis. Therefore, in Figure 3.7d the replication impulse produce by the sampling operator is inside the interval where the signal lives. In Figure 3.7f, however, the impulse resides outside the region containing the signal. Hence, in the case of Figure 3.7f one can utilize band-limitation on the k_x axis to erase the sampling artifacts. Conversely, in Figure 3.7d band-limitation cannot be used to eliminate the sampling artifacts.

Figure 3.8a shows a sampling function which has eliminated two Y slices between each available Y slice. Figure 3.8c shows another sampling function which also has eliminated two samples between available samples and the available samples are located in all of the Y and X slices. Figures 3.8b and 3.8d show the scaling functions of the Figures 3.8a and 3.8c. In Figure 3.8d the band-limitation information on the k_x axis can be used to eliminate the artifacts. This is not possible in the situation depicted in Figure 3.8b.

Figures 3.9a and 3.9c show two 2D random sampling operators with 50% and 85% missing samples, respectively. Figures 3.9b and 3.9d depict the scaling functions of 3.9a and 3.9c, respectively. It is interesting that in both scaling functions the amplitude of impulses are very small compared to the impulse of the original data located in the center of the plots. Therefore, for randomly sampled multidimensional data with a high percentage of missing samples, the artifacts should have small amplitudes. Figure 3.9e shows a 2D sampling operator for a gap inside the data. The scaling function of 2D gap sampling (Figure 3.9f) contains relatively high amplitude impulses only in the vicinity of the main original impulse.

3.3 Fourier reconstruction

In this section we derive basic formulas for Fourier reconstruction methods using general inverse theory. First, we formulate the general least squares problem and give its general solution based on appropriate regularization terms. Two specific choices of regularization terms are explained in details.

3.3.1 General inversion problem

An ill-posed or ill-conditioned linear system can be represented by

$$\mathbf{d} = \mathbf{G}\mathbf{m} \tag{3.8}$$

where, $\mathbf{d}$ and $\mathbf{m}$ show data and model space vectors, respectively. Matrix $\mathbf{G}$ is an operator which is built based on governing physical or mathematical rules that connect model

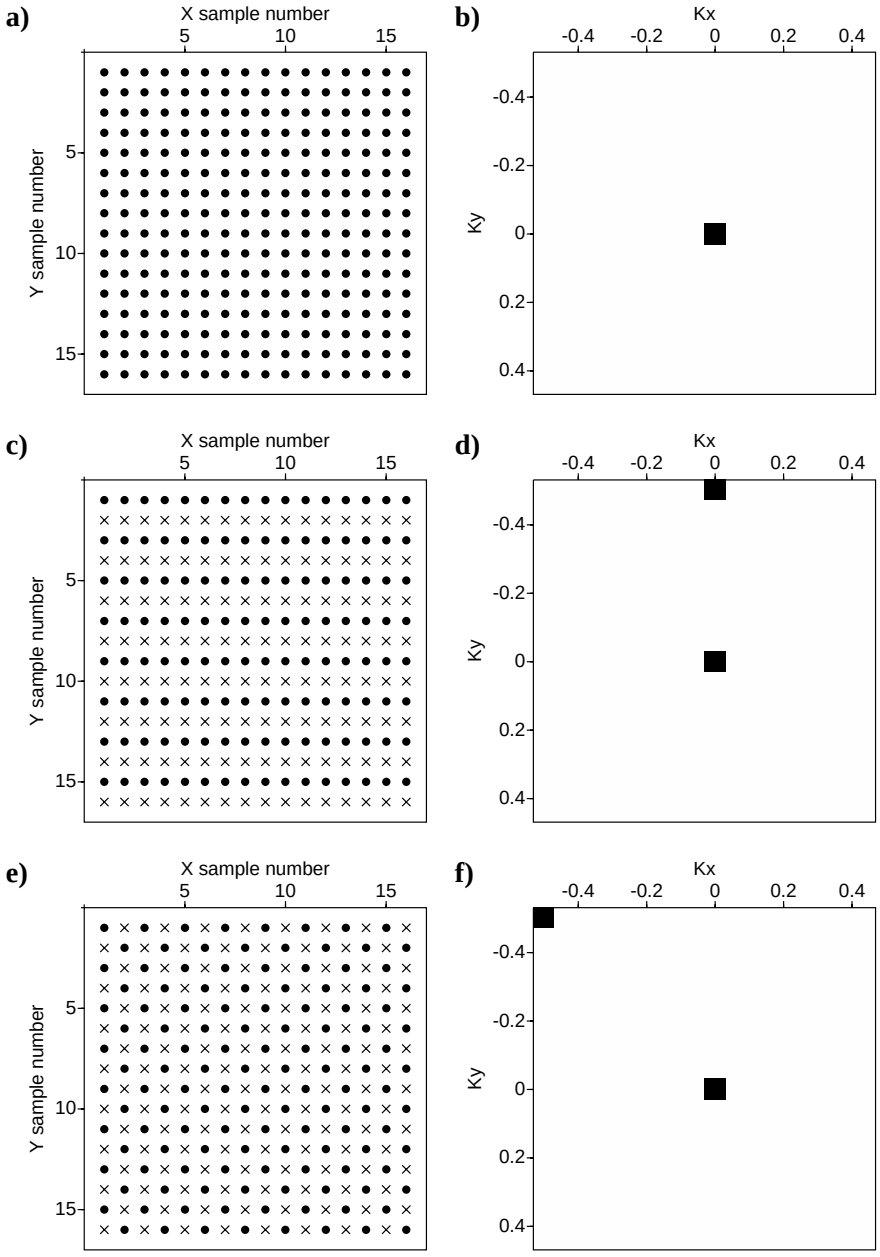

Figure 3.7: a) Full 2D sampling operator. c) 2D regular sampling operator with decimation of factor 2 in the Y direction. e) 2D regular sampling operator with chessboard pattern decimation. b), d), and f) are 2D scaling functions of (a), (c), and (e), respectively.

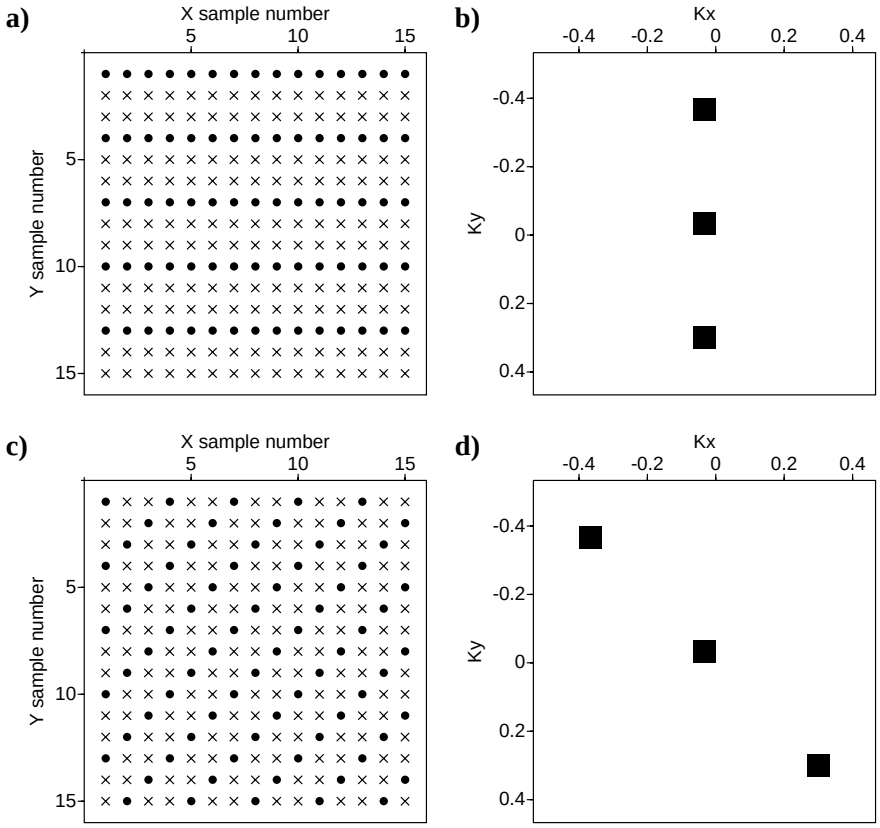

Figure 3.8: a) and c) Two different 2D regular sampling function with decimation factor of 3. b) and d) are the scaling functions of (a) and (c), respectively.

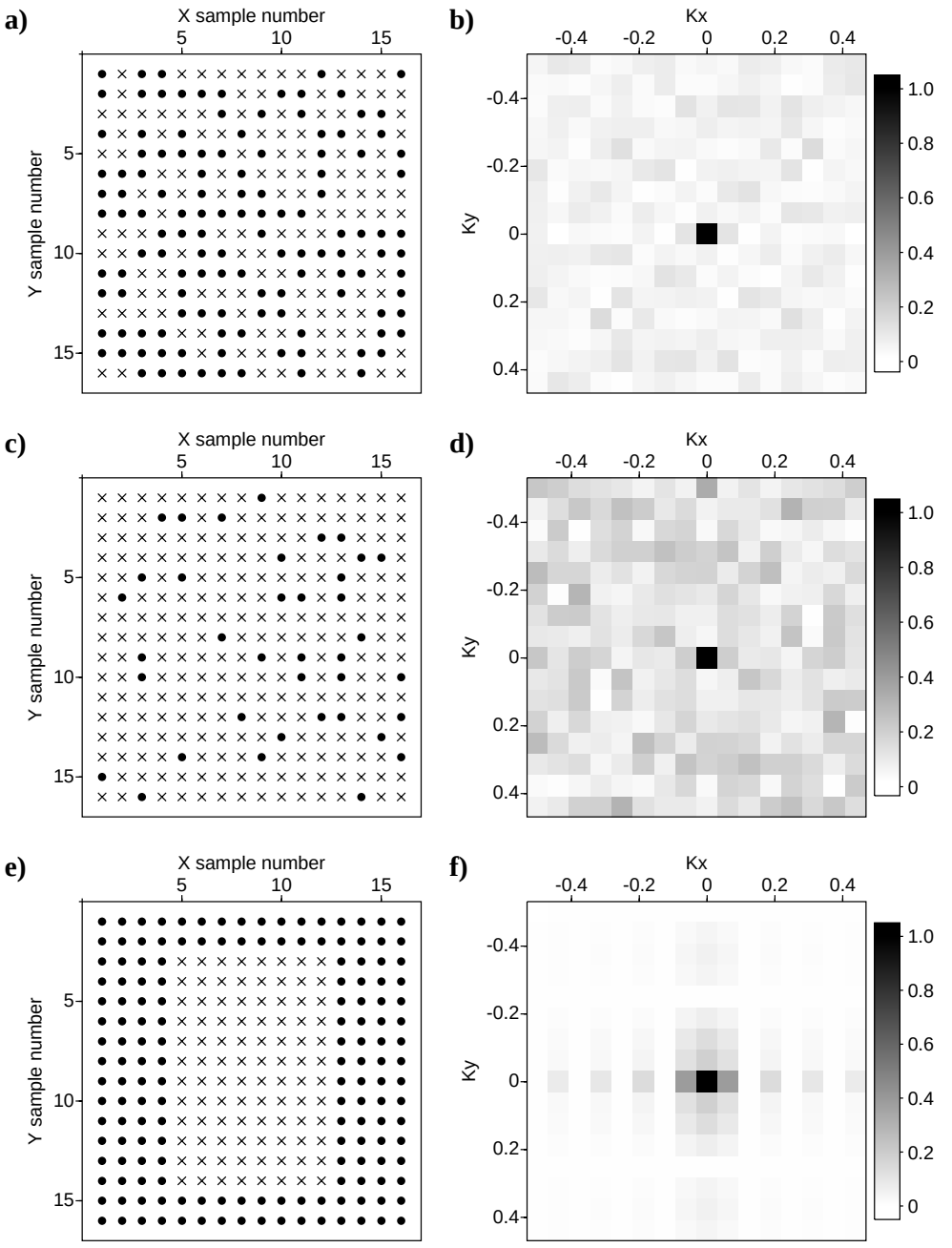

Figure 3.9: a) and c) Two 2D random sampling Sampling operators with 50% and 85% of missing samples, respectively. e) 2D gap sampling operator. b), d), and f) are 2D scaling functions of (a), (c), and (e), respectively.

parameters to data. For Fourier reconstruction purposes, $\mathbf{G}$ represents inverse Fourier transform operation. In order to obtain the least squares solution to equation (3.8) we build an objective function (Sacchi et al., 1998; Tarantola, 2005; Zwartjes, 2005):

$$J = \|\mathbf{C}_n^{-\frac{1}{2}}(\mathbf{d} - \mathbf{G}\mathbf{m})\|_2^2 + \|\mathbf{C}_m^{-\frac{1}{2}}(\mathbf{m} - \mathbf{m}_0)\|_2^2 \tag{3.9}$$

where, $\mathbf{C}_n^{-1}$ and $\mathbf{C}_m^{-1}$ are noise covariance matrix and a priori model covariance matrix, respectively. $\mathbf{C}_n^{-1}$ contains any information regarding measurement noise and possible shortcomings in the governing theory. Meanwhile, $\mathbf{C}_m^{-1}$ contains any known information about the model parameters and $\mathbf{m}_0$ is the initial solution for model parameters. Taking derivatives of expression (3.9) with respect to $\mathbf{m}$ and setting them to zero, the solution for a linear system of equations can be obtained by

$$\hat{\mathbf{m}} = (\mathbf{G}^H\mathbf{C}_n^{-1}\mathbf{G} + \mathbf{C}_m^{-1})^{-1}(\mathbf{G}^H\mathbf{C}_n^{-1}\mathbf{d} - \mathbf{C}_m^{-1}\mathbf{m}_0) \tag{3.10}$$

3.3.2 Minimum Norm Solution

To further simplify the solution (3.10) for Fourier reconstruction, one can represent the noise covariance matrix for the irregular sampling case as $\mathbf{C}_n^{-1} = \frac{1}{c^2}\mathbf{W}$, where, $W_{nn} = \Delta x_n$, $\sum \Delta x_n = 2\pi/\Delta k$ and c^2 is a constant (Sacchi and UIrych, 1996). Here, $\Delta x_n = (x_{n+1} - x_{n-1})/2$ where x_n indicates the location of nth sample of data and Δk is the wave-number sampling interval. By assuming the same variance for all model parameters, i.e. $\mathbf{C}_m^{-1} = \sigma_m^2\mathbf{I}$, and $\mathbf{m}_0 = \mathbf{0}$ we obtain a solution known as minimum norm solution (Duijndam et al., 1999)

$$\hat{\mathbf{m}} = (\mathbf{G}^H\mathbf{W}\mathbf{G} + \frac{c^2}{\sigma_m^2}\mathbf{I})^{-1}\mathbf{G}^H\mathbf{W}\mathbf{d}. \tag{3.11}$$

When the noise variance is unknown we simplify the problem by considering $\mathbf{W} = \mathbf{I}$ and denote $\lambda = \frac{c^2}{\sigma_m^2}$, then Equation (3.11) reduces to the well-known least squares solution form

$$\hat{\mathbf{m}} = (\mathbf{G}^H\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^H\mathbf{d}. \tag{3.12}$$

The scalar λ is the trade-off parameter. If $\lambda = 0$ then the solution is completely found based on fitting the data. The higher the value of λ, the more the solution honors the initial model (which is zero) and avoids fitting the data.

3.3.3 Sparse Solution

In order to impose spareness constraints on the solution of the least squares problem, one can build the following objective function (Zwartjes and Gisolf, 2006)

$$J = \frac{1}{c^2}\|\mathbf{W}^{\frac{1}{2}}(\mathbf{d} - \mathbf{G}\mathbf{m})\|_2^2 + \rho(\mathbf{m}), \tag{3.13}$$

where, $\rho(\mathbf{m})$ represents a penalty function. The solution can be represented as

$$\hat{\mathbf{m}} = (\mathbf{G}^H\mathbf{W}\mathbf{G} + c^2\mathbf{C}_m^{-1})^{-1}\mathbf{G}^H\mathbf{W}\mathbf{d} \tag{3.14}$$

where, $C_{m,ii}^{-1} = \frac{1}{m_i}\frac{\partial\rho(m_i)}{\partial m_i}$. By properly choosing the penalty function $\rho(\mathbf{m})$, one can obtain the desired sparse solution. A special choice for the penalty function is

$$\rho(m_i) = \frac{1}{2(1-a)}(m_i m_i^* + \sigma_m^2)^{1-a},$$

which leads to the following model covariance matrix

$$C_{m,ii}^{-1} = \frac{1}{m_i}\frac{\partial\rho(m_i)}{\partial m_i} = (m_i m_i^* + \sigma_m^2)^{-a}. \tag{3.15}$$

Setting the parameter a to the values $1/2, 1$ and 2 is equal to imposing the $\ell_{1/2}$, the Cauchy and the Geman-McClure constraint, respectively (Zwartjes and Gisolf, 2006). In order to investigate the mechanism of applying sparseness using these norms, we analyze the Cauchy norm with penalty function (Sacchi and UIrych, 1996)

$$\rho(x) = \frac{1}{2}\ln(1 + x^2).$$

Notice that the derivative of the above expression leads to expression (3.15) with $a = 1$. Therefore, we have

$$c^2\mathbf{C}_m^{-1} = c^2 C_{m,ii}^{-1} = \frac{c^2}{m_i^* m_i + \sigma_m^2} = \lambda\frac{1}{1 + \frac{m_i^* m_i}{\sigma_m^2}}. \tag{3.16}$$

Comparing expressions (3.16), (3.14) and (3.12), it can be concluded that the difference between Minimum Norm solution and sparse solution is in the weight function $\phi = \frac{1}{1+\frac{m_i^* m_i}{\sigma_m^2}}$.

The sparseness constraint can be interpreted as follows

- for the case $\sigma_m^2 \leq m_i^* m_i$, ϕ has a value in the range $(0, \frac{1}{2}]$ and therefore little to no damping is given to these parameters and they are determined by fitting the available data.

- for the case $\sigma_m^2 \geq m_i^* m_i$, ϕ has a value in the range $[\frac{1}{2}, 1)$ and therefore more damping is given to these parameters and more important becomes the apriori model. Since in our case the apriori model was zero, these components become zero.

Overall it can be concluded that by using the Cauchy norm as a penalty function, the small amplitude samples in the spectrum are eliminated while the large ones are intensified. Another alternative to obtain a sparse Fourier solution is to use a weighted function derived from the spectrum of data in order to magnify the high amplitudes and suppress the low ones. By iteratively repeating this routine inside a least squares optimization scheme, one can reach the sparse solution. This is explained in detail in the next section.

3.4 Minimum Weighted Norm Interpolation (MWNI)

Interpolation of band-limited data with missing samples can be summarized in the following inversion scheme (Liu and Sacchi, 2004)

$$\text{Minimize} \quad \|\mathbf{x}\|_{\mathcal{W}}^2 \qquad \text{Subject} \quad \textbf{to} \quad \mathbf{G}\mathbf{x} = \mathbf{y}, \tag{3.17}$$

where $\|.\|_{\mathcal{W}}^2$ indicates a specific weighted norm and $\mathbf{G}$ is the sampling matrix, which maps desired data samples $\mathbf{x}$ to available samples $\mathbf{y}$. Its transpose, $\mathbf{G}^T$, fills the position of missing samples with zeros. A regularization norm can be selected in the wave-number domain as follows

$$\|\mathbf{x}\|_{\mathcal{W}}^2 = \sum_{k \in \mathcal{K}} \frac{X_k^* X_k}{W_k^2}, \tag{3.18}$$

where X_k indicates the coefficients of the Fourier transform of the vector of spatial data $\mathbf{x}$. The values of W_k determine the type of interpolation. For band-limited Fourier reconstruction of data, a diagonal matrix is built by

$$\Upsilon_k = \begin{cases} W_k^2 & k \in \mathcal{K}, \\ 0 & k \notin \mathcal{K}, \end{cases} \tag{3.19}$$

where $\mathcal{K}$ indicates the region of support of the Fourier transform. The pseudoinverse of $\boldsymbol{\Upsilon}$ is defined by

$$\Upsilon_k^\dagger = \begin{cases} W_k^{-2} & k \in \mathcal{K}, \\ 0 & k \notin \mathcal{K}. \end{cases} \tag{3.20}$$

For band-limited minimum norm interpolation, the values of W_k are equal to one, while for the MWNI method, their values must be iteratively updated to find an optimal reconstruc-

tion. The minimizer of the cost function (3.17) is given by

$$\hat{\mathbf{x}} = \mathbf{F}^H \boldsymbol{\Upsilon} \mathbf{F} \mathbf{G}^T (\mathbf{G} \mathbf{F}^H \boldsymbol{\Upsilon} \mathbf{F} \mathbf{G}^T + \alpha \mathbf{I})^{-1} \mathbf{y}, \tag{3.21}$$

where $\mathbf{F}$ is the Fourier matrix, α is the trade-off parameter, $\mathbf{I}$ is the identity matrix, T and H stand for transpose and Hermitian transpose operators, respectively. For further details see Liu (2004) and Liu and Sacchi (2004). Equation (3.21) can be efficiently solved using Conjugate Gradient (CG) method. Algorithms 3.1 - 3.3 demonstrates the CG algorithm used for solving the MWNI in this thesis. The sign $\odot$ represents component by component multiplication of vectors and the sign $\circledast$ indicates convolution. Notice that lower-case bold variables represent the data vectors in time (space) and upper-case bold variable indicate the data vectors in Fourier domain.

The MWNI algorithm uses a data-driven weight function to obtain a sparse solution for the reconstruction problem. For each update of the weighting function, the coefficients with high values are enhanced by gaining more weight. The final outcome of MWNI algorithm will be a sparse solution with a smooth structure. This avoids the generation of unphysical solutions that are too sparse. In this dissertation I will consider the MWNI algorithm as the method of choice for my examples when working with Fourier reconstruction. It should be kept in mind that MWNI is different from sparse Fourier reconstruction methods that use the Cauchy or ℓ_1 norms (Zwartjes, 2005; Hennenfent and Herrmann, 2006).

3.5 Examples

3.5.1 Synthetic 1D examples

In order to examine the performance of sparse reconstruction methods with various sampling operators, a synthetic 1D signal with 120 samples composed of two harmonics was created (Figure 3.10a). For 1D examples, left panels will show the spatial domain (or time domain) and the right panels will represent their corresponding spectra. To proceed with the first example, 80% of the data was randomly eliminated and the missing samples (Figure 3.10c) were reconstructed using MWNI method. Figure 3.10e shows the result of the reconstruction. Since the available samples were picked by a random nonuniform sampling operator the reconstruction was successful. This is due to the fact that the scaling function for the nonuniform sampling case has only one high amplitude impulse (Figure 3.2). Hence, its convolution with the spectrum of the original data (which is sparse as well) will create a sparse signal (with peaks at the same location of the peaks of the original spectrum) suitable to be reconstructed using MWNI or sparse Fourier inversion methods. Figures 3.10b, 3.10d, and 3.10f show the Fourier panels of Figures 3.10a, 3.10c, and 3.10e, respectively.

Algorithm 3.1: Minimum Weighted Norm Interpolation

Input:
Total number of data samples:nt
Indices of the available samples: $\mathbf{h}_{ns\times 1}$
Values of the available samples: $\mathbf{y}_{ns\times 1}$
Hanning window: $\mathbf{H}_{nh\times 1}$
Output:
Final reconstructed data samples: $\hat{\mathbf{x}}$

begin
  Initiate band-limiting function : $\boldsymbol{\Upsilon}_{nt\times 1} = \begin{cases} 1 & \textit{Desired wavenumbers} \\ 0 & \textit{Elsewhere} \end{cases}$
  for $i = 1$ **to** $niter_1$ **do**
    $\mathbf{X} = \mathbf{0}_{nt\times 1}$
    $\mathbf{s} = \mathbf{y}$
    Forward Operator (Algorithm 3.2)
    $\mathbf{P} = \mathbf{R}$
    $\epsilon_1 = \mathbf{R}^H\mathbf{R}$
    $\delta_1 = \mathbf{s}^H\mathbf{s}$
    for $j = 1$ **to** $niter_2$ **do**
      Adjoint Operator (Algorithm 3.3)
      $\alpha = \epsilon_1/(\mathbf{q}^H\mathbf{q})$
      $\mathbf{X} = \mathbf{X} + \alpha\mathbf{P}$
      $\mathbf{s} = \mathbf{s} - \alpha\mathbf{q}$
      Forward Operator (Algorithm 3.2)
      $\epsilon_2 = \epsilon_1$
      $\epsilon_1 = \mathbf{R}^H\mathbf{R}$
      $\beta = \epsilon_1/\epsilon_2$
      $\mathbf{P} = \mathbf{R} + \beta\mathbf{P}$
      $\delta_2 = \mathbf{s}^H\mathbf{s}$
      if $[(\delta_1 - \delta_2) < tol_1]$ ***or*** $[\delta_2 < tol_2]$ **then**
        Break
      end
      $\delta_1 = \delta_2$
    end
    Update and smooth band-limiting function: $\boldsymbol{\Upsilon} = abs(\mathbf{X}) \circledast \mathbf{H}$
  end
  $\hat{\mathbf{x}} = IFFT(\mathbf{X})$
end

Algorithm 3.2: Forward operator of MWNI

begin
  $\mathbf{t} = \mathbf{0}_{nt\times 1}$
  $\mathbf{t}(\mathbf{h}) = \mathbf{s}$
  $\mathbf{T} = FFT(\mathbf{t})$
  $\mathbf{R} = \boldsymbol{\Upsilon} \odot \mathbf{T}$
end

Algorithm 3.3: Adjoint operator of MWNI

begin
  $\mathbf{T} = \boldsymbol{\Upsilon} \odot \mathbf{P}$
  $\mathbf{t} = IFFT(\mathbf{T})$
  $\mathbf{q}_{ns\times 1} = \mathbf{t}(\mathbf{h})$
end

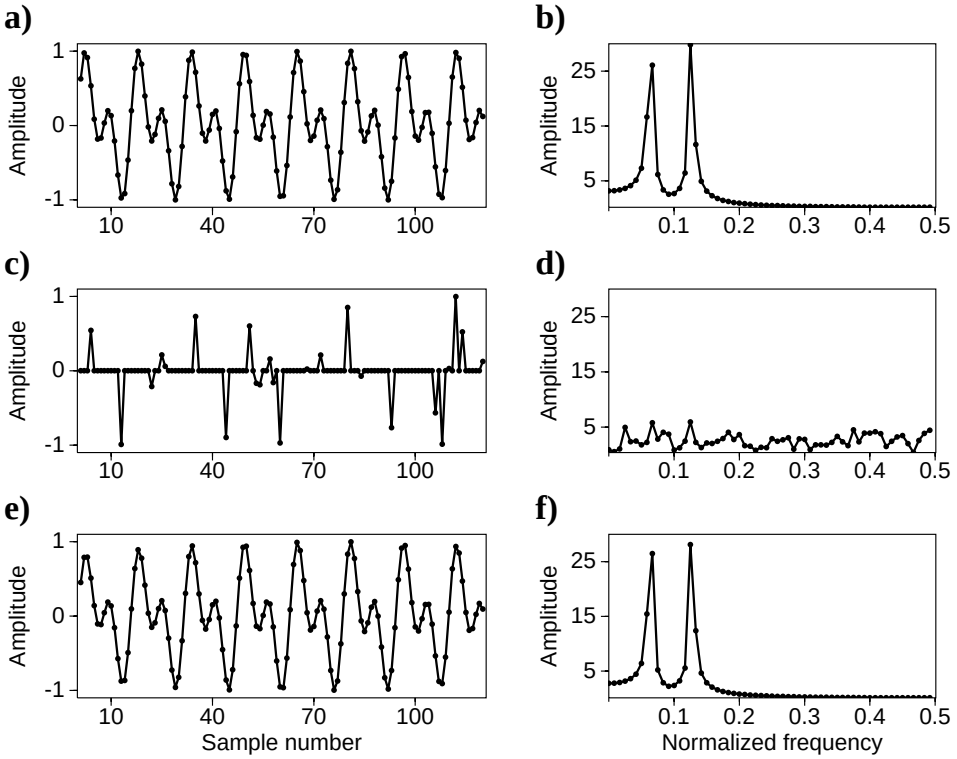

Figure 3.10: a) Original data. c) Data with 80% randomly missing samples. e) Reconstructed data via MWNI. b), d), and f) are the Fourier panels of (a), (c), and (e), respectively.

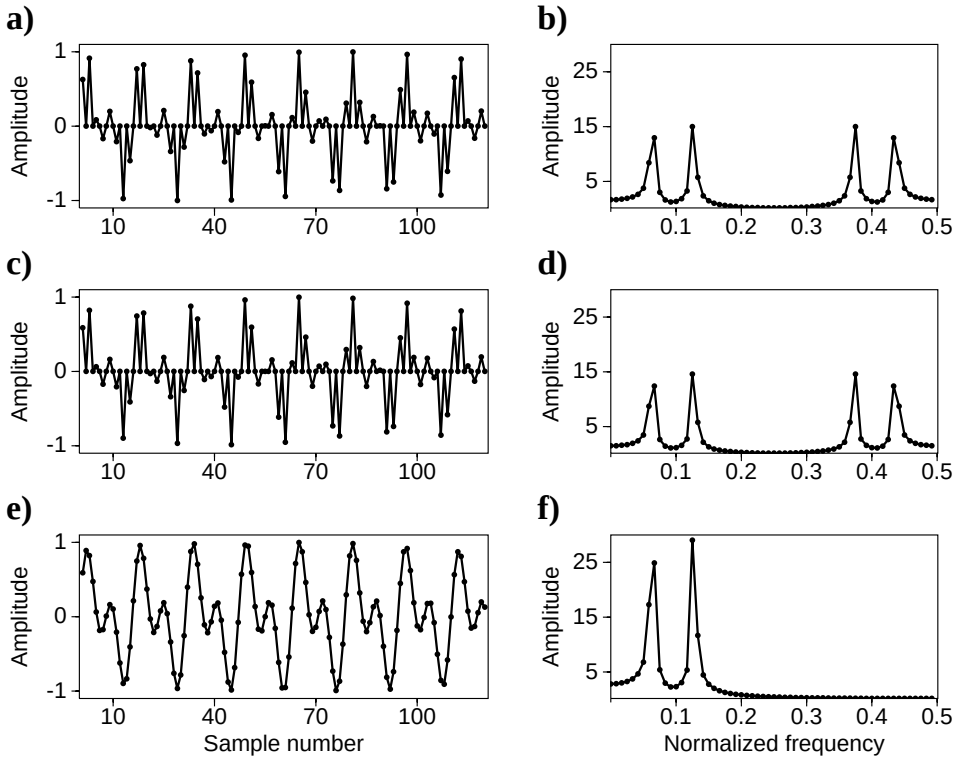

Figure 3.11: a) Regularly decimated data (decimation factor of 2). c) Reconstructed data via MWNI. e) Reconstructed data using band-limited MWNI in the normalized frequency interval of [0,0.15]. b), d), and f) are the Fourier panels of (a), (c), and (e), respectively.

For the next example (Figure 3.11a), the signal was decimated by a decimation factor of 2 (every other sample). Due to the regularity of the samples and considering the shape of the scaling function (Figure 3.1), the MWNI method was not able to reconstruct the missing samples (Figure 3.11c). While MWNI fails to recover the missing data, one can utilize band-limitation in the Fourier domain for reconstruction purposes. Figure 3.11e shows the successful reconstruction of the data using the band-limited MWNI method. The band-limiting function in the Fourier domain is designed to eliminate any event out of the normalized frequencies $[0, 0.15]$. The interval is chosen based on the apriori information regarding the spectrum of the original data. Figures 3.11b, 3.11d, and 3.11f show the Fourier panels of Figures 3.11a, 3.11c, and 3.11e, respectively.

A sampling operator can be a combination of regular and random sampling functions (Figure 3.4). This means that first, a discrete signal is decimated using a regular sampling operator and then the resultant sampled function is sampled using a random sampling operator.

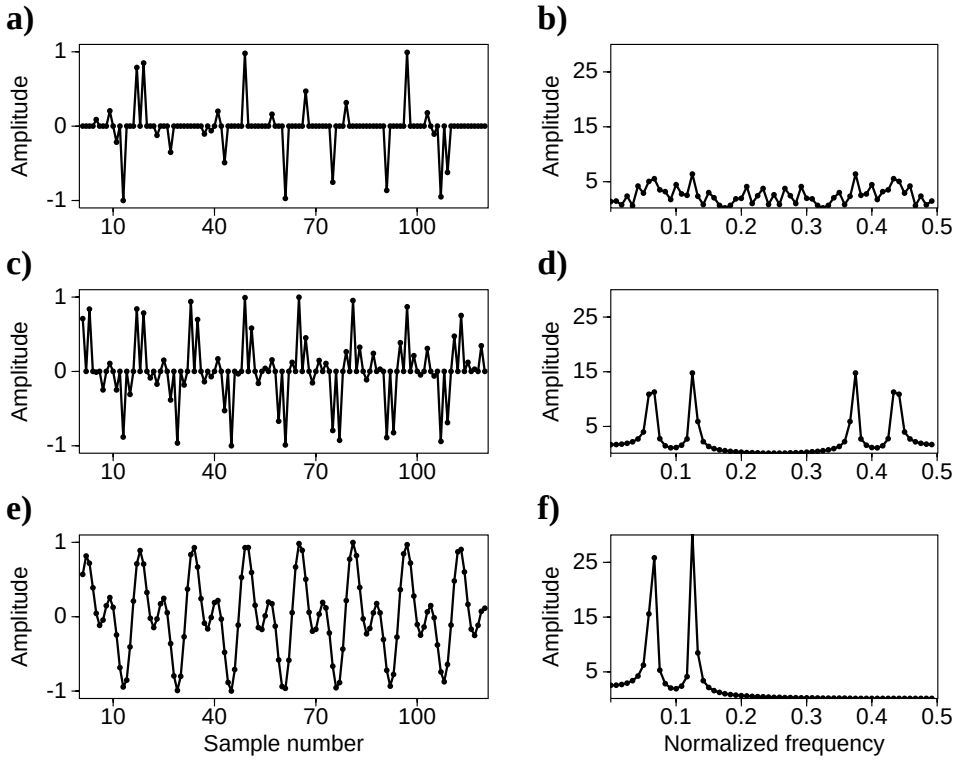

Figure 3.12: a) Random sampling from a regularly decimated data (decimation factor of 2). c) Reconstructed data via MWNI. e) Reconstructed data using band-limited MWNI in the normalized frequency interval of [0,0.15]. b), d), and f) are the Fourier panels of (a), (c), and (e), respectively.

Figure 3.12a shows an example where the available samples have been randomly picked from an already decimated original signal by a factor of 2 (Figure 3.11a). Figure 3.12c shows the reconstructed data using the MWNI method. Since the random sampling operator is applied on the already decimated signal, MWNI was able to reconstruct the decimated signal, not the original one. Figure 3.12e shows the reconstructed data using MWNI when band-limitation is included. Notice that we have simultaneously utilized band-limitation and the MWNI constraint to remove the effects of both regular and random sampling operators. Figures 3.12b, 3.12d, and 3.12f show the Fourier panels of Figures 3.12a, 3.12c, and 3.12e, respectively.

Figure 3.13a shows an example where the available samples were randomly selected from an already decimated signal by a factor of 3. This means that the scaling function has three high amplitude impulses. The MWNI method was only able to reconstruct the decimated

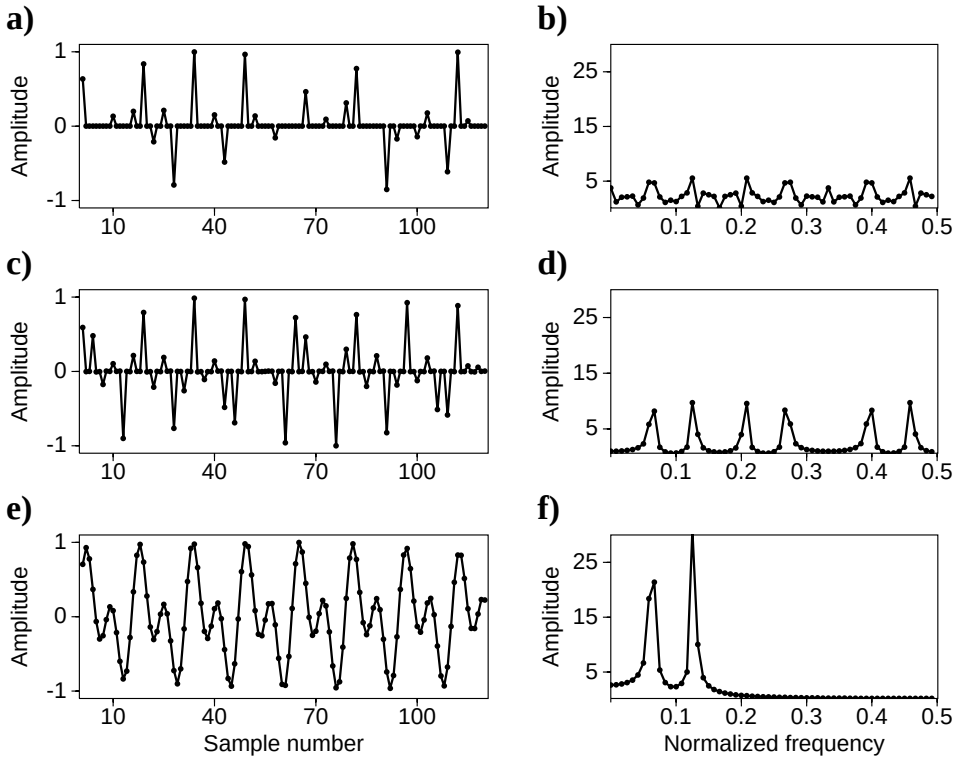

Figure 3.13: a) Random sampling from a regularly decimated data (decimation factor of 3). c) Reconstructed data via MWNI. e) Reconstructed data using band-limited MWNI in the normalized frequency interval of [0,0.15]. b), d), and f) are the Fourier panels of (a), (c), and (e), respectively.

signal by a factor of 3 (Figure 3.13c) and was not able to recover the missing samples from the already decimated signal. Conversely, MWNI with band-limitation has successfully reconstructed the original data (Figure 3.13e). Figures 3.13b, 3.13d, and 3.13f show the Fourier panels of Figures 3.13a, 3.13c, and 3.13e, respectively.

While MWNI with the addition of band-limitation can eliminate high amplitude artifacts caused by regular sampling operators, it will fail to do so if the artifacts are mixed with the original spectrum of the data. Figure 3.14a show a regularly decimated signal with decimation factor of 6. Figure 3.14b shows the Fourier panel of Figure 3.14a. It is evident that parts of the artificial events introduced by the regular sampling operator are interfering with the original spectrum of the data. Figure 3.14c and 3.14d show the reconstructed data using band-limited MWNI and its Fourier panel, respectively. The reconstructed data are different from the original data due to the presence of high amplitude artifacts in the band

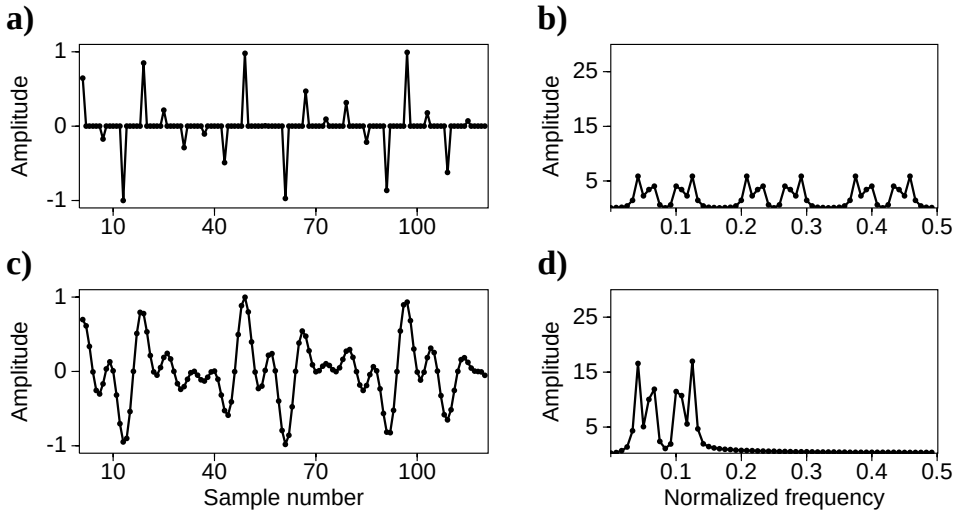

Figure 3.14: a) Regularly decimated data (decimation factor of 6). c) Reconstructed data via MWNI. e) Reconstructed data using band-limited MWNI in the normalized frequency interval of [0,0.15]. b), d), and f) are the Fourier panels of (a), (c), and (e), respectively.

selected for reconstruction.

There is a common misinterpretation pertaining random sampling, which needs to be clarified. One might think that the shortest distance between consecutive samples (or average sampling rate) determines the success of a reconstruction method based on sparsity or MWNI. Figure 3.15a shows an example of randomly picked samples in which the shortest distance between consecutive samples was forced to be more than 4 samples. The reconstructed data using the MWNI method (Figure 3.15c) shows that the signal was reconstructed successfully even though we did not have samples with less than 4 samples apart. This reflects the fact that it is the main underlying grid of the sampling operator that determines the success of the sparse Fourier reconstruction methods. Figures 3.15b and 3.15d show the Fourier panels of Figures 3.15a and 3.15c, respectively.

It is important to mention that for random sampling even relatively high amplitude artifacts can be eliminated using the MWNI method. This is due to the fact that the MWNI algorithm is a non-linear interpolation method which iteratively updates its weighting function in the frequency domain by comparing the latest result of the reconstruction to the available samples. Hence, if there were high amplitude artifacts in the spectrum that do not match the available samples, they are eliminated by iterative updating. Unfortunately, artifacts in the spectrum due to regular decimation in the spatial domain will completely match the available samples and therefore, non-linear updating can not eliminate them.

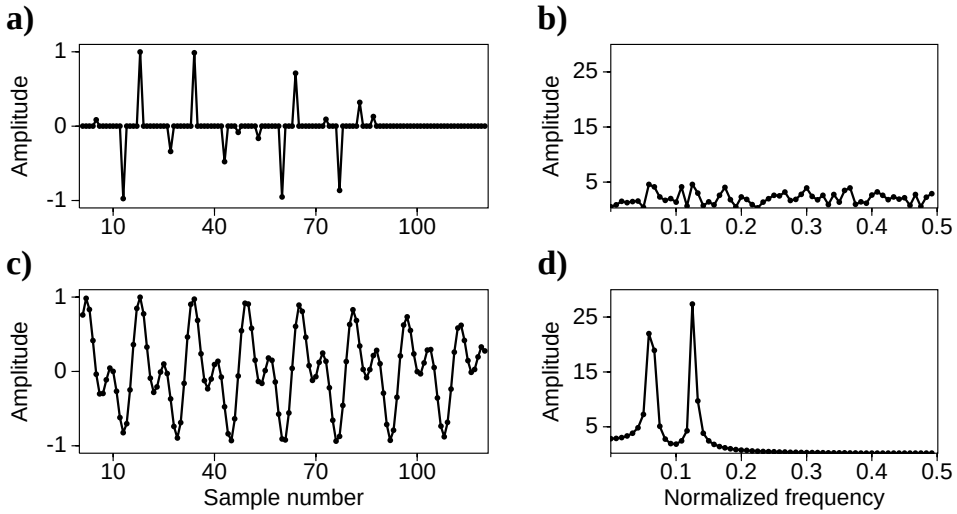

Figure 3.15: a) Randomly sampled data with the restriction of a minimum 4 samples distance between consecutive available samples. c) Reconstructed data via MWNI. b) and d) are the Fourier panels of (a) and (c), respectively.

Figure 3.16a shows an example with a gap inside the data. The reconstructed data using MWNI (Figure 3.16c) shows a successful recovery of the samples in the gap. It is clear that the artifacts caused by the gap in the frequency domain (3.16b) are all in the vicinity of the main spectrum of the original signal (3.10b). This was expected since the scaling function of the gap sampling operator (Figure 3.5) has high value impulses only in the vicinity of the main impulse of the original spectrum. However, as the Fourier panel of the reconstructed data (Figure 3.16d) shows, the MWNI method was capable of eliminating the artifacts caused by a gap inside the regularly sampled grid of data.

Figure 3.17 shows the relationship between the percentage of available samples and the reconstruction error for MWNI when random sampling is adopted. For each percentage of missing samples, 20 different random sampling realizations were performed and the mean of reconstruction error and its standard deviation were plotted. As the number of available samples decreases, the reconstruction error and its variance increases.

3.5.2 Synthetic 2D examples

For 2D data reconstruction, the Fourier reconstruction is separately applied for each frequency in the *f*-*x* domain. Figure 3.18a shows an original synthetic 2D section composed of three linear events. The original data are severely aliased for the normalized frequencies

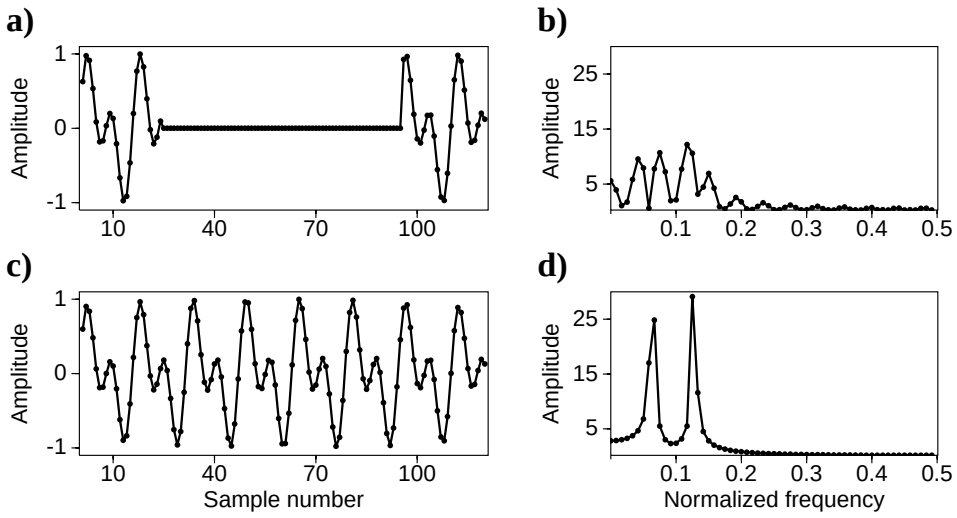

Figure 3.16: a) Data with erasures due to the presence of a gap. c) Reconstructed data via MWNI. b) and d) are the Fourier panels of (a) and (c), respectively.

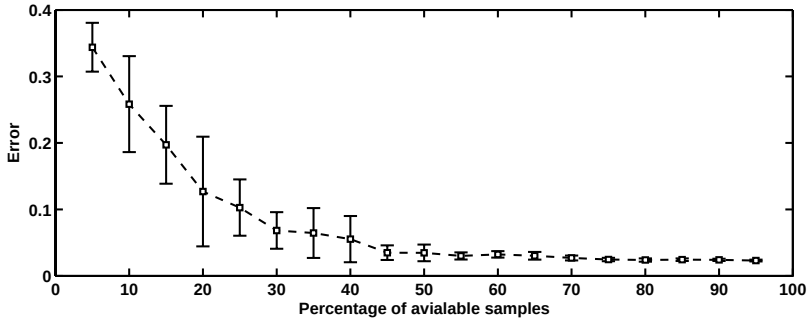

Figure 3.17: Reconstruction error vs percentage of available samples for 1D MWNI of randomly sampled data. For each case 20 different random realization carried out. The variance is shown by the error bars.

higher than 0.12. More than 60 percent of traces are randomly eliminated in the section of missing traces (Figure 3.18b). Figure 3.18c shows the reconstructed data using the MWNI method. Figures 3.18d, 3.18e and 3.18f show the *f-k* domain of Figures 3.18a, 3.18b and 3.18c, respectively. The random sampling in the spatial domain creates low amplitude artifacts around the spectrum of the original data. This is the behavior expected from the random sampling operator as shown in Figure 3.2. The MWNI method is capable of eliminating all the small amplitude artifacts (Figure 3.18c). However, there are some isolated high amplitude artifacts left in the *f-k* domain of the reconstructed data. In the next chapter, I will discuss a methodology to eliminate these artifacts.

The same 2D synthetic section was also decimated by a factor of 2 (Figure 3.19a) in order to be reconstructed by the MWNI method. Since the sampling operator was regular, the MWNI method was not able to recover the missing traces (Figure 3.19b). Figures 3.19c and 3.19d show the *f-k* domain representation of the figures 3.19a and 3.19b, respectively. The regular sampling function creates a replica of the original spectrum. Since the original spectrum of data was aliased, the resulted replicated completely mixes with the original spectrum. Applying the MWNI method separately for each frequency can not discriminate between the original and the replicated spectrum. The replicated spectrum is well-separated in the low frequency portion of the data and a band-limiting (low-pass) function can be deployed in the wavenumber axis to eliminate them. However, for the high frequency portion of the data instead of low-pass functions one needs several band-pass functions. The basics for building the multi-band-pass functions will be discussed in the next Chapter.

Next, I investigate the gap inside a 2D synthetic example (Figure 3.20a). As it was expected and shown in Figure 3.20c, the gap causes low amplitude repetition of the spectrum in the vicinity of the original spectrum. Figures 3.20b and 3.20d show the reconstructed data using MWNI in the *t-x* and *f-k* domain, respectively. The results show that the MWNI method is a good candidate for reconstruction of a gap inside the data.

3.5.3 3D examples

Synthetic linear events

In order to examine the performance of MWNI for multidimensional data, a 3D cube of synthetic data was created. The original data are composed of three dipping planes which are aliased in both spatial directions. The data set contains 2400 traces in a spatial grid of 40×60 and slightly contaminated with random noise. Figure 3.21a shows a perspective view of the cube containing the synthetic data. The trajectory lines of the three synthetic events can be seen on the boundaries of the cube. In order to get an in-depth view of the

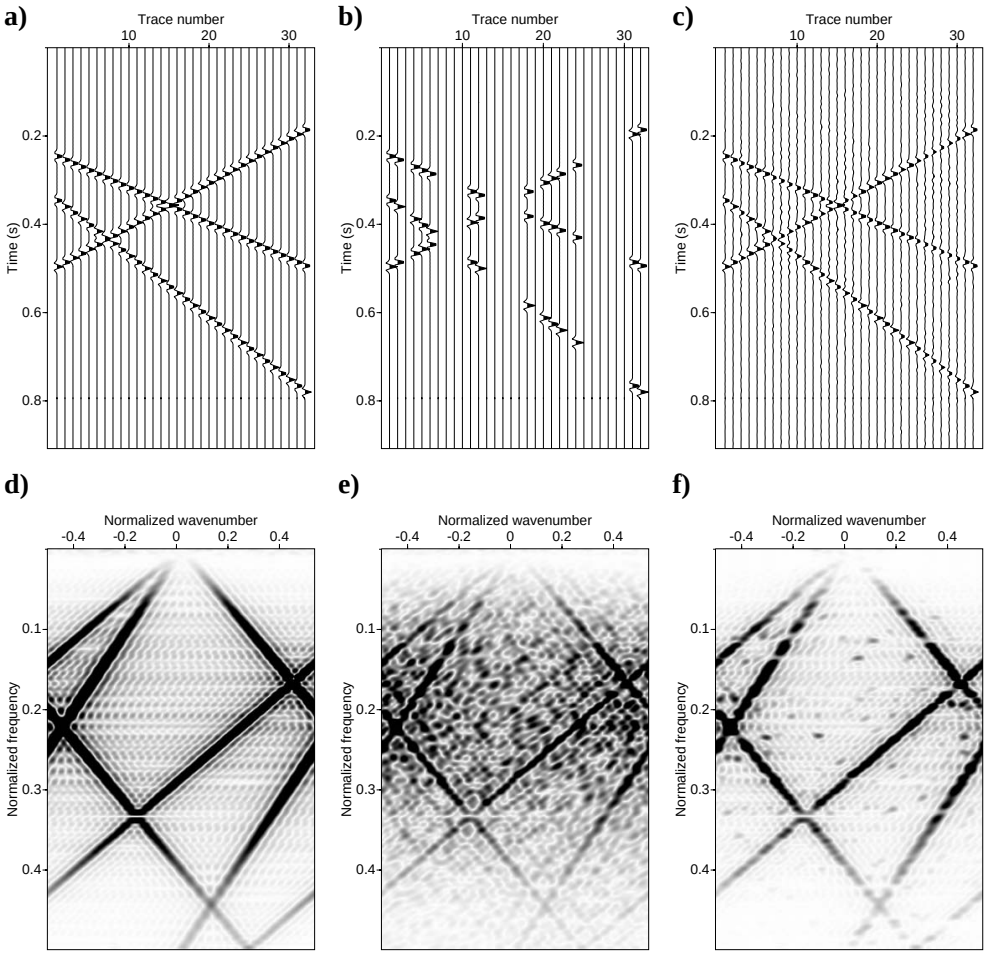

Figure 3.18: 2D data in the t-x domain. a) Original data. b) Randomly sampled data. c) Reconstructed data via MWNI. d), e), and f) are the Fourier panels of (a), (b), and (c), respectively.

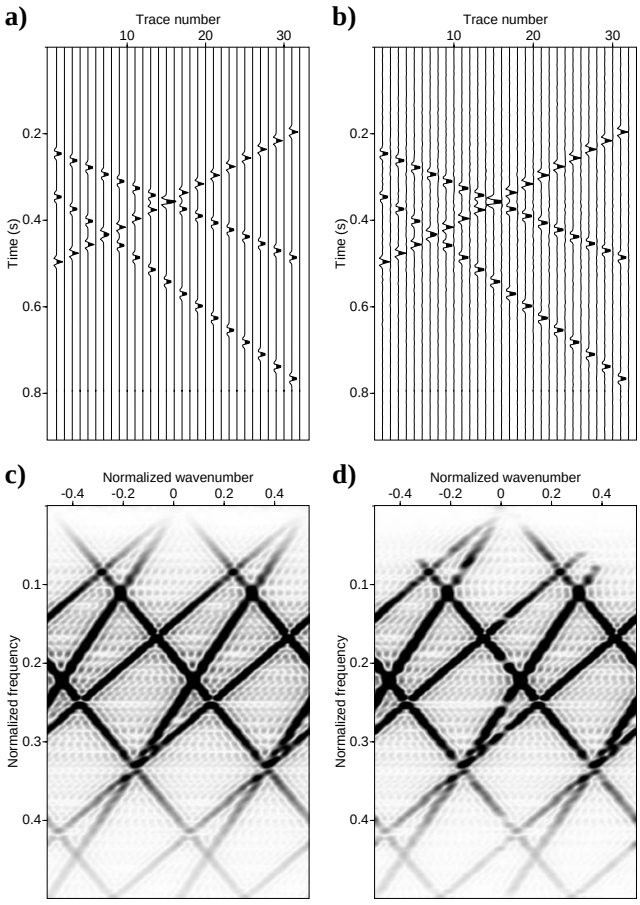

Figure 3.19: a) Regular decimation of data in Figure 3.19a. b) Reconstructed data via MWNI. c) and d) are the Fourier panels of (a) and (b), respectively.

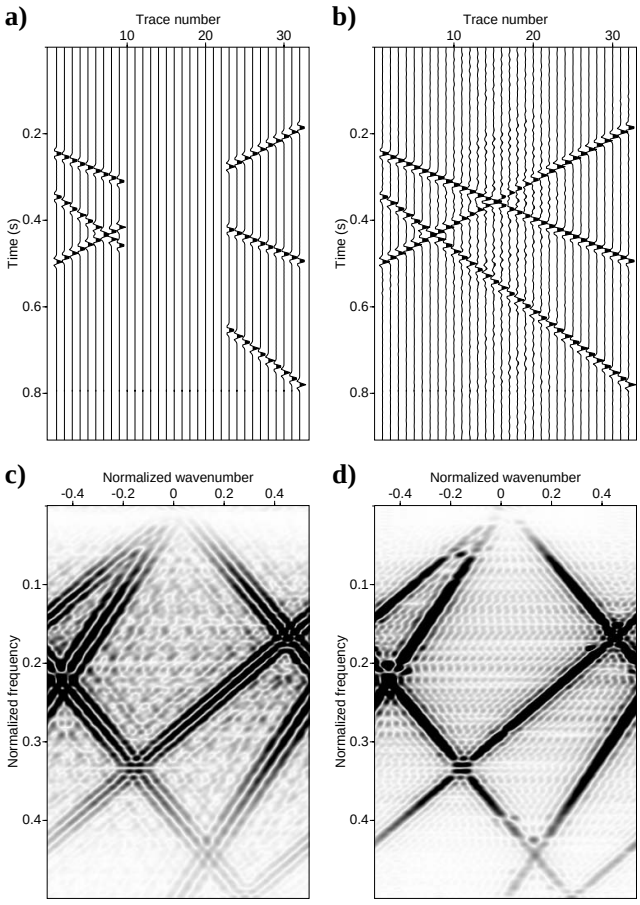

Figure 3.20: a) Data with gap inside the original data in Figure 3.19a. b) Reconstructed data via MWNI. c) and d) are the Fourier panels of (a) and (b), respectively.

data one can pick slices of data in different locations and project them to the sides of the cube. Figure 3.21b shows a cube of the original data where the top view is the time slice at 0.65 (s), the front view is the 21st slice in the Y direction, and the side view is 17th slice in the X direction. Figure 3.21c shows the cube with missing traces on which about 95% of traces have been eliminated randomly. Figure 3.21d shows the reconstructed data using the MWNI method. Despite the large number of missing traces, the reconstruction was successful. Figures 3.22a, 3.22b, and 3.22c show the *f-k* panels of the front views in Figures 3.21b, 3.21c, and 3.21d, respectively.

Figure 3.23a shows a cube with missing traces which was created from Figure 3.21b by first decimating every other slice in the X direction and then eliminating 50% of the remaining traces. Overall, in Figure 3.23a, about 75% of the original traces were eliminated. Figure 3.23b shows the reconstructed cube using the MWNI method. It is interesting to note that only every other slice in the X direction is successfully reconstructed. This was expected since the original decimation of slices in X direction results in a 2D scaling function similar to the one shown in Figure 3.7d. This means that the replicated artifacts in the X direction are as large as the amplitudes in the original spectrum and cannot be removed by sparse Fourier reconstruction or MWNI. Since the original data was aliased in both spatial directions we could not use band-limited MWNI to eliminate the artifacts. Figures 3.24a and 3.24b show the *f-k* panels of the front views in Figures 3.24a and 3.24b, respectively.

Synthetic curved events

It is interesting to examine the performance of Fourier reconstruction for non-linear (curved) events. Events with curvatures in the *t-x* domain do not have a simple or sparse representation in the frequency-wavenumber domain. Figure 3.25a shows a cube of synthetic 3D data which contains curved events. Figure 3.25b shows slice views of the original data at time 0.75 (s) (top view), X slice 17 (side view), and Y slice 21 (front view). Figure 3.26a shows the *f-k* panel of the front view in Figure 3.25b. The wrap-around energy on the frequency axis is an indication of aliased events. Figure 3.25c shows slice views of the data after randomly eliminating 60% of the traces. The *f-k* panel of the front view in Figure 3.25c is depicted in Figure 3.26b. Figure 3.25d shows the reconstructed data using the MWNI method. Figure 3.26c shows the *f-k* panel of the front view in Figure 3.25d. The existence of curved aliased events resulted in the failure of the MWNI method.

To continue with the analysis, I have created a synthetic cube of data composed of events with mild curvatures (Figure 3.27). Figure 3.27b shows slice views of the original data at time 0.75 (s) (top view), X slice 17 (side view), and Y slice 21 (front view). Next, 60% of the traces were eliminated to create the cube with missing traces (Figure 3.27c). The reconstructed data using the MWNI method is shown in Figure 3.27d. Figures 3.28a, 3.28b,

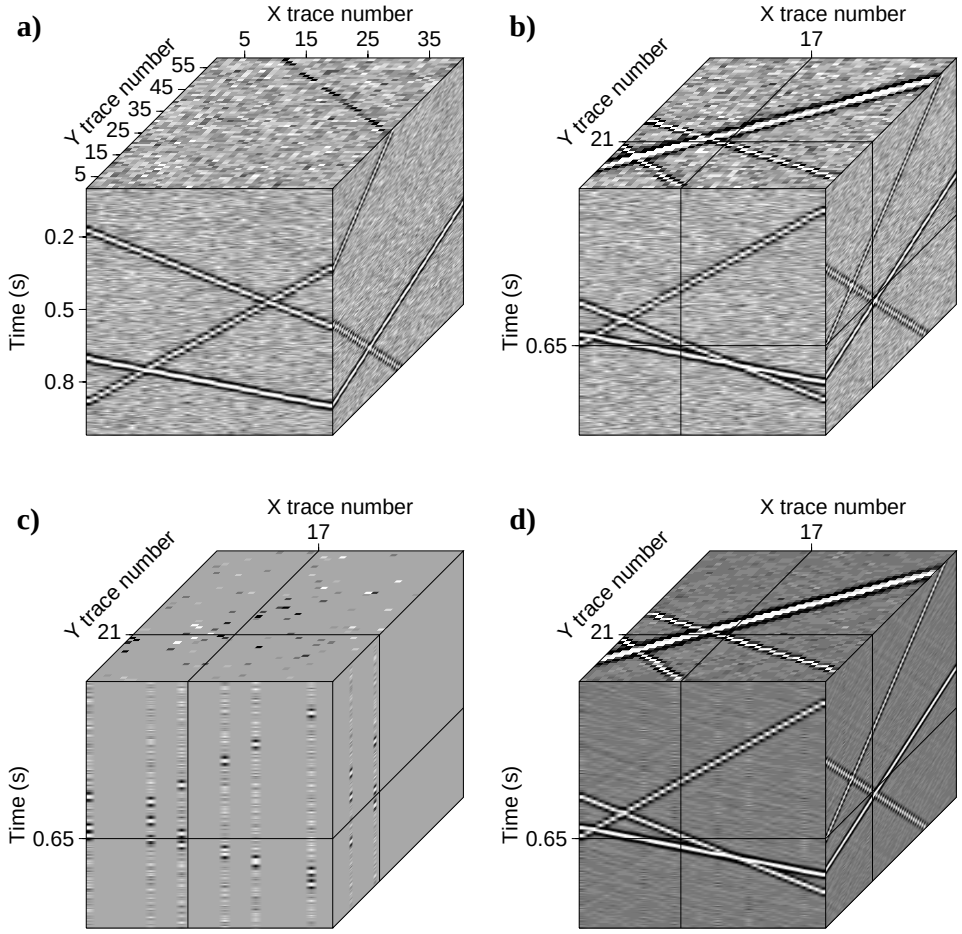

Figure 3.21: a) 3D synthetic cube of data. b) Slice views from inside the original data. c) Data with 95% randomly missing traces. d) Reconstructed data using sparse Fourier reconstruction.

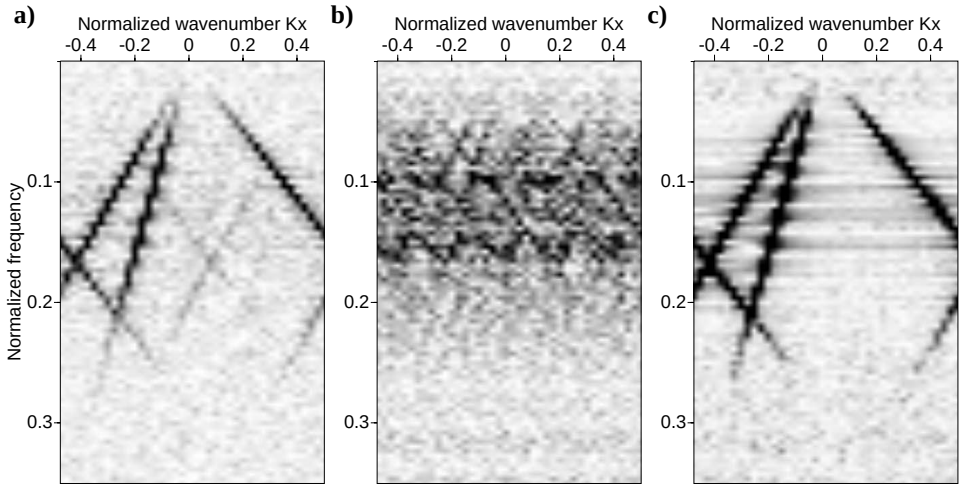

Figure 3.22: a), b), and c) are the *f-k* panels of the front views of the data in Figures 3.21b, 3.21c, and 3.21d.

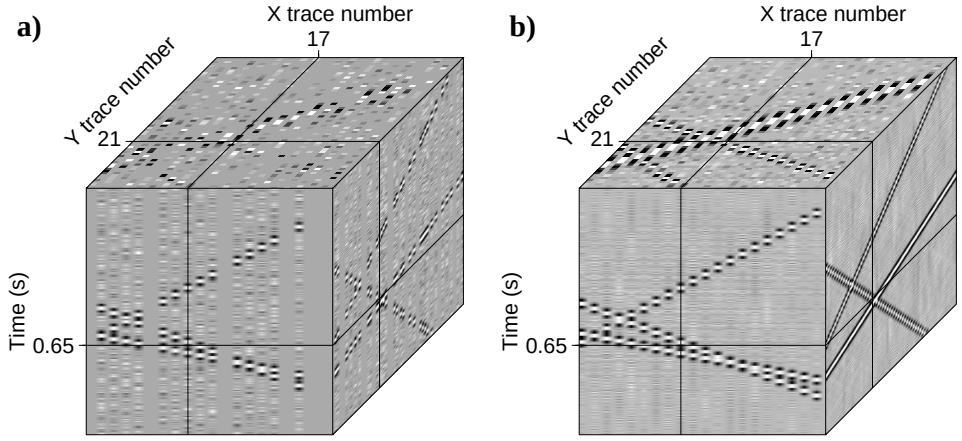

Figure 3.23: a) Cube of missing traces created from the original data shown in Figure 3.21b by first eliminating every other X slice and then randomly elimination 50% of the remaining traces. b) Reconstructed data using MWNI.

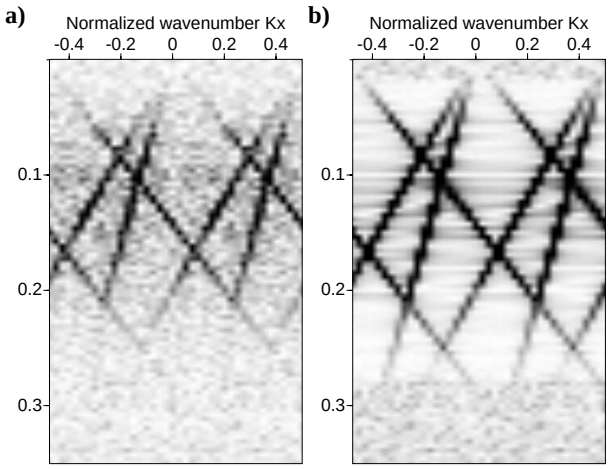

Figure 3.24: a) and b) are the *f-k* panels of the front views of the data in Figures 3.23a and 3.23b.

and 3.28c show the *f-k* panel of front views in Figures 3.27b, 3.27c, and 3.27d, respectively. Despite the presence of mild curvature in the original data, the MWNI method was able to recover the missing traces.

Figure 3.29a shows another synthetic cube of curved events which are aliased in the X direction and alias-free in the Y direction. Figure 3.29b shows slice views of the original data at time 0.35 (s) (top view), X slice 17 (side view), and Y slice 21 (front view). Figure 3.29c shows data after eliminating every other slice of data in the X direction. The sampling operator is equivalent to the one depicted in Figures 3.7c and 3.7d. Figure 3.29d shows the reconstructed data using band-limited MWNI by applying band-limitations on the normalized wavenumber in the Y direction ($k_y \in [-0.15, 0.15]$). The reconstruction was not successful because the sampling artifacts were inside the band-width used by the inversion. Figure 3.29e shows the cube of data with missing traces which has been sampled in a chessboard pattern. The sampling operator is equivalent to the one in Figure 3.7e. Figure 3.29f shows the reconstruction of the data in Figure 3.29e using band-limited MWNI in the bandwidth $k_y \in [-0.15, 0.15]$. For the chessboard pattern, the artifacts fall outside the chosen band-width and as a result the band-limited MWNI can recover the missing traces successfully. Figures 3.30a, 3.30b, and 3.30c show the *f-k* panel of front views in Figures 3.29b, 3.29e, and 3.29f, respectively.

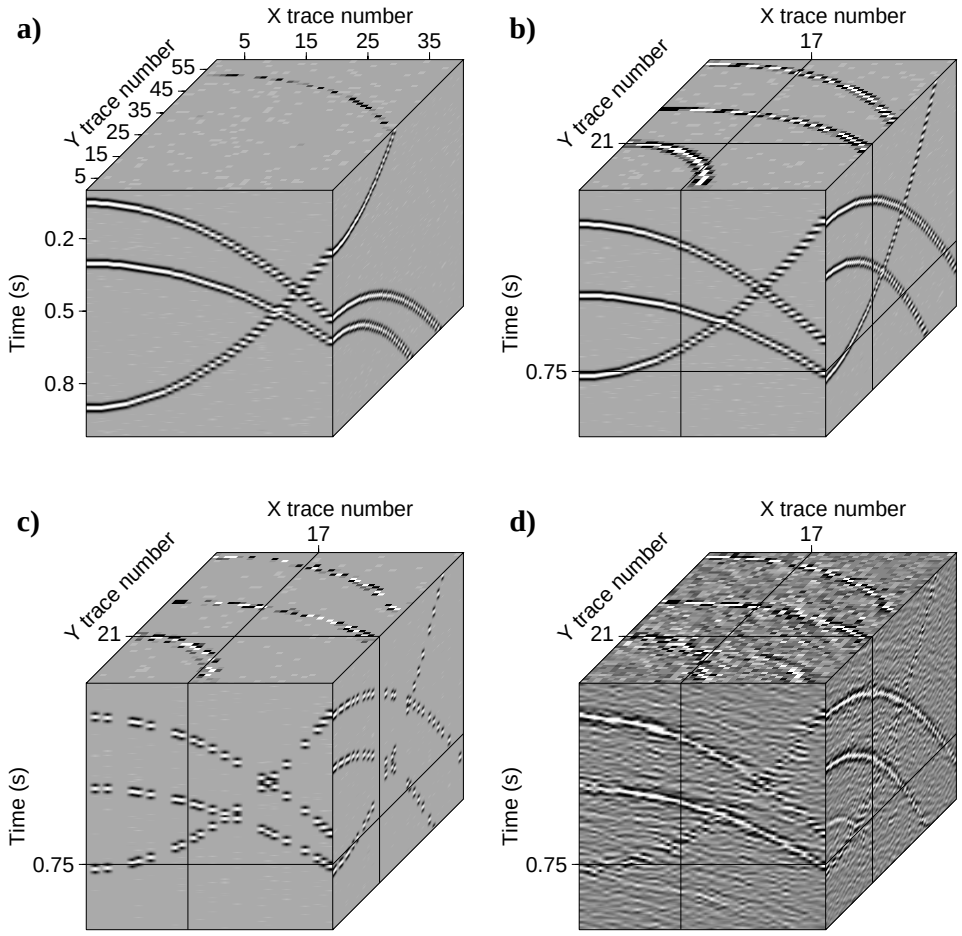

Figure 3.25: a) 3D synthetic cube of data with curved and aliased events. b) Slice view from inside the original data. c) Data with 60% randomly missing traces. d) Reconstructed data using MWNI.

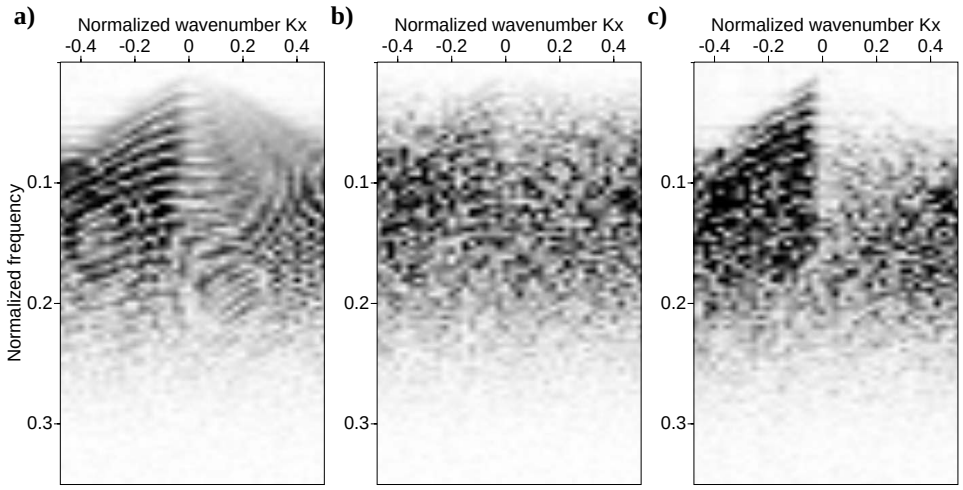

Figure 3.26: a), b), and c) are the *f-k* panels of the front views of the data in Figures 3.25b, 3.25c, and 3.25d.

Real data

A real data set from the Gulf of Mexico was selected to analyze the performance of MWNI under different sampling scenarios. Nineteen consecutive shots each with 91 traces, each trace having 900 time samples are windowed for this specific test. Figure 3.31a shows a perspective view of the cube containing the selected real data with trajectories of the events on the boundaries of the cube. Figure 3.31b shows slice views of Figure 3.31a at time 3.5 (s) (top view), X slice 46 (side view), and Y slice 10 (front view). Next 50% of the original traces were eliminated randomly to create a cube of missing traces (Figure 3.31c). Figure 3.31d shows the reconstructed data using the MWNI method. Figures 3.32a, 3.32b, and 3.32c show the *f-k* panels of the front views in Figures 3.32b, 3.32c, and 3.32d, respectively. Because of the presence of curved events there is some differences between the original data and the reconstructed data. By choosing a proper windowing strategy in the spatial and time directions, one can further reduce the differences between the original and reconstructed data. In Chapter 5 I will utilize a spatial windowing scheme for reconstruction of this real data set.

Figure 3.33a shows the cube of data from the Gulf of Mexico on which the original traces are decimated by a chessboard pattern. The 2D sampling operator for this example is similar to the one in Figure 3.7e. Figure 3.34a shows the *f-k* panel of the front view of Figure 3.33a. It is clear that regular sampling has created a replication of the spectrum of the original data. Also there is a prominent overlap between the original and replicated spectrum. Figure

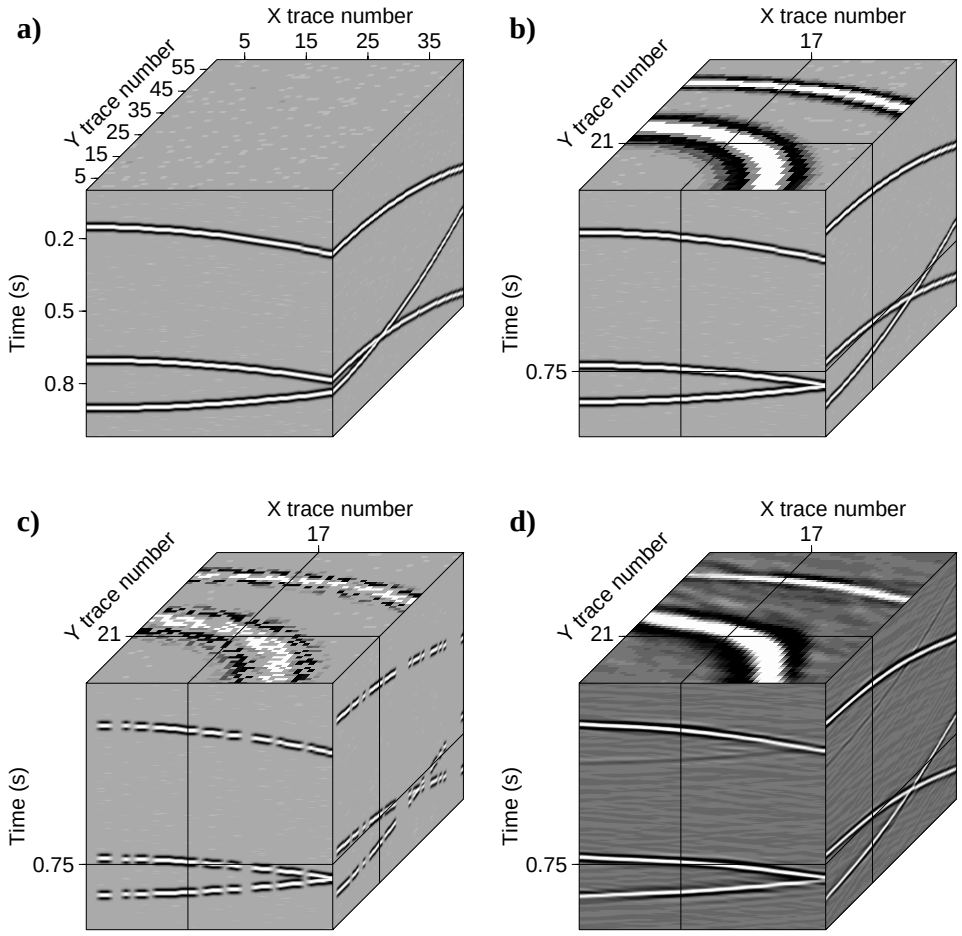

Figure 3.27: a) 3D synthetic cube of data with mildly-curved events. b) Slice view from inside the original data. c) Data with 60% randomly missing traces. d) Reconstructed data using MWNI.

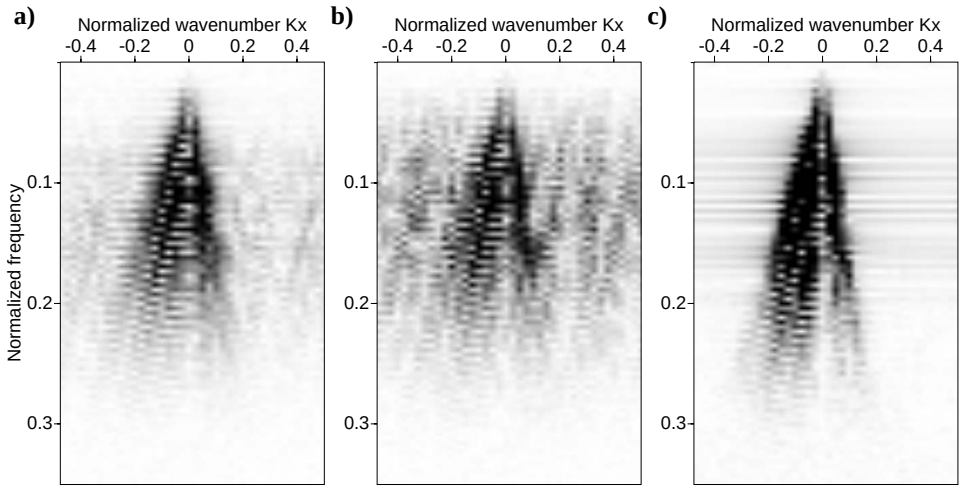

Figure 3.28: a), b), and c) are the *f-k* panels of the front views of the data in Figures 3.27b, 3.27c, and 3.27d.

3.33b shows the reconstructed data using MWNI method. Figure 3.34b shows the *f-k* panel of the front view of Figure 3.34b. The MWNI method was not able to eliminate the artifacts. Figure 3.34c shows the reconstructed data using band-limited MWNI method. The band-limitation was applied in the interval of $k_y = [-0.2, 0.2]$ normalized wavenumbers and full-band on k_x axis. The band-limitation on the Y direction (common-offset direction) makes sense since one would not expect strong dips as well as sharp changes in this direction. The reconstructed data still contains some leftover artifacts. This could be due to the presence of noise and curvature in the original data. Figure 3.34c shows the *f-k* panel of the front view of Figure 3.34c.

3.6 Conclusions

In this chapter I examined the effects of random sampling and regular sampling operators on the performance of the MWNI method. The spectrum of sampling operators and their influence on the original spectrum of the data were mathematically derived. The spectrum of the random sampling operators has only one peak and the rest of its components are small. Conversely, for regularly sampled data, depending on the decimation factor, several equal size impulses will be present in the spectrum domain. The latter is responsible for spectral mixing in the form of alias. The spectrum of the sampling operator is convolved with the spectrum of original data to give the spectrum of sampled data. Provided that

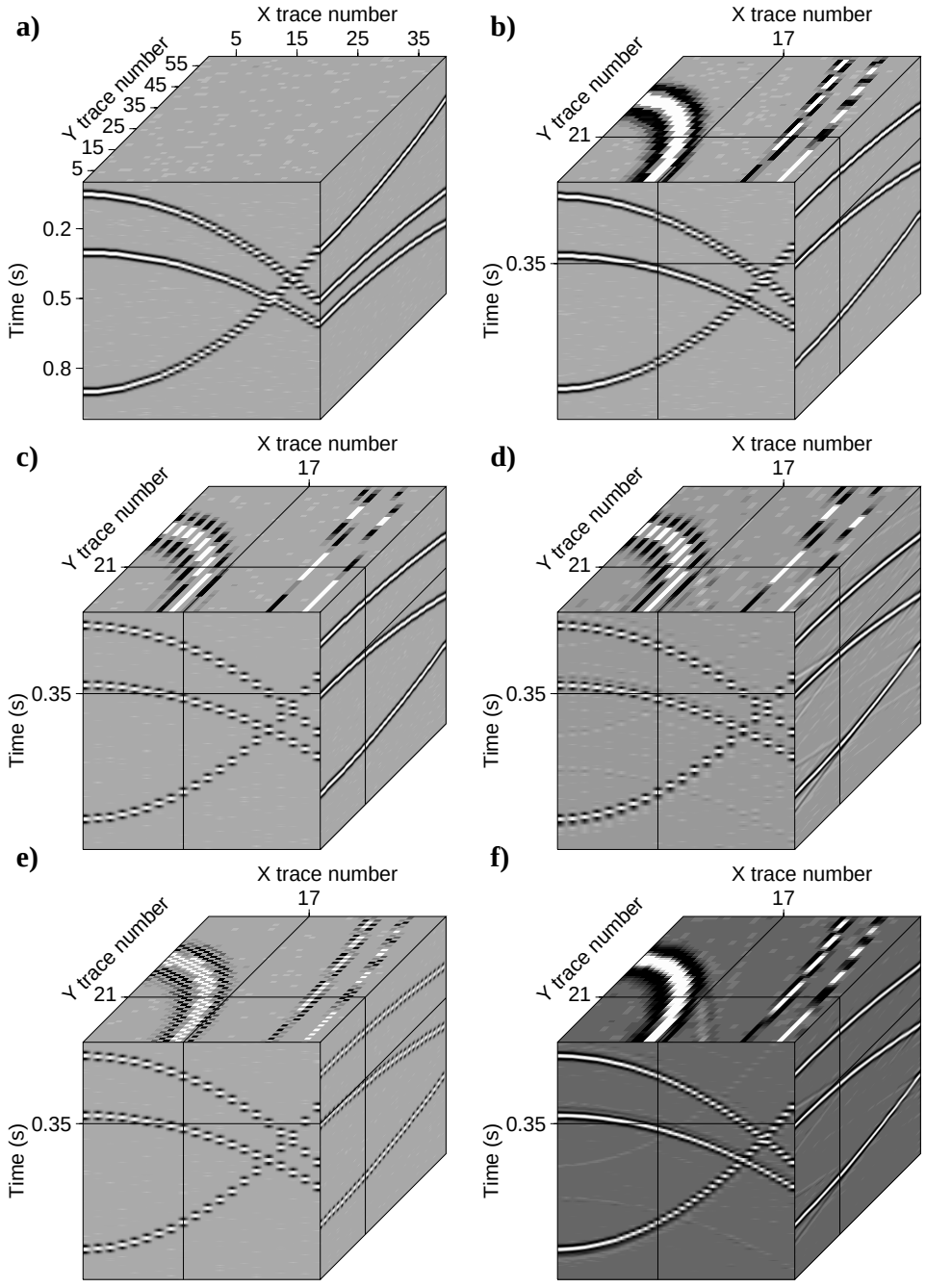

Figure 3.29: a) 3D synthetic cube of data with mild curvature in the Y direction and alias in the X direction . b) Slice view from inside the original data. c) Data after elimination of every other slice in the X direction. d) Reconstruction of (c) using band-limited MWNI. e) Data after chessboard elimination of traces. f) Reconstruction of (e) using band-limited MWNI.

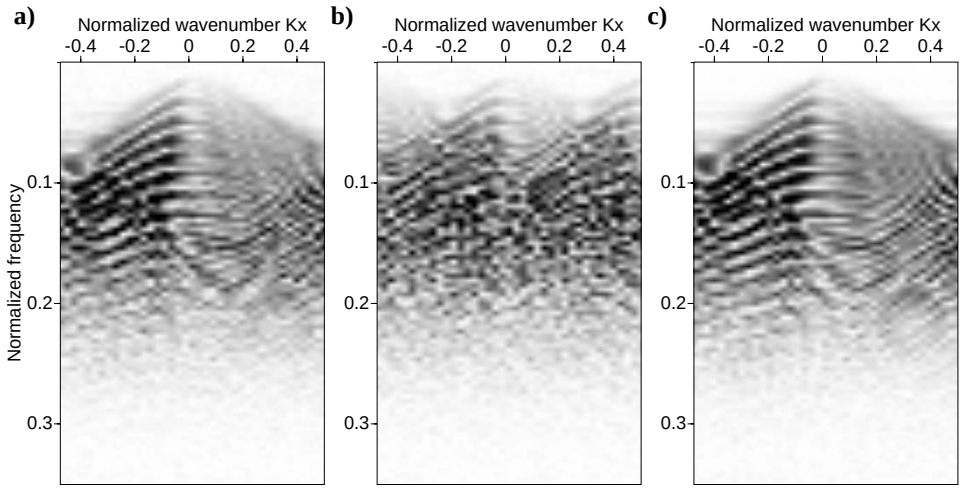

Figure 3.30: a), b), and c) are the *f-k* panels of the front views of the data in Figures 3.29b, 3.29e, and 3.29f.

the original data had a sparse spectrum, its convolution with the spectrum of the random sampling operator will still create a sparse signal with small amplitude artifacts. Therefore, randomly sampled signals are recoverable by assuming simplicity in the Fourier domain. The latter can be in the form of sparsity or via the MWNI constraint studied in this chapter. The spectrum of the regular sampling operator contains replications and its convolution with the spectrum of original data will produce unwanted repetitions of the spectrum of original data. Hence, eliminating the artifacts caused by the regular sampling operator using MWNI will be impossible unless the original data were band-limited.

For the multidimensional case (more than one spatial dimensions), the success of band-limited MWNI will depend on the type of regular sampling operator and band-limited properties of the original data. For a chessboard pattern decimation, the sampling artifacts in the Fourier domain will have the optimal possible separation from the spectrum of original data. This property helps to use the band-limitation information in one spatial direction to recover the signal in other aliased spatial directions.

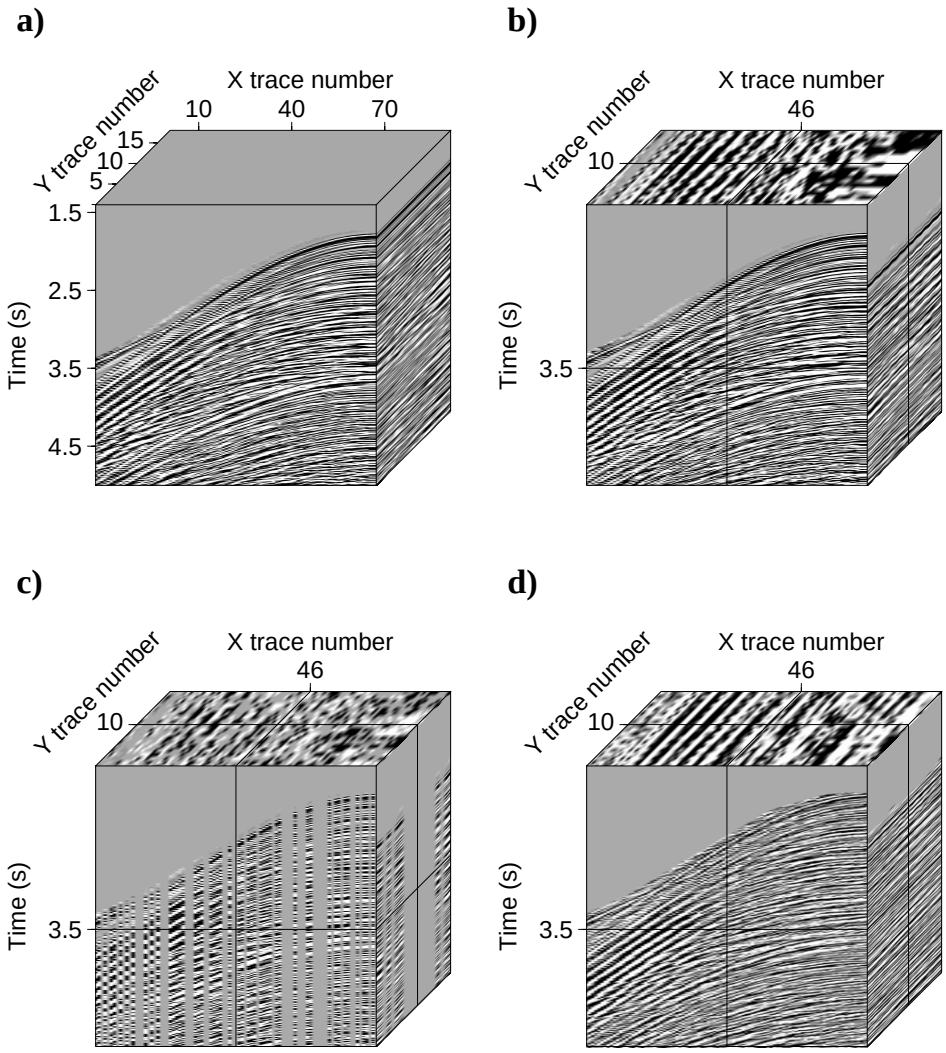

Figure 3.31: a) original 3D cube of data from Gulf of Mexico. b) Slices view from inside the original data. c) Data with 50% randomly missing traces. d) Reconstructed data using MWNI.

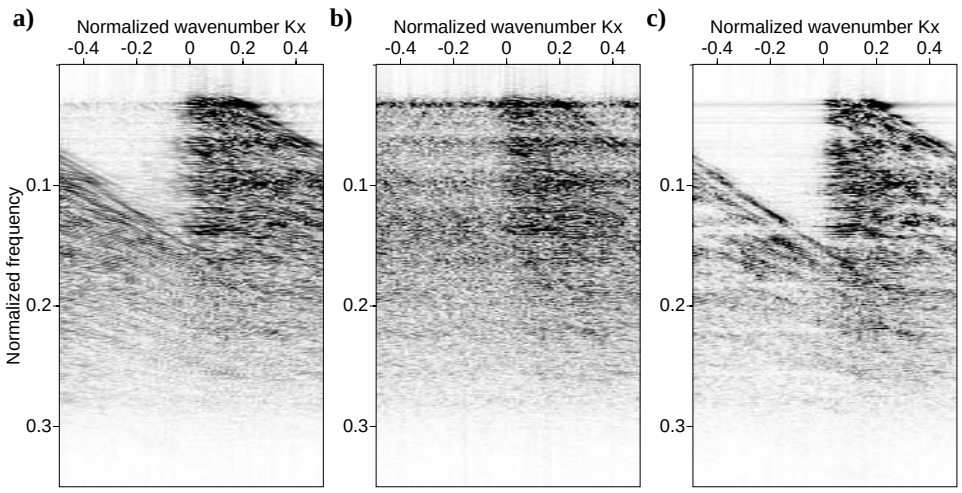

Figure 3.32: a), b), and c) are the *f-k* panels of the front views of the data in Figures 3.31b, 3.31c, and 3.31d.

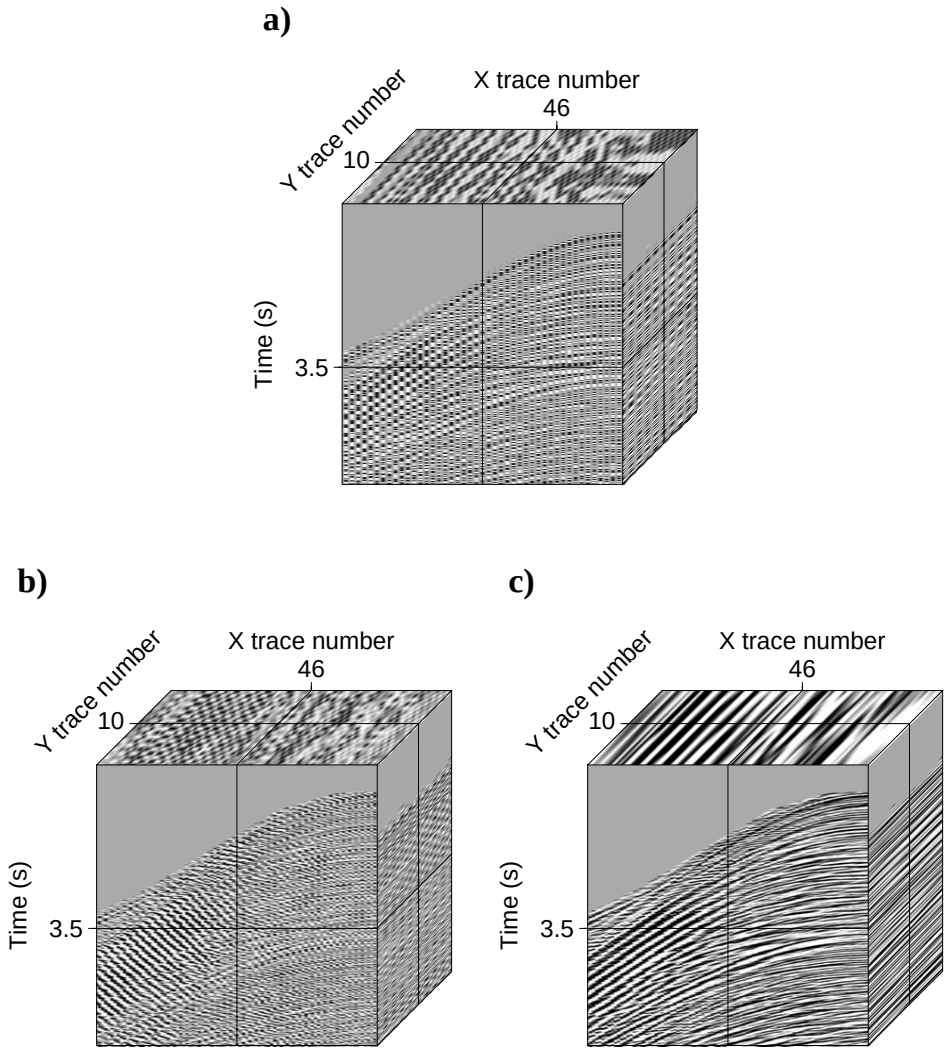

Figure 3.33: a) Decimation of data in Figure 3.31b with a chessboard pattern. b) Reconstructed data using MWNI. c) Reconstructed data using band-limited MWNI in the interval of $k_y = [-0.15, 0.15]$.

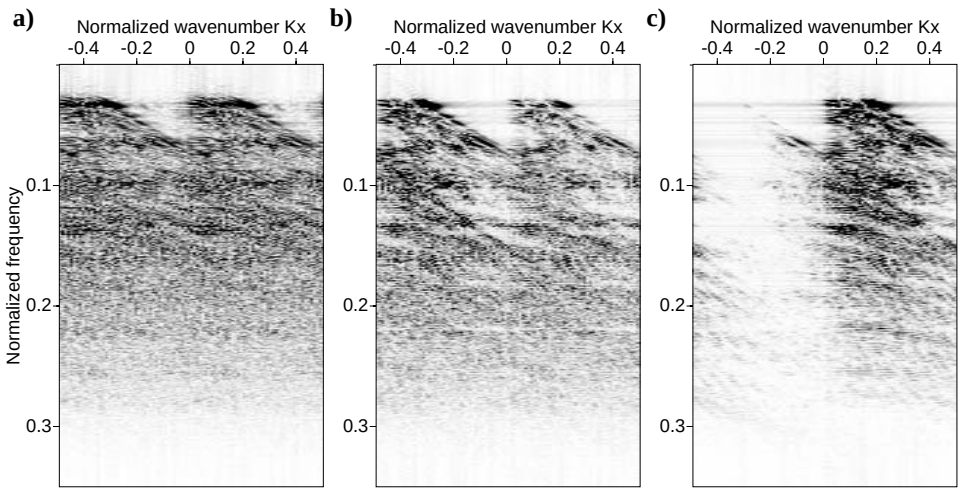

Figure 3.34: a), b), and c) are the *f-k* panels of the front views of the data in Figures 3.33a, 3.33b, and 3.33c.

CHAPTER 4

Multistep autoregressive reconstruction

4.1 Introduction

In this chapter I will introduce a new data reconstruction method which is a combination of Fourier reconstruction and autoregressive modeling. The aim is to alleviate the shortcomings of the aforementioned methods. In the previous chapter, problems related to Fourier reconstruction methods were discussed. In this chapter, I will expand more on autoregressive reconstruction. Autoregressive modeling has a broad range of applications in signal processing. It refers to techniques that model the evolution of a signal as a function of its past/future samples (Lau et al., 2002; Takalo et al., 2005; Picinbono and Benidir, 1986; Russ et al., 2004). Autoregressive modeling can be applied to signal prediction, de-noising (Canales, 1984), interpolation (Spitz, 1991; Porsani, 1999) and for parametric spectral analysis (Marple, 1987). Linear events in the t-x domain map to complex sinusoids in the f-x domain. Complex sinusoids can be modeled via autoregressive operators (McClellan, 1982; Kumaresan and Tufts, 1981). The autoregressive reconstruction methods proposed by Spitz (1991) and Porsani (1999) successfully address interpolating spatially aliased regularly sampled data. These methods utilize low frequency information to recover high frequency data components. Spitz (1991) proposed computing prediction filters (autoregressive operators) from low frequencies to predict interpolated traces at high frequencies. This methodology is applicable only if the original seismic section is regularly sampled in space. Conversely, irregularly sampled data can be reconstructed using Fourier methods. In this case the Fourier coefficients of the irregularly sampled data are retrieved by inverting the inverse Fourier operator with a band limiting (Duijndam et al., 1999) and/or a sparsity constraint (Sacchi et al., 1998; Liu and Sacchi, 2004; Zwartjes and Gisolf, 2006).

I introduce a new strategy that combines the strengths of both prediction error methods and Fourier based methods to cope with the problem of reconstructing non-uniformly sampled, aliased data. The first step of the proposed algorithm involves the reconstruction of irregularly missing spatial data on a regular grid at low frequencies using Fourier based algorithms. Due to the band-limited nature of the wavenumber spectra at low frequencies, this portion of the data can be reconstructed with high accuracy (Duijndam et al., 1999). Then, using a multistep autoregressive (MSAR) algorithm the prediction filters of all the frequencies are extracted from the reconstructed low frequency portion of the data. These prediction filters are utilized to reconstruct the missing spatial samples of each frequency. This step can be applied either in the *f-x* domain (MSAR-X) or *f-k* domain (MSAR-K).

It is important to stress that the technique presented in this chapter can only be used to reconstruct data that live on a regular grid with missing observations. Zwartjes and Sacchi (2007) introduced a method that combines the *f-k* algorithm of Gulunay (2003) and sparse inversion (Zwartjes and Gisolf, 2006) to reconstruct irregularly sampled aliased data. In their algorithm the Nonuniform Discrete Fourier Transform (NUDFT) is utilized to handle the irregularly sampled data.

4.2 Theory and implementation

Consider a seismic gather containing linear events. In addition, we assume that some traces in the gather are missing. By applying the Discrete Fourier Transform (DFT) with respect to time, the gather is transformed to the *f-x* domain. We let $\mathbf{x}(f)$ be the length-N vector of *f-x* data sampled on a regular grid $x_1(f), x_2(f), x_3(f), \ldots, x_N(f)$, of which only M traces are available. Let the sets of integers $\mathcal{K} = \{k(1), k(2), k(3), \ldots, k(M)\}$ and $\mathcal{U} = \{u(1), u(2), u(3), \ldots, u(N-M)\}$ represent the indices of the available (known traces) and missing samples (unknown traces), respectively. The goal is to recover $\mathbf{x}_{\mathcal{U}}(f)$ from $\mathbf{x}_{\mathcal{K}}(f)$.

4.2.1 Reconstruction of the unaliased portion of the data using MWNI

Fourier reconstruction methods are well suited to reconstruct seismic data in the low frequency (unaliased) portion of the Fourier spectrum. In addition, as it was shown by Duijndam et al. (1999), the reconstruction problem is well-conditioned at low frequencies where only a few wave numbers are required to honor the data. This makes the problem well-posed; therefore, it is quite easy to obtain a low frequency spatial reconstruction of the data. Seismic data at low frequencies are band-limited in the wavenumber domain. Figure

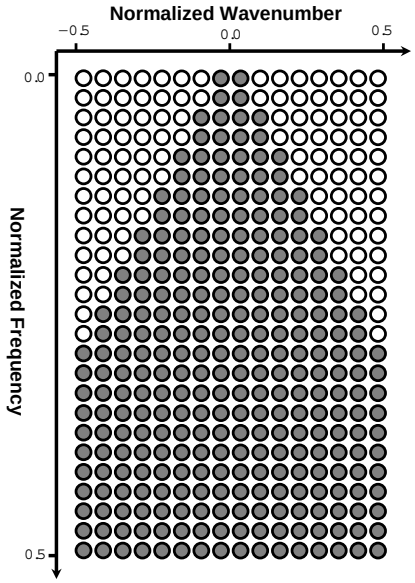

Figure 4.1: Schematic view of the $f-k$ domain representation of a seismic section. The area with empty circles indicates zero value samples.

4.1 shows a cartoon representing seismic data in the *f-k* domain. Empty circles represent zero values and grey circles indicate samples containing seismic information. The slope of the triangles made by the empty circles is determined by the minimum velocity present in the seismic section (Duijndam et al., 1999). The lower the frequency the smaller the number of wavenumber samples needed to represent the spatial signal. If we employ a Fourier reconstruction method the number of unknowns to be determined (wavenumbers) is small compared to the number of spatial samples, and as a result, the low frequency components can be reconstructed with high accuracy using Fourier inversion methods.

With the previous reasoning in mind, we first proceed to restore the low frequency portion of the data using a Fourier reconstruction method. In other words, we estimate the missing samples of $\mathbf{x}(f)$, that is $\mathbf{x}_{\mathcal{U}}(f)$ from $\mathbf{x}_{\mathcal{K}}(f)$ for temporal frequencies $f \in [f_{min_r}, f_{max_r}]$, where f_{min_r} and f_{max_r} denote the minimum and maximum (unaliased) frequencies in the data. In general, due to the band-limited nature of the seismic wavelet, we consider $f_{min_r} > 0$.

Recently, two Fourier-based reconstruction methods were introduced: Band Limited Fourier Reconstruction (BLFR) (Duijndam et al., 1999; Schonewille et al., 2003) and Minimum Weighted Norm Interpolation (MWNI) (Liu and Sacchi, 2004; Sacchi and Liu, 2005; Liu et al., 2004). In our implementation, we have adopted MWNI. It is important to mention, however, that similar results were obtained using BLFR. These methods can retrieve the

complex Fourier coefficients of the reconstructed data directly from the observations by inverting the inverse Fourier operator. The non-uniqueness of the reconstruction problem (Sacchi et al., 1998) is circumvented by the incorporation of a constraint in the form of a spectral norm. Details pertaining the MWNI method were discussed in Chapter 3.

4.2.2 Estimation of Prediction Filters using MSAR

Let us consider reconstructed data in the band $f \in [f_{min_r}, f_{max_r}]$. In Appendix A I show that linear events in the f-x domain can be predicted using Multi-Step Auto-Regressive (MSAR) operators of the form,

$$x_n(f) = \sum_{j=1}^{L} P_j(\alpha f) x_{n-\alpha j}(f), \qquad n = \alpha L + 1, \ldots, N, \tag{4.1}$$

$$x_n^*(f) = \sum_{j=1}^{L} P_j(\alpha f) x_{n+\alpha j}^*(f), \qquad n = 1, \ldots, N - \alpha L, \tag{4.2}$$

where $*$ denotes complex conjugate. These equations correspond to a special type of Auto-Regressive (AR) model where forward (equation (4.1)) and backward (equation (4.2)) AR equations are computed by "jumping" α steps at the time. The length of the AR operator is L and $P_j(f)$ is the prediction filter. The parameter $\alpha = 1, 2, ..., \alpha_{max}$ is the step factor used to extract the prediction filter for frequency αf from frequency f. Since the step factor is a positive integer it is clear that low frequencies provide vital information for our data reconstruction algorithm.

The parameter α_{max} is the upper limit of the step factor in equations 4.1 and 4.2. The latter depends on the number of traces N, and the length of prediction filter L. This parameter is given by

$$\alpha_{max} = \lfloor \frac{N - \frac{L+1}{2}}{L} \rfloor,$$

where $\lfloor . \rfloor$ denotes the integer part.

Equations (4.1) and (4.2) can be considered an extension of the prediction filters used by Spitz (1991). In our case, however, multiple prediction filters are extracted for a given high frequency $f' = \alpha f$. In other words, all possible combinations of α and f leading to the product $f' = \alpha f$ will deliver a prediction filter that can be used to reconstruct data at frequency component f'. Since more than one prediction filter can be found for the reconstruction of $\mathbf{x}(f')$, our algorithm utilizes the average of prediction filters. Alternatively, we could have estimated one prediction filter by solving an augmented system of equations with contributions from more than one low frequency. The final result of both strategies

should be quite similar. We have adopted the former rather than the latter because it leads to a computationally more efficient reconstruction algorithm.

Figure 4.2 shows a simple example of computing prediction filters for different steps α. Prediction filters are computed by setting a linear system of equations using forward and backward prediction for each α step. The forward and backward application is based on the well-known covariance least squares prediction algorithm (Honig and Messerschmidt, 1984). For this specific example, with 20 data samples and length of prediction filters equal to 3, the number of resulting equations for a linear system are shown in the black squares at the right hand side of the Figure 4.2. In this example, the maximum step which leads to a suitable number of equations for a linear system of equations is $\alpha_{max} = 6$. It should be mentioned that the backward prediction is applied on the complex conjugate of the data samples (Spitz, 1991). Note that such forward and backward filter design can not predict the data if the magnitudes are laterally varying as in the case of data modeled via decaying oscillations. One can use an adaptive prediction filtering scheme to cope with this issue (Naghizadeh and Sacchi, 2009a).

Figure 4.3 shows the MSAR algorithm implementation in the frequency domain. First, using MWNI a band of frequencies is reconstructed in the interval (f_{min}, f_{max}). The MSAR algorithm finds the prediction filters for the band $(\alpha f_{min}, \alpha f_{max})$ for each step α. We finally average the prediction filters of frequencies which happen to have more than one prediction filter. This is a great property of MSAR since the estimated prediction filters for a single frequency are extracted from different parts of the low frequency band. Hence, the averaging of prediction filters can also be regarded as a noise reduction method.

To continue with our analysis, a few comments are in order. The MSAR strategy requires us to find prediction filters in the reconstructed band $[f_{min_r}, f_{max_r}]$, where min_r and max_r are discrete frequency indices. We might encounter the case where for some frequency f', the frequency f'/α may not fall in the reconstructed interval $[f_{min_r}, f_{max_r}]$; and then, the MSAR method will not be applicable. This situation can be solved by extrapolating prediction filters using the method proposed in Appendix B of Spitz (1991). It should be mentioned that this situation can be avoided by properly choosing the values of f_{min_r} and f_{max_r} in such a way that $max_r \geq 2(min_r - 1)$. In the sections devoted to examples, we provide a synthetic test where the extrapolation of prediction filters is required to reconstruct aliased data.

4.2.3 Reconstruction in the *f-x* domain (MSAR-X)

So far we have outlined a method to extract, from reconstructed low frequency data components $\mathbf{x}(f)$, prediction filters for high frequency data component $\mathbf{x}(f') = \mathbf{x}(\alpha f)$.

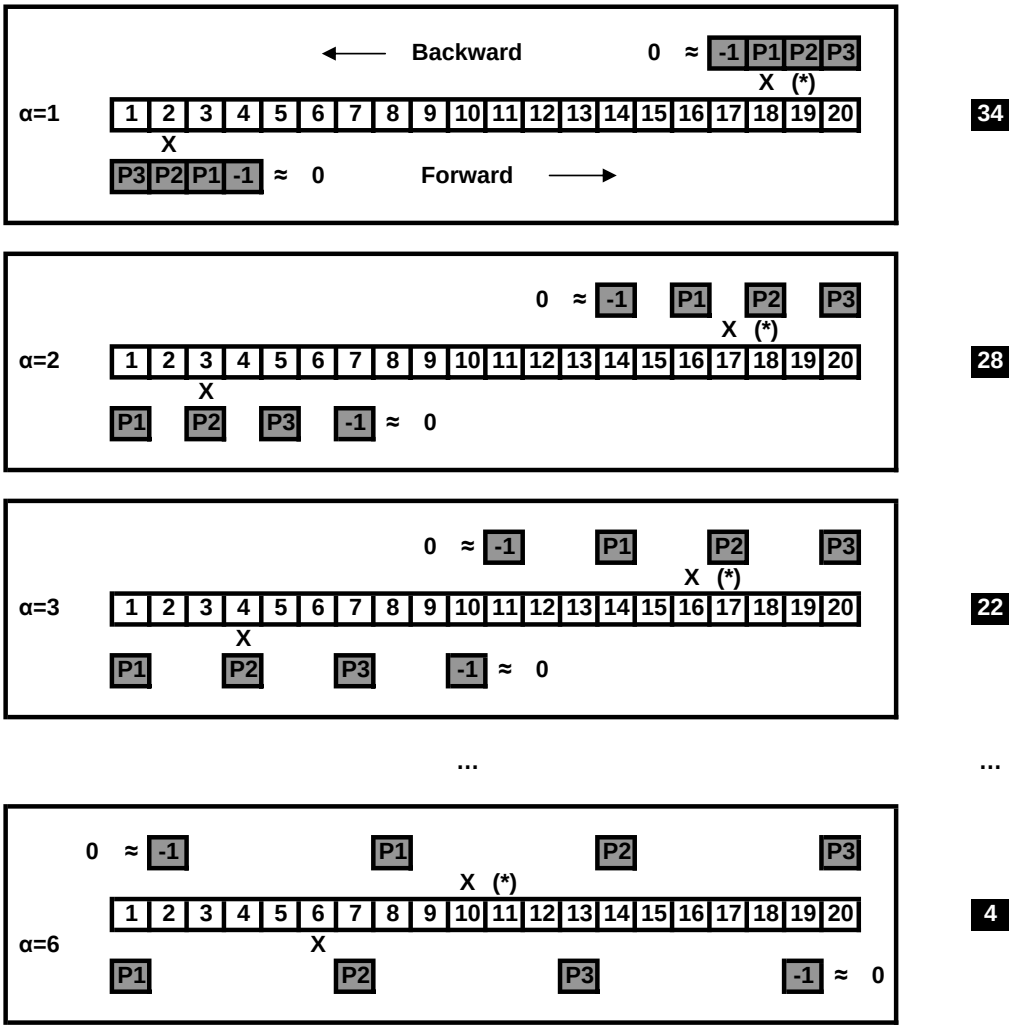

Figure 4.2: Flowchart of the MSAR algorithm (1D example). The gray and white squares show the prediction filters and data samples, respectively. The black squares at the right side represent the number of equations for the resultant linear system of equation for each α step.

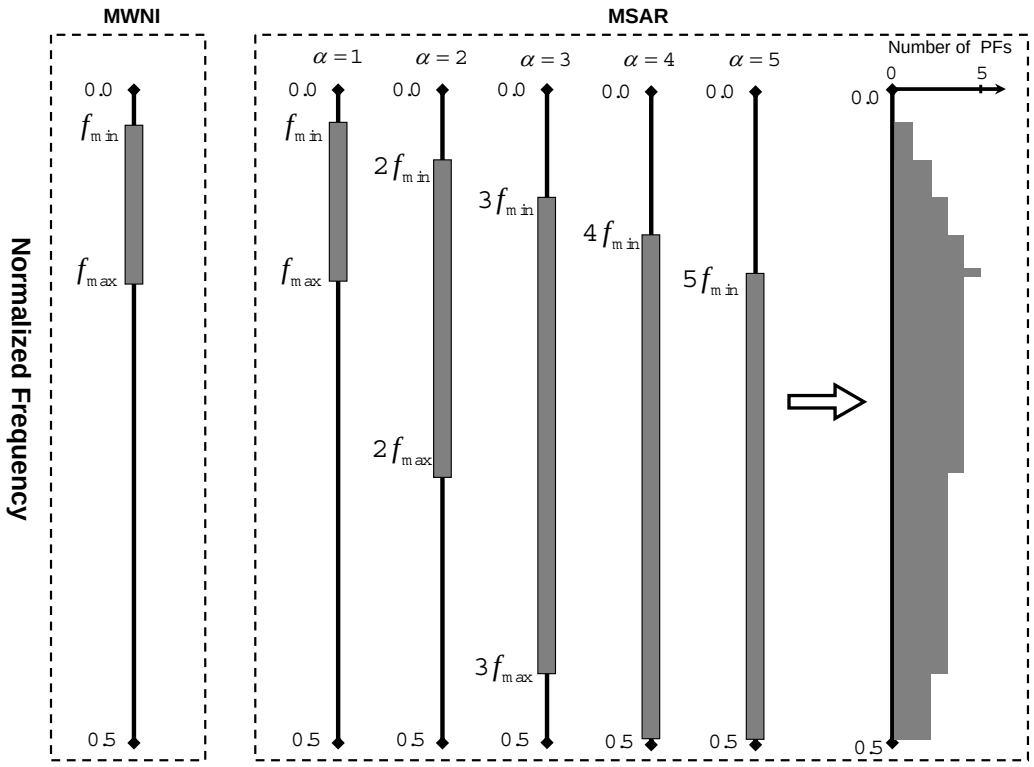

Figure 4.3: Flowchart of the MSAR implementation on the frequency axis.

Following the procedure proposed by Wiggins and Miller (1972) and Spitz (1991), one can compute the missing samples from known data and prediction filter coefficients. In this case, the forward and backward autoregressive equations,

$$x_n(f') = \sum_{j=1}^{L} P_j(f') x_{n-j}(f'), \qquad n = L+1, \ldots, N, \tag{4.3}$$

$$x_n(f') = \sum_{j=1}^{L} P_j^*(f') x_{n+j}(f'), \qquad n = 1, \ldots, N-L, \tag{4.4}$$

are used to isolate the unknown data components. Equations (4.3) and (4.4) are the specific forms of equations (4.1) and (4.2) when $\alpha = 1$. It is also important to note that equation (4.4) is obtained by taking the complex conjugate of both sides of equation (4.2). Now both forward and backward prediction equations contain the same unknown $x(f')$. The rest of the procedure is quite simple: we first expand equations (4.3) and (4.4). The resulting system of equations contains linear combinations of known and unknown samples. The system is algebraically manipulated and rewritten as follows:

$$\widetilde{\mathbf{A}}(\mathbf{P}(f')) \begin{bmatrix} x_{u(1)}(f') \\ x_{u(2)}(f') \\ \vdots \\ x_{u(N-M)}(f') \end{bmatrix} = \widetilde{\mathbf{B}}(\mathbf{P}(f')) \begin{bmatrix} x_{k(1)}(f') \\ x_{k(2)}(f') \\ \vdots \\ x_{k(M)}(f') \end{bmatrix}. \tag{4.5}$$

The notation $\widetilde{\mathbf{A}}(\mathbf{P}(f'))$ and $\widetilde{\mathbf{B}}(\mathbf{P}(f'))$ reflects the fact that these two matrices only depend on the coefficients of the prediction filter. Equation (4.5) closely follows equation 5 of Spitz (1991). The main difference, however, is that the known and unknown traces are no longer interlaced but are given at arbitrary positions. In addition, it is important to stress that $\mathbf{P}(f')$ is the average of prediction filters computed from reconstructed low frequencies using the MSAR scheme. Hence, the missing samples are computed using:

$$\mathbf{x}_{\mathcal{U}}(f') \;=\; [\,\widetilde{\mathbf{A}}^{\star}(\mathbf{P}(f'))\widetilde{\mathbf{A}}(\mathbf{P}(f')) \;+ \mu\mathbf{I}]^{-1}\; \widetilde{\mathbf{A}}^{\star}(\mathbf{P}(f'))\, \widetilde{\mathbf{B}}(\mathbf{P}(f'))\, \mathbf{x}_{\mathcal{K}}(f'), \tag{4.6}$$

where $\mathbf{x}_{\mathcal{U}}(f')$ and $\mathbf{x}_{\mathcal{K}}(f')$ indicate the vectors of unknown and known data samples, respectively. In addition, $\widetilde{\mathbf{A}}^{\star}$ stands for the transpose and complex conjugate of $\widetilde{\mathbf{A}}$, $\mathbf{I}$ is the identity matrix and μ is a regularization parameter required to stabilize the inversion in the presence of noise. In this dissertation the method of Conjugate Gradients is used to solve equation (4.5) (Hestenes and Stiefel, 1952).

Figure 4.4a shows how to find prediction filters from regular samples of a signal. From the available 10 samples of data (white cells) we try to find a prediction filter (black cells). The

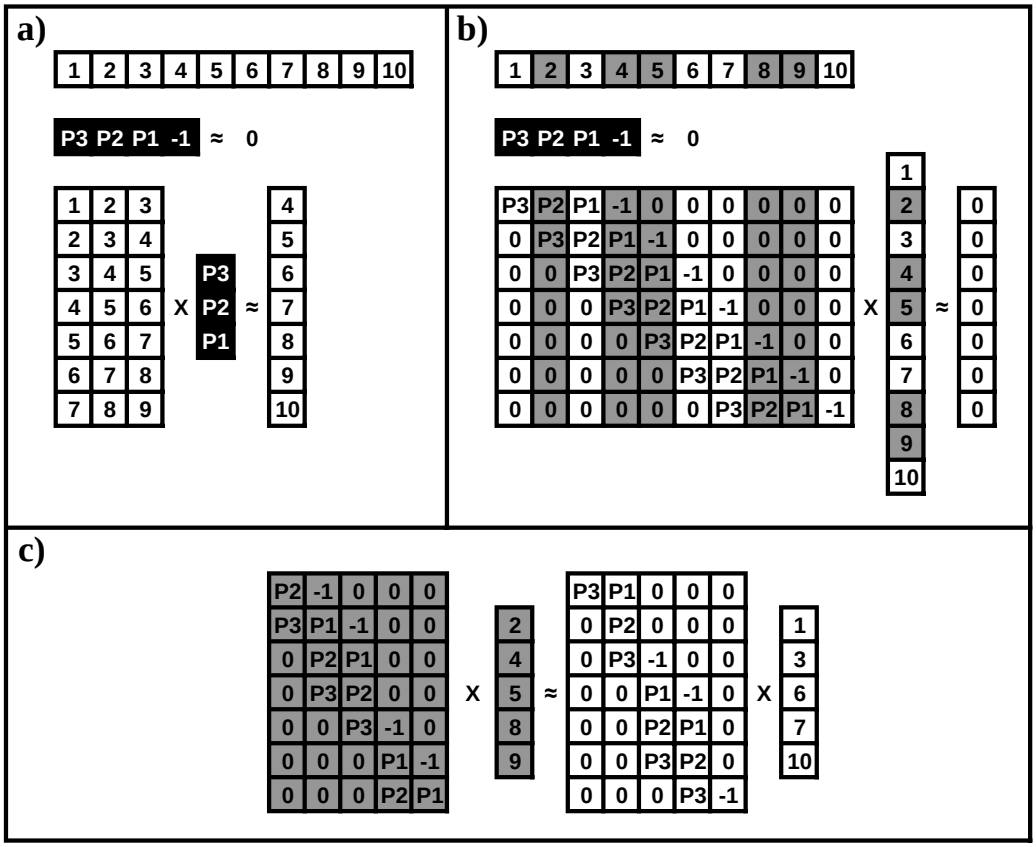

Figure 4.4: Flowchart of AR reconstruction for irregularly sampled data. a) Computing prediction filter from known data samples. b) and c) Computing unknown samples (gray cells) of the data using known prediction filters.

linear system of equations in the bottom shows the system that has to be solved to find the prediction filter. Notice that for simplicity of representation we just show the forward prediction. Once the values of the prediction filters are known one can use them to find the values of missing samples in the signal. Figure 4.4b shows how to make a linear system of equations in order to find the missing samples (gray cells) from known samples (white cells) utilizing a known prediction filter. We separate the columns of the Hankel matrix into the white cells (always multiplied by known samples) and the gray cells (always multiplied by unknown samples). The final step, shown in Figure 4.4c, is to move the known samples with their correspondent columns of the Hankel matrix to the right hand side and leave the unknowns on the left hand side. This way we obtain a system of equations that can be solved to find the unknown samples.

4.2.4 Reconstruction in the *f-k* domain (MSAR-K)

Utilizing prediction filters (AR operators) in the space domain for the purpose of data reconstruction suffers from some shortcomings. First, it could not recover the gaps with more missing samples than the length of the prediction filter. This is also true for data extrapolation. Second, finding a reasonable regularization term for the resultant under-determined linear system of equations (more unknown samples than known ones) in the space domain is very difficult. To overcome these shortcomings one needs to transform the under-determined problem to an over-determined one. One way to achieve this is to analyze the data in other domains. One of the most useful representation of data is their Fourier representation.

The AR spectrum of data can be computed from 1D prediction filter with order M using

$$S_{pf}(\kappa) = \frac{\sigma_\epsilon^2}{\left|1 - \sum_{m=1}^{M} P_m e^{-i2\pi m\kappa}\right|^2}, \quad -\frac{1}{2} \leq \kappa \leq \frac{1}{2} \tag{4.7}$$

where κ is the normalized frequency (wavenumber) and σ_ϵ^2 is the noise variance.

The AR spectrum of data is a smoothly varying function with peak values at the dominant frequencies (wave-numbers). While the location and number of peaks are accurate, the amplitude information can be variable and unreliable. However, the most important information needed for data reconstruction is the location and number of dominant frequencies. Hence, after computing the AR spectrum of data, one needs to identify the location of the peaks.

Identifying the location of peaks in the AR spectrum of data can be achieved by a basic sample by sample comparison algorithm. Considering discrete values for the frequency (wavenumber) axis as $\kappa = (k_1, k_2, \ldots, k_q)$, the peak values of the AR spectrum of data can found by

$$\Lambda_{pf}(k_j) = \begin{cases} 1 & \textbf{if} \quad S_{pf}(k_{j-1}) < S_{pf}(k_j) > S_{pf}(k_{j+1}) \\ 0 & \textbf{otherwise} \end{cases} \tag{4.8}$$

where the value 1 in function Λ_{pf} will indicate the peak location. While in the 1D case the peak location is identified by comparing the current sample with the two samples around it, in the 2D case the current sample will be considered as a peak value if it was greater than all the 8 samples around it. Similarly in the 3D case, the current sample should be greater than all the 26 samples around it to be considered as a peak value.

In order to take into account any uncertainties on the estimation of prediction filter, the location of peaks can be widened to several samples around the identified peak value. This

can be achieved by convolving a box function Γ_l with Λ_{pf}:

$$\Upsilon_{pf} = \Gamma * \Lambda_{pf}, \tag{4.9}$$

where, Υ_{pf} is considered as a multi-band-pass filter derived from the AR spectrum of data and l is the length of box function. Since the convolution operator is used here only to widen the peak location, any possible values bigger than 1 in Υ_{pf} should be truncated to 1. The last step is to replace Υ in Equation (3.21) with Υ_{pf} and utilize MWNI to reconstruct the missing samples. This means that a low-pass function (Υ) is replaced with a multi-band-pass function (Υ_{pf}).

4.3 Examples

4.3.1 Synthetic examples

In order to examine the performance of the MSAR reconstruction technique we analyze synthetic data examples. Amplitude spectra are portrayed in terms of normalized frequency and normalized wavenumber. Normalized frequency and wavenumber axes are obtained by considering $\Delta t = 1$ and $\Delta x = 1$, respectively. This means that in order to obtain frequency axes in Hz and Cycles/m one must divide the normalized ones by Δt and Δx, respectively.

The first synthetic data example consists of three severely aliased linear events (Figure 4.5a). Next we randomly eliminated 70% of the traces from the original data to obtain a section with missing traces (Figure 4.5b). Figures 4.5c and 4.5d depict the *f-k* panels of Figures 4.5a and 4.5b, respectively. First, the missing traces are reconstructed using MWNI for all frequencies (Figure 4.6a). While MWNI method was able to eliminate most of the artifacts in the *f-k* domain (Figure 4.6e), it was not able to erase some of the high amplitude artifacts. Next, a band of low frequencies from the normalized frequencies 0.03 to 0.07 (Figure 4.6b) is reconstructed. The *f-k* panel of Figure 4.6b is shown in Figure 4.6f. We must reiterate that this is the portion of reconstructed data from where the MSAR technique will extract the prediction filters needed to reconstruct high frequency components. Due to lack of information, the very low frequencies were excluded from the reconstruction. Prediction filters for the low frequency end of the data could have been estimated by extrapolation of prediction filters as suggested in Spitz (1991, Appendix B). Figure 4.7 shows the number of prediction filters extracted for each frequency using the MSAR method. After extracting prediction filters from low frequency data components, the complete data spectrum (for normalized frequencies in the range 0 to 0.5) was reconstructed with the MSAR-X method (Figure 4.6c). The length of the prediction filter is chosen equal to the number of linear

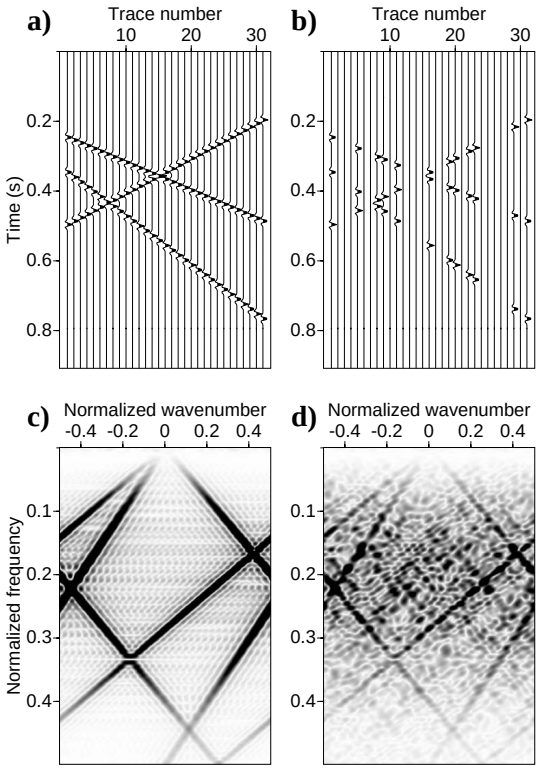

Figure 4.5: 2D synthetic example. a) Original data. b) Data with 70% randomly missing traces. c) and d) are the f-k panel of a and b, respectively.

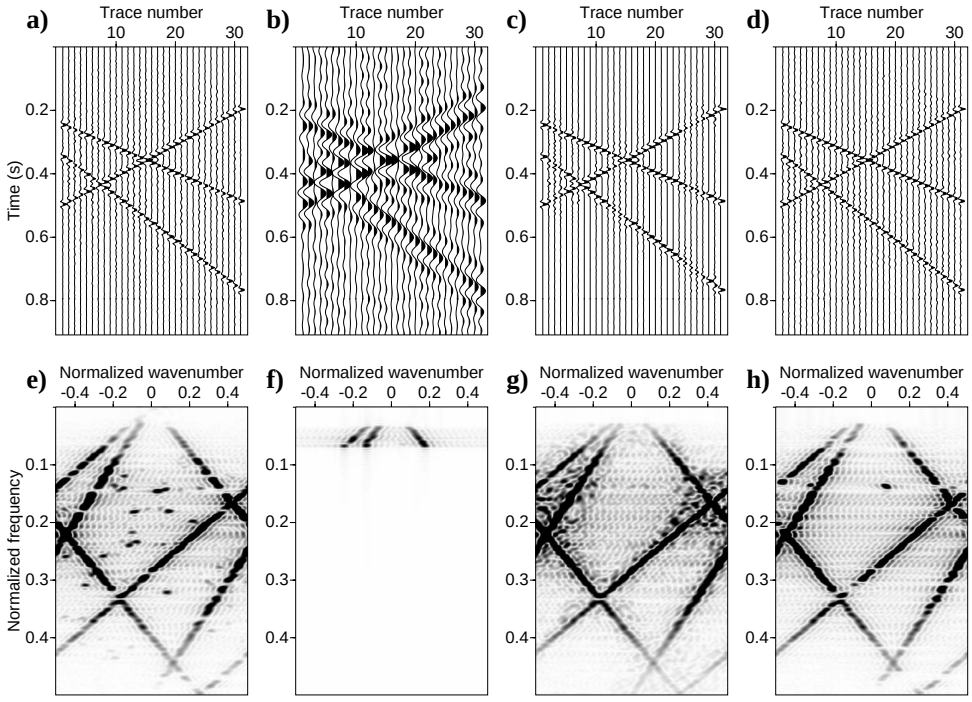

Figure 4.6: Reconstruction of missing data in Figure 4.5b. a) MWNI reconstruction of all frequencies. b) MWNI reconstruction of the normalized frequencies 0.03 to 0.07. c) MSAR-X reconstruction. d) MSAR-K reconstruction. e), f), g), and h) are the *f-k* panel of (a), (b), (c), and (d), respectively.

events ($L = 3$). The *f-k* panel of Figure 4.6c is depicted in Figure 4.6g. In this case, the reconstruction algorithm struggles to recover the missing data samples for large gaps. It is also clear from the *f-k* panel (Figure 4.6g) that MSAR-X is not able to eliminate the artifacts close to the original signal. Figure 4.6d shows the reconstruction result of using the MSAR-K method in the *f-k* domain. Comparing the *f-k* panel of Figure 4.6d (Figure 4.6h) with Figures 4.6e and 4.6g shows that by utilizing the AR spectrum of data most of the artifacts in *f-k* domain are eliminated. Figure 4.8 shows the AR spectrum of data (solid line with solid circles) for the normalized frequency 0.3 and the multi-band-pass filter (Dashed line). By a comparison between the *f-k* panel of original data for the normalized frequency 0.3 (Figure 4.5c) and the location of peaks in Figure 4.8, it is obvious that the AR spectrum of data has successfully identified the correct location of the original signal.

While the previous example was applied to a section of data with randomly missing traces, it is desirable also to examine the performance of seismic data reconstruction methods for different sampling operators. The sampling operator is an important factor which determines

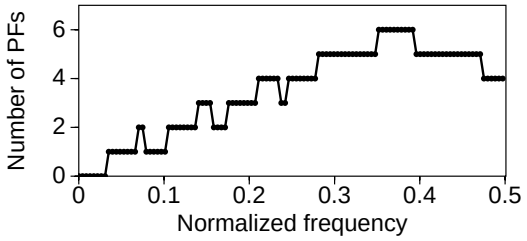

Figure 4.7: The number of prediction filters contributing to each frequency components in the example in Figures 4.6c and 4.6d. The average filter for any given frequency is used in the reconstruction stage.

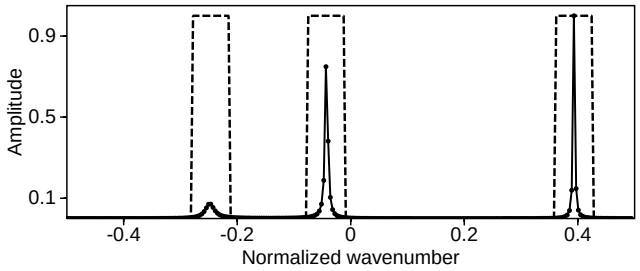

Figure 4.8: The AR spectrum of data (Solid line with solid circles) and multi-band-pass filter (dashed line) for the normalized frequency 0.3 used for MSAR-K reconstructed data shown in Figure 4.5d.

the success or failure of the implemented reconstruction algorithm (Naghizadeh and Sacchi, 2008, 2009c). For the rest of this section I will compare the performance of the MSAR-X, MSAR-K and MWNI methods. It should be mentioned that for the MSAR-X and MSAR-K methods, we will first reconstruct the low frequency band between 0.03 to 0.07 normalized frequency in order to extract the prediction filters for all frequencies. The result of low frequency reconstruction will not be shown and only the final results of the MSAR-X and MSAR-K methods will be depicted.

To examine the performance of the proposed method for regular sampling operators, I eliminated every other traces of the original data (Figure 4.5a) to create a section with regularly missing traces (Figure 4.9a). The missing data in Figure 4.9a is reconstructed using MWNI (Figure 4.9b), MSAR-X (Figure 4.9c) and MSAR-K (Figure 4.9d). The f-k panels of Figures 4.9a, 4.9b, 4.9c, and 4.9d are shown in Figures 4.9e, 4.9f, 4.9g, and 4.9h, respectively. The MWNI fails to reconstruct the missing data while the MSAR-X and MSAR-K methods were able to restore the missing traces. The artificial events created by regular sampling (alias) are of the same amplitude as the original spectrum of data. Therefore, the MWNI method fails to eliminate the unwanted events. The success of MSAR-X and MSAR-K methods relates to the extraction of information from the low frequency (non-aliased) portion of data. Figure 4.10 shows the AR spectrum of data (solid line with solid circles) as well as the derived multi-band-pass filter (dashed line) for the normalized frequency 0.3. The identified dominant wavenumbers are the same as the ones in the original spectrum of data.

Next we examine the reconstruction of a example data with a gap (Figure 4.11a). MWNI, MSAR-X and MSAR-K reconstruction of Figure 4.11a are shown in Figures 4.11b, 4.11c, and 4.11d, respectively. The existence of a gap in the t-x domain, causes low amplitude artifacts around the main events in the f-k domain (Figure 4.11e). While the f-k panel of MWNI (Figure 4.11f) and MSAR-K (Figure 4.11h) show that these artifacts are removed successfully, the low amplitude artifact still remains in the f-k domain of MSAR-X reconstructed data (Figure 4.11g). This means that using the AR spectrum of data in the wavenumber domain (MSAR-K) is more effective to fill the gaps in data than using the prediction filters in the space domain (MSAR-X). Figure 4.12 shows the AR spectrum of data (solid line with solid circles) as well as the multi-band-pass filter (dashed line) for the normalized frequency 0.3. Notice that the identified dominant wavenumbers in Figure 4.12 are almost the same as the ones in Figures 4.10 and 4.8, even though the amplitudes of the identified peaks are different.

It is also interesting to investigate the case of extrapolation of data in the space domain. Figure 4.13a was created by eliminating 7 traces from each side of the original data (Figure 4.5a). Figure 4.13e shows the f-k panel of Figure 4.13a. Eliminating traces from the sides of

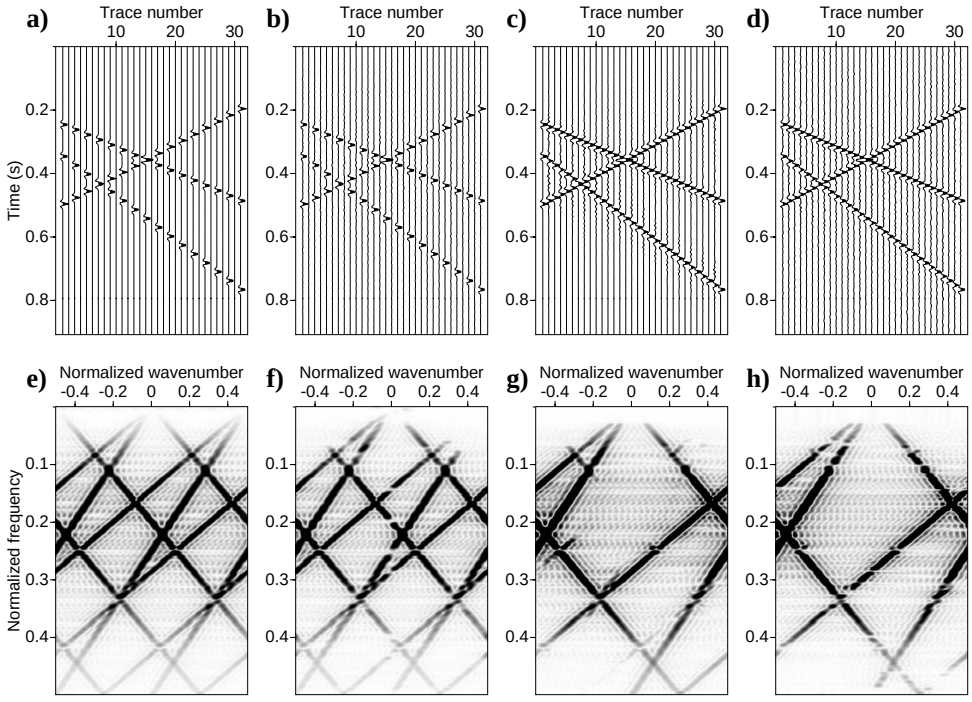

Figure 4.9: 2D synthetic example with regularly missing traces. a) Section of missing traces created from Figure 4.5a. b), c), and d) are the reconstruction of a using the MWNI, MSAR-X, and MSAR-K methods, respectively. e), f), g), and h) are the *f-k* panel of (a), (b), (c), and (d), respectively.

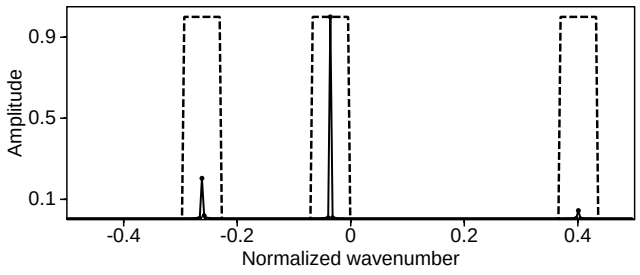

Figure 4.10: The AR spectrum of data (Solid line with solid circles) and multi-band-pass filter (dashed line) for the normalized frequency 0.3 used for MSAR-K reconstructed data shown in Figure 4.9d.

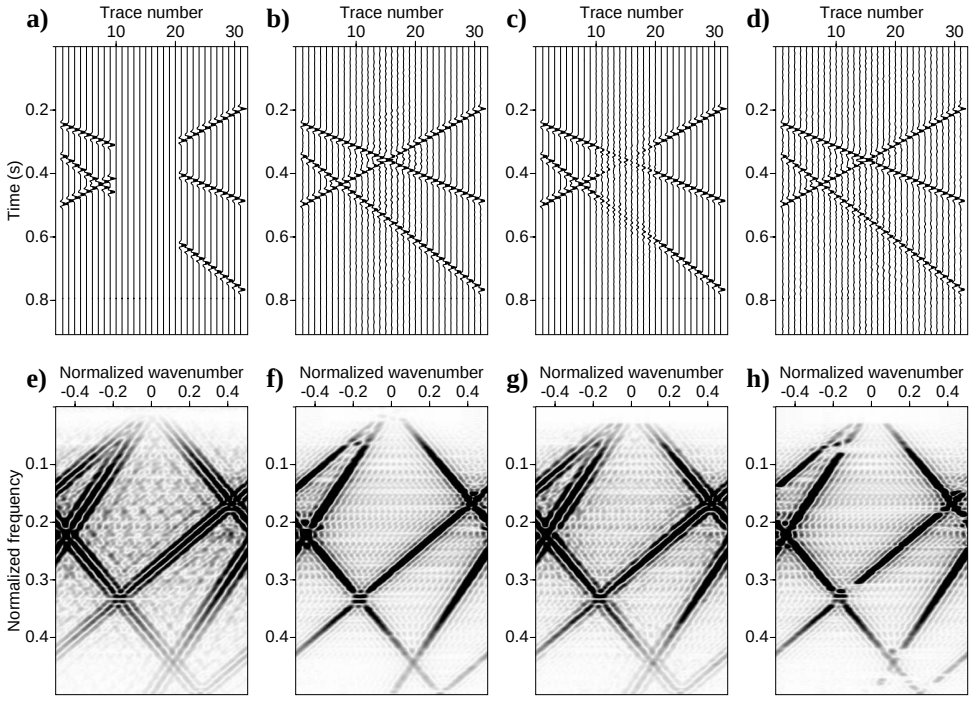

Figure 4.11: 2D synthetic example with a gap. a) Section of missing traces created from Figure 4.5a. b), c), and d) are the reconstruction of a using the MWNI, MSAR-X, and MSAR-K methods, respectively. e), f), g), and h) are the *f-k* panel of (a), (b), (c), and (d), respectively.

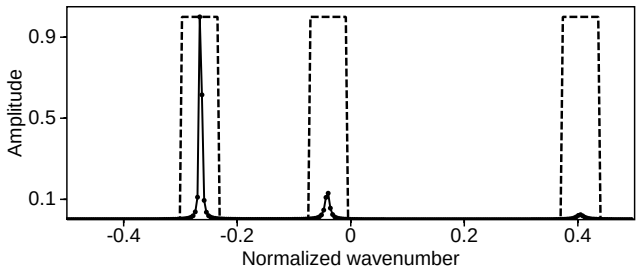

Figure 4.12: The AR spectrum of data (Solid line with solid circles) and multi-band-pass filter (dashed line) for the normalized frequency 0.3 used for MSAR-K reconstructed data shown in Figure 4.11d.

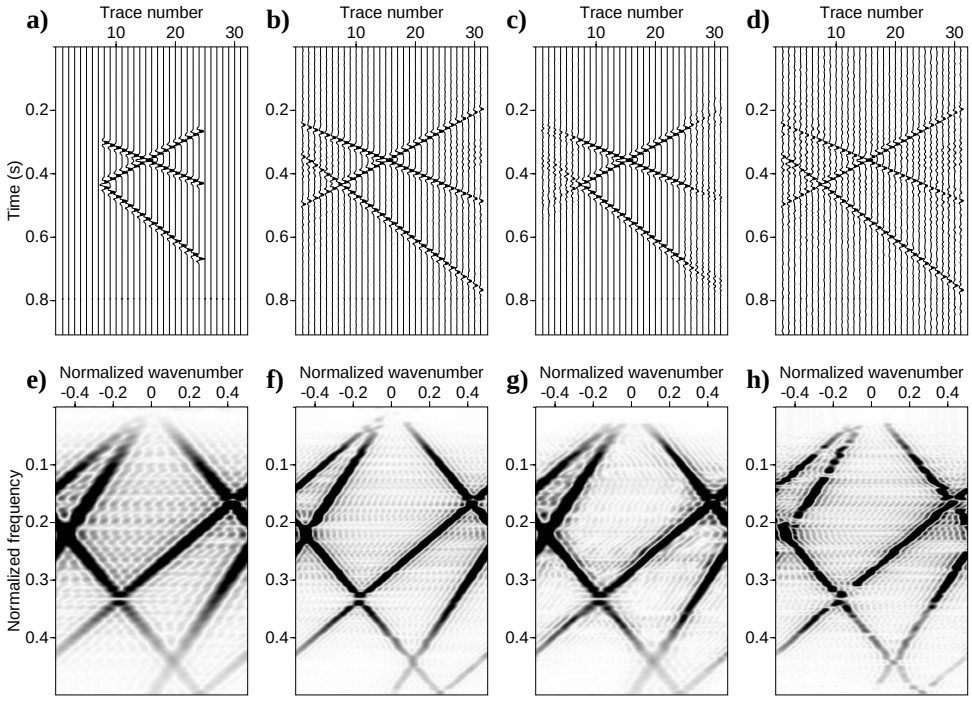

Figure 4.13: 2D synthetic example for testing extrapolation. a) Section of missing traces created from original data in Figure 4.5a. b), c), and d) are the reconstruction of a using the MWNI, MSAR-X, and MSAR-K methods, respectively. e), f), g), and h) are the *f-k* panel of (a), (b), (c), and (d), respectively.

the data widens the original spectrum of data in the *f-k* domain. Figures 4.13b, 4.13c, and 4.13d show the results of reconstruction using MWNI, MSAR-X, and MSAR-K methods, respectively. Figures 4.13f, 4.13g, and 4.13h show the *f-k* panel of Figures 4.13b, 4.13c, and 4.13d, respectively. While the MWNI and MSAR-K methods were able to extrapolate the data, MSAR-X failed to do so. The AR spectrum of data and the derived multi-band-pass filter for the normalized frequency 0.3 is shown in Figure 4.14. The multi-band-pass filter (dashed lines in Figure 4.14) was used as a weight function to indicate the location of original data in the *f-k* domain.

Figure 4.15a shows another example of a sampling function composed of a decimation process as well as a gap. Figures 4.15b, 4.15c, and 4.15d show the reconstructed data using MWNI, MSAR-X, and MSAR-K, respectively. The *f-k* panel of Figures 4.15a, 4.15b, 4.15c, and 4.15d are depicted in Figures 4.15e, 4.15f, 4.15g, and 4.15h, respectively. It is interesting that MWNI is only able to recover the decimated data and still can not fill in the regularly missing traces. Meanwhile, the MSAR-X method successfully fills the decimated data outside the

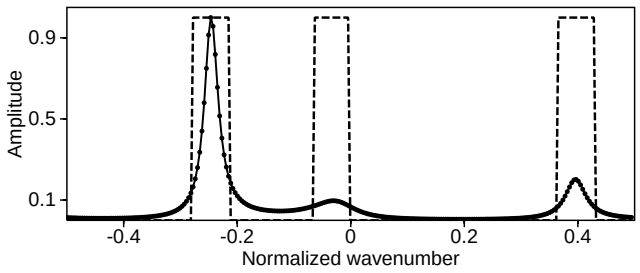

Figure 4.14: The AR spectrum of data (Solid line with solid circles) and multi-band-pass filter (dashed line) for the normalized frequency 0.3 used for MSAR-K reconstructed data shown in Figure 4.13d.

gap area but struggles to recover the data at the gap location. It is important to notice that the MSAR-K method overcomes the shortcomings of both MWNI and MSAR-X methods and recovers all the missing traces successfully. The AR spectrum of data and multi-band-pass filter built based on spectrum peaks was shown in Figure 4.16 for the normalized frequency 0.3 .

The next 2D example is a section with decimated traces as well as missing traces on the sides (Figure 4.17a). In this case the reconstructed data using MWNI (Figure 4.17b) and MSAR-X (Figure 4.17c) are not fully recovered, while the MSAR-K method has successfully reconstructed all the missing data. The *f-k* panels of Figures 4.17a, 4.17b, 4.17c, and 4.17d are illustrated in Figures 4.17e, 4.17f, 4.17g, and 4.17h, respectively. Figure 4.18 shows the AR spectrum of data (solid line) and created multi-band-pass filter (dashed line) for the normalized frequency 0.3. By comparing the AR spectrum of data for the normalized frequency 0.3 from all of the above synthetic 2D examples (Figures 4.8, 4.10, 4.12, 4.14, 4.16, and 4.18), it is clear that all of them have successfully identified the location of dominant wavenumbers contained in the original data. Also, it is obvious that there is a large variation in the amplitudes of the peaks in the computed AR spectrum of data but their location on the wavenumber axis are the same.

To continue testing our algorithm, I provide another example (Figures 4.19a and 4.19d) where MWNI was used to reconstruct the normalized frequencies in the range 0.07 to 0.084 (Figures 4.19b and 4.19e). I have chosen this particular frequency band to simulate a situation where MSAR will fail in finding enough prediction filters to reconstruct the data. Figure 4.20 shows the number of prediction filters computed using MSAR for each frequency. The gaps in Figure 4.20 belong to the frequencies for which the MSAR scheme was not able to find a prediction filter. For a frequency component f' falling within a gap, the corresponding

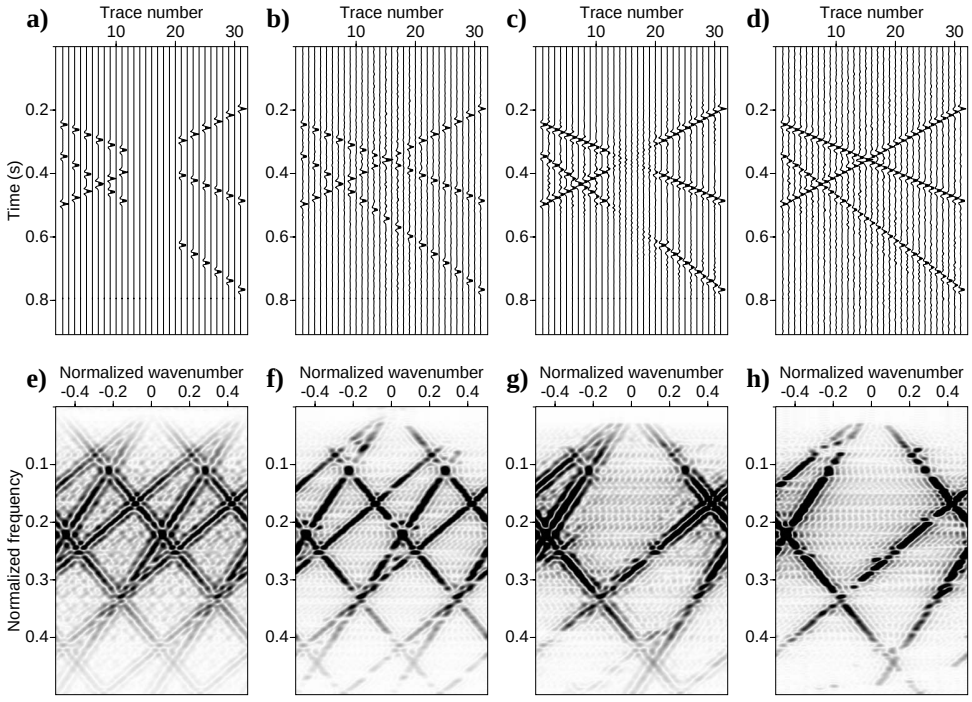

Figure 4.15: 2D synthetic example for testing mixture of decimation and gap. a) Section of missing traces created from original data in Figure 4.5a. b), c), and d) are the reconstruction of a using the MWNI, MSAR-X, and MSAR-K methods, respectively. e), f), g), and h) are the *f-k* panel of a, b, c, and d, respectively.

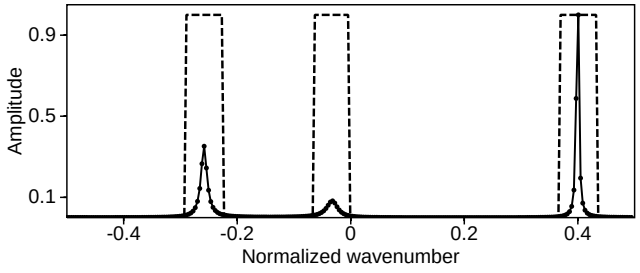

Figure 4.16: AR spectrum of data (Solid line with solid circles) and multi-band-pass filter (dashed line) for the normalized frequency 0.3 used for MSAR-K reconstructed data shown in Figure 4.15d.

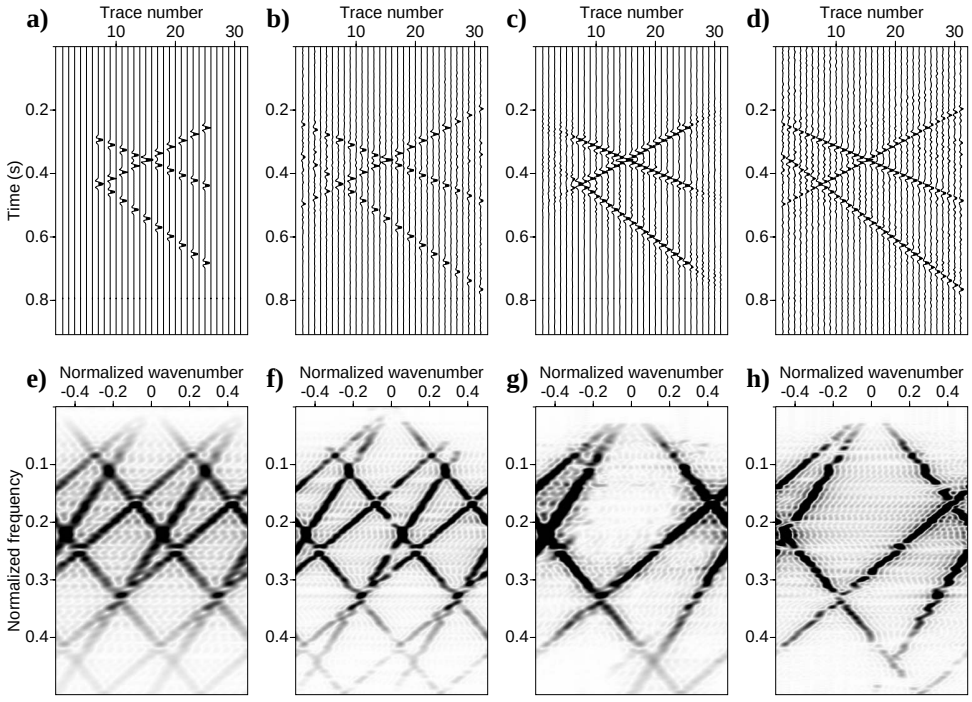

Figure 4.17: 2D synthetic example for testing mixture of decimation and extrapolation. a) Section of missing traces created from original data in Figure 4.5a. b), c), and d) are the reconstruction of a using the MWNI, MSAR-X, and MSAR-K methods, respectively. e), f), g), and h) are the *f-k* panel of (a), (b), (c), and (d), respectively.

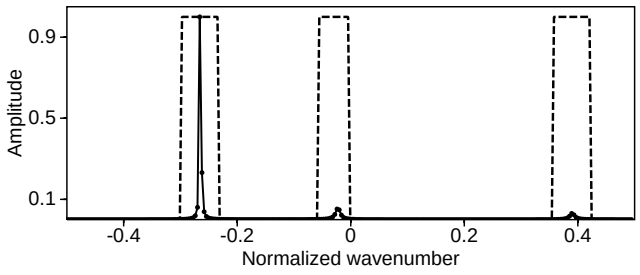

Figure 4.18: The AR spectrum of data (Solid line with solid circles) and multi-band-pass filter (dashed line) for the normalized frequency 0.3 used for MSAR-K reconstructed data shown in Figure 4.17d.

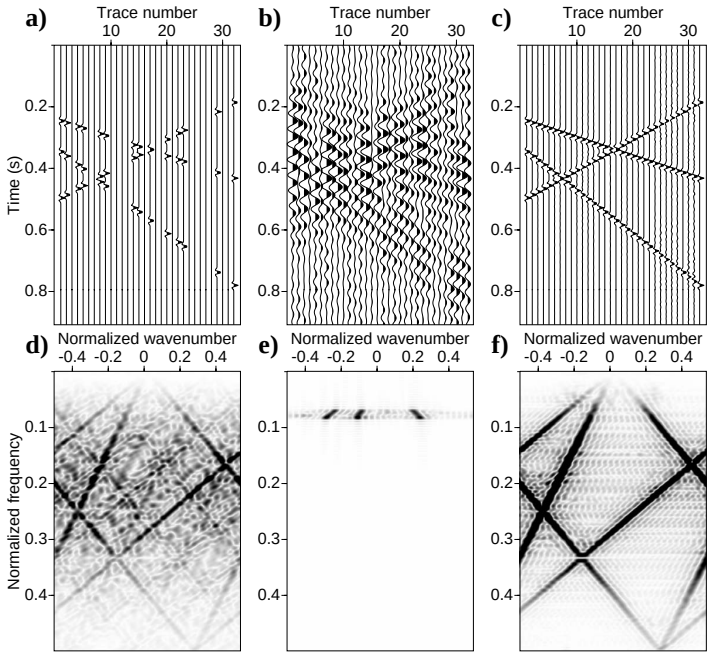

Figure 4.19: A synthetic example with three linear events. a) Data with missing traces. b) Reconstructed low frequencies using MWNI. c) Reconstructed section using MWNI and MSAR-X with extrapolation of prediction filters to avoid gaps in Figure 4.20. d), e), and f) are the *f-k* panels of (a), (b), and (c), respectively.

low frequency component f'/α falls out of the interval $[f_{min_r}, f_{max_r}]$; therefore, there is no way of extracting prediction filters for those frequencies. As I have already mentioned, these missing prediction filters can be estimated using extrapolation methods. In this case, the known prediction filters can be used to find the prediction filters in the gaps using the method provided in Appendix B of Spitz (1991).

In summary, by using MSAR plus extrapolation of prediction filters, I have been able to retrieve all the information necessary to reconstruct the data for all the frequencies. The reconstructed data using MSAR-X and its *f-k* panel are shown in Figures 4.19c and 4.19f, respectively. The result is quite good considering that the prediction filters in the four gaps in Figure 4.20 were computed using estimated prediction filters followed by extrapolation. The situation described in this example is very unlikely to happen. I must stress that the combination of MSAR and Spitz's extrapolation technique (Spitz, 1991), however, do provide a solution to this problem. In addition, it is important to mention that the low frequency end of the data f_{min_r} could be high enough to yield this situation if aggressive low pass filtering was used, for instance, to suppress ground roll.

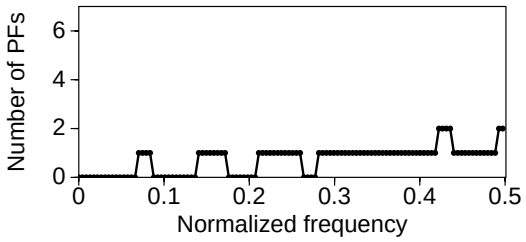

Figure 4.20: The number of prediction filters contributing to each frequency component in Figure 4.19c.

4.3.2 Real data example

In order to test the performance of the MSAR reconstruction on a real data set, I apply the technique to the reconstruction of a near offset section from a marine data set from the Gulf of Mexico. Events arising from diffractions on a salt body make the reconstruction difficult for the MWNI method. About 40 % of the traces were removed from the original section (Figure 4.21a) to simulate a section with missing traces (Figure 4.21b). The section of missing traces is reconstructed using MWNI (Figure 4.21c) and MSAR-X (Figure 4.21d). For this particular test, the length of the prediction filters is set to $L = 4$. From a comparison of these figures (notice in particular the areas indicated by squares) it is easy to see that the combined application of MWNI and MSAR-X produce a result where the steeply dipping events are better preserved after reconstruction. In addition, Figure 4.22 shows the *f-k* representation of the data in Figure 4.21.

The *f-k* domain representation shows that the aliased event is preserved when MSAR-X is used to reconstruct the high frequencies (Figure 4.22d). The MSAR method is based on autoregressive modeling of events in the *f-x* domain. In other words, MSAR can optimally model linear events in the *t-x* domain (complex sinusoids in *f-x*). The aforementioned assumption can be validated by applying MSAR in small spatio-temporal windows where curved events will appear as linear events. One must realize, however, that the scarcity of data in small windows can lead to suboptimal reconstructions.

Figure 4.23 shows the part of data reconstructed using MWNI. Figure 4.24 portrays the number of prediction filters contributing to the reconstruction of each frequency. It must be noted that because there was no signal at normalized frequencies larger than 0.22, the MSAR-X reconstruction was only performed for the normalized frequencies smaller than 0.22.

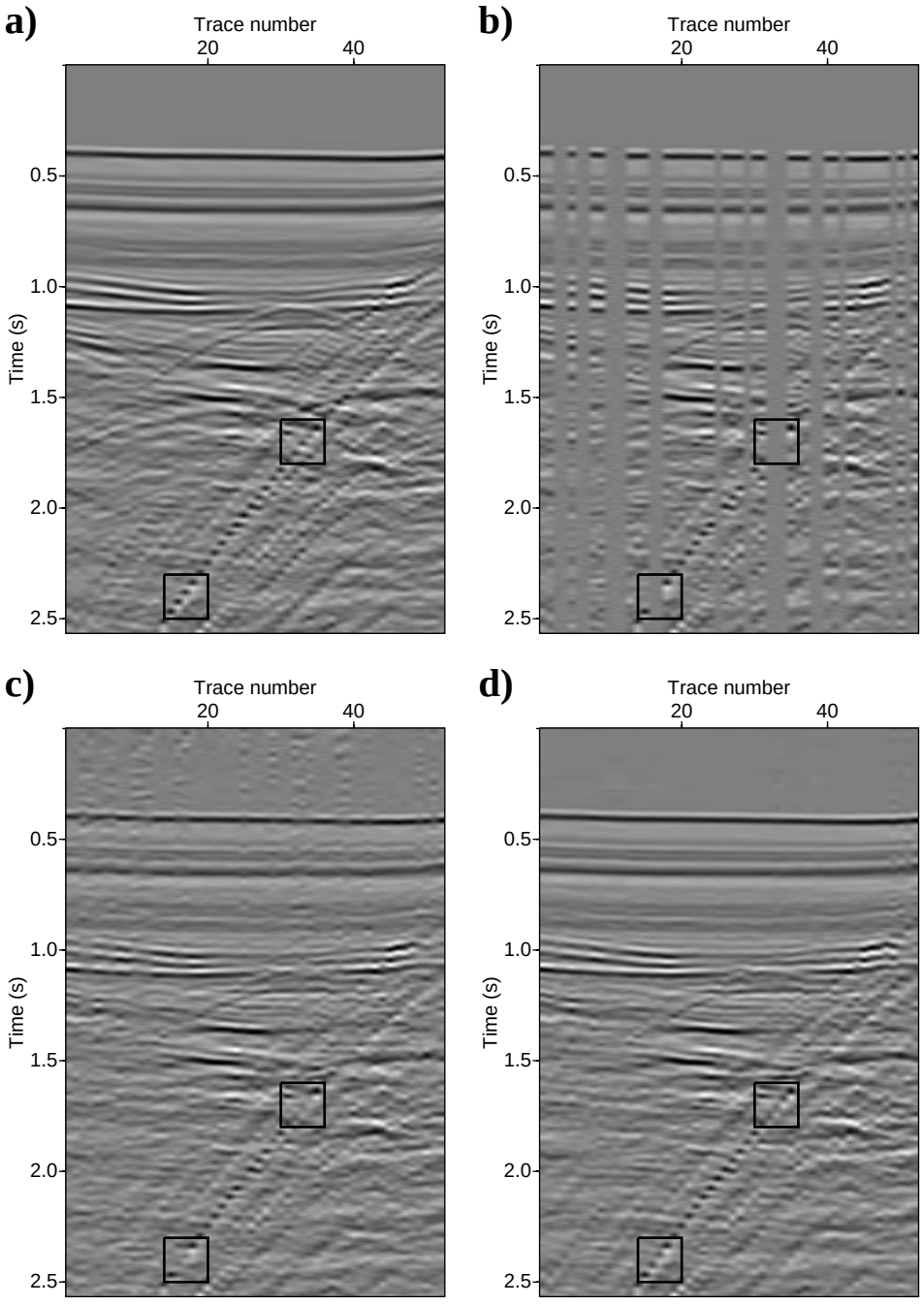

Figure 4.21: Reconstruction of a near offset section. a) Original section. b) Section with missing traces. c) Reconstructed section using MWNI. d) Reconstructed section using MSAR-X. Regions indicated by squares show the advantages of using MSAR-X.

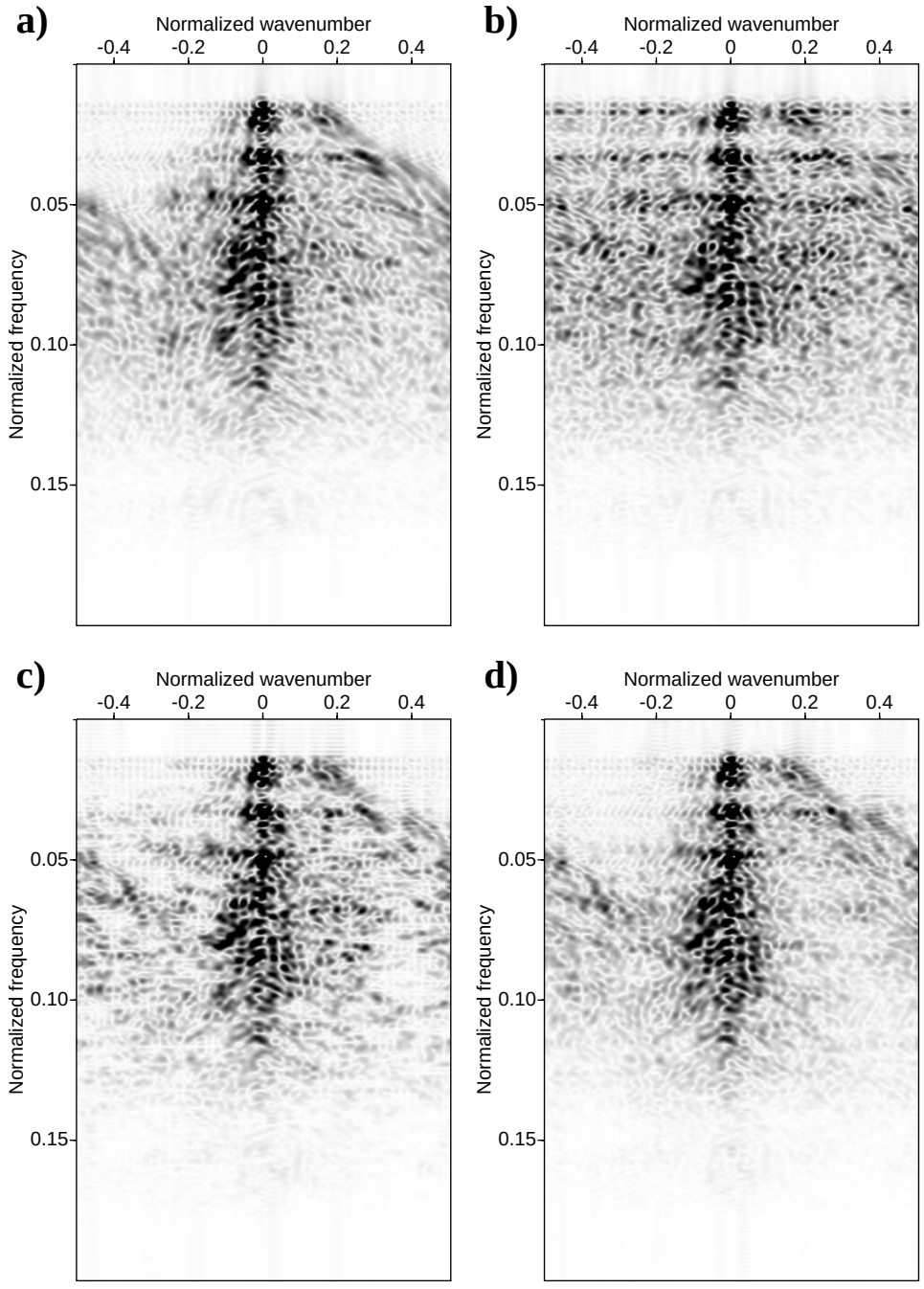

Figure 4.22: The *f-k* domain representation of Figure 4.21. a) Original data. b) The section of missing traces. c) Reconstructed section using MWNI. d) Reconstructed section using MSAR-X.

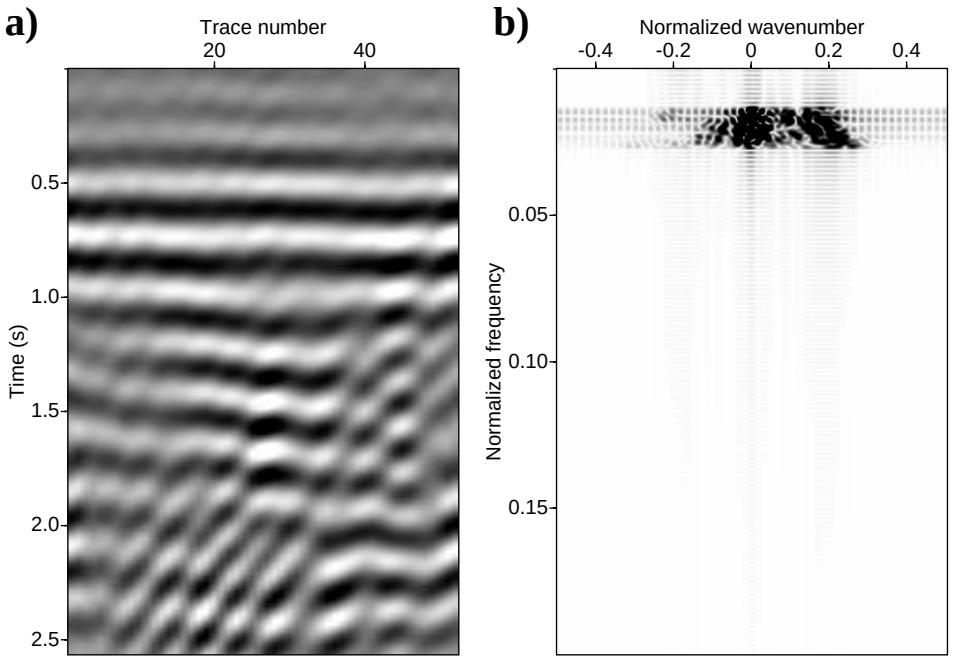

Figure 4.23: The reconstructed part of data using MWNI used to compute prediction filters for Figure 4.21d. a) The data in t-x domain. b) The data in f-k domain.

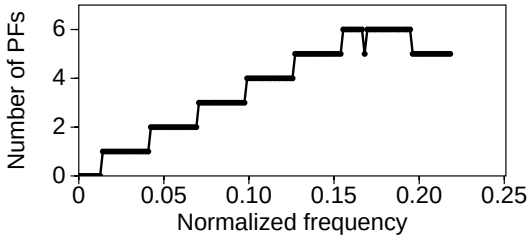

Figure 4.24: The number of prediction filters contributing to each frequency in the reconstruction portrayed in Figure 4.21d.

It should be noticed that I have not yet removed the alias from the data. At this stage I have reconstructed the data (obtained the missing traces). An interpolation technique for regularly sampled data can now be used to de-alias the reconstructed data. Figures 4.25a and 4.25b show the result of first order interpolation of the final reconstructed section (Figure 4.21d) using Gulunay's *f-k* interpolation method (Gulunay, 2003) and MSAR-X, respectively. In the MSAR-X de-aliasing case I have considered that every second trace is missing. Figures 4.25c and 4.25d show the *f-k* representation of the Figures 4.25a and 4.25b, respectively. The aliasing of the diffraction event is removed to a high extent. Gulunay's interpolation shows some low frequency artifacts which could be a by-product of the mask required by the method to annihilate the aliased part of the signals. In this implementation, MSAR-X de-alias uses the average of prediction filters. This is the main difference with Spitz (1991) where only one prediction filter for each frequency component is utilized.

4.4 Discussion

The MSAR method could have potential problems in finding the required low frequency components needed to extract the prediction filter associated to the reconstruction of a given high frequency component. In this case, one can alleviate the problem by zero padding the data before applying the DFT. This reduces the frequency sampling interval; therefore, making it easier for the algorithm to find the required low frequencies. Alternatively, one can use the Wraped Discrete Fourier Transform (WDFT) to densely sample some specific band of frequencies (Franz et al., 2003; Makur and Mitra, 2001).

It is clear that any method based on the assumption of linear events will have problems modeling events with curvature. This facet does not escape the MSAR technique. In fact, departure from the linear event model will introduce non-stationarity in the prediction problem and, consequently, the multi-step algorithm will produce different prediction filters at each step. Claerbout and Fomel (2006) discuss this problem using expanding anti-alias operators in the *t-x* domain. As it has been mentioned by Spitz (1991), one can use small windows as a mean to validate the assumption of linear events. It is well-known that in most geophysical problems there is no free lunch; to solve a problem we require assumptions, in general these assumptions have to be partially violated in order to process real data. A good interpolation method should be robust when the assumptions under which it was designed are not completely honored. Our algorithm is a good example of the latter; it can tolerate mild curvature in the waveforms as portrayed by our real data examples.

Modifying MSAR for multidimensional data reconstruction is straightforward. In higher dimensions, the reconstruction of low frequencies using MWNI should be more stable, and as a result the prediction filters can be calculated with higher precision, leading to a better

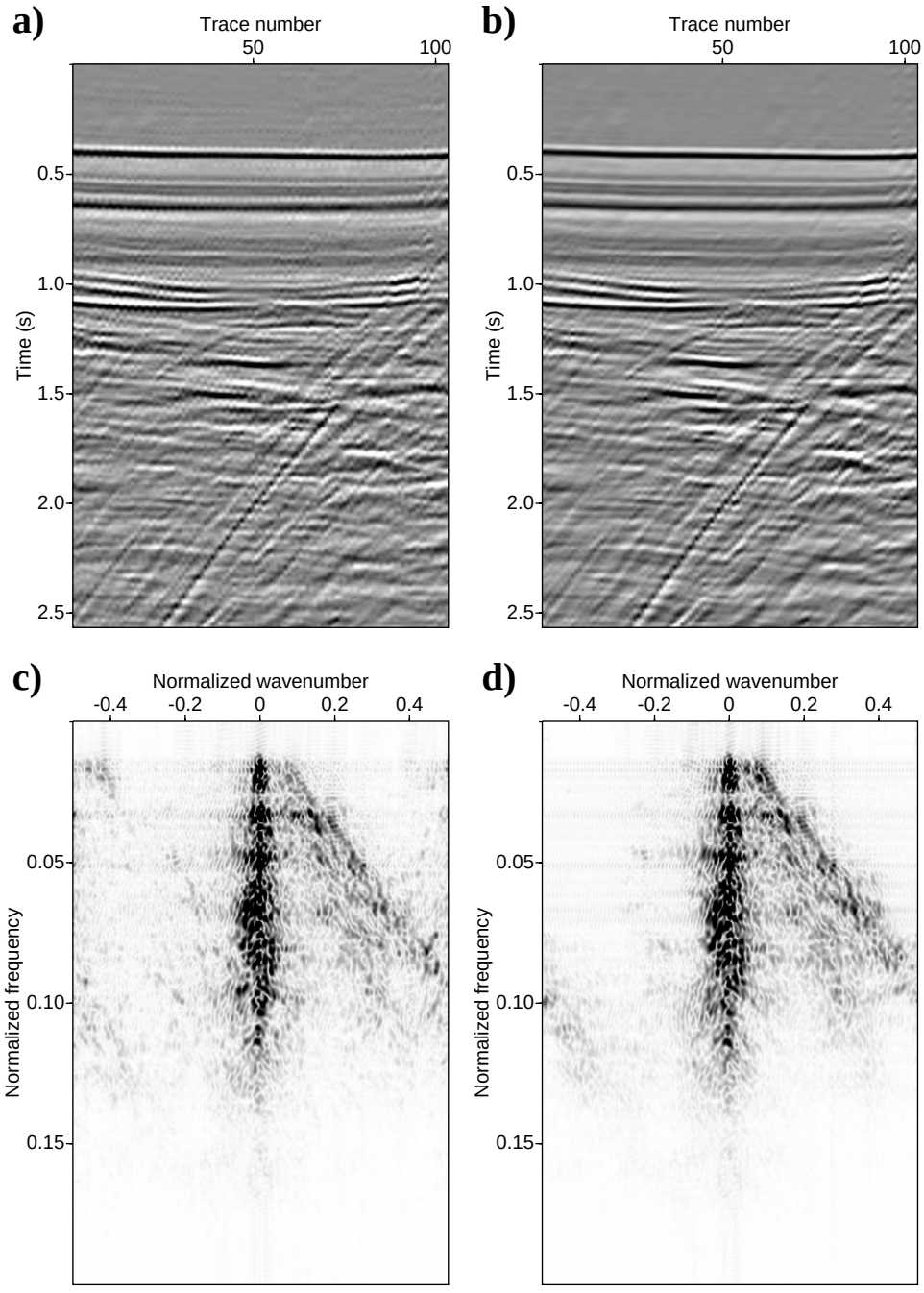

Figure 4.25: First order interpolation of the final reconstructed section in Figure 4.21d. a) Gulunay's *f-k* interpolation b) The MSAR-X method. c) The *f-k* panel of a. d) The *f-k* panel of b.

reconstruction of high frequencies. Spitz (1990) and Wang (2002) introduced a way to use 2D AR operators to interpolate the regularly sampled data in the *f-x-y* domain. The same style of AR operators can be used to extend the MSAR method to the 2D case. The multidimensional application of the MSAR method is discussed in Chapter 5.

4.5 Conclusions

In this Chapter a method for spatial reconstruction of high frequency data components was introduced. The method entails the cooperative application of a Fourier-based technique (MWNI) to reconstruct the unaliased part of the data and a Multi-Step Auto-Regressive (MSAR) algorithm to reconstruct the high frequency and potentially aliased part of the data. The MSAR algorithm relies on extracting information from low frequencies (reconstructed via MWNI) to reconstruct high frequencies. Since the prediction filter of a given high frequency can be computed from more than one reconstructed low frequency, more than one prediction filter can be extracted for a given high frequency. In this case, an average of prediction filters is used to reconstruct the data. This averaging scheme helps eliminate any potential noise contamination from corrupted frequency components. The reconstruction of missing data samples using extracted prediction filters can be either performed in the *f-x* domain (MSAR-X) or *f-k* domain (MSAR-K).

The results of synthetic data reconstruction show that the MSAR-K method is more stable than MWNI and MSAR-X. In addition, it is capable of eliminating various sampling artifacts. The MSAR method can be applied to regularly sampled sections as well. A real data example of interpolation of a regularly sampled section shows that MSAR can effectively be adopted to de-alias seismic data.

CHAPTER 5

Multidimensional MSAR reconstruction

5.1 Introduction

In Chapter 4 the multistep autoregressive (MSAR) method was introduced for the 1D spatial interpolation problem in the f-x domain with nonuniformly sampled data. In this chapter I discuss the implementation of the MSAR method to the data with more than one spatial dimension[1]. Figure 5.1 show a schematic view of how MSAR is applied for 3D data . First a patch of data from original data is selected and the FFT is applied in the time direction. Then each slice of frequencies are reconstructed into a regular grid using MSAR (or MWNI). Finally the reconstructed patch of data is put back in its corresponding location in the final regularized data. The patches of data are selected with overlaps to guarantee smooth reconstruction on the sides of the patches. In this chapter, synthetic and real data examples are used to examine the performance of the proposed method. Also, field data are used to illustrate the applicability of multidimensional MSAR for regularization of seismic data.

5.2 Theory

5.2.1 Multidimensional Fourier Reconstruction of low frequencies

As it was explained in Chapter 4, the first step of applying MSAR reconstruction is to reconstruct the low frequency portion of data into a regular grid. For the 1D case we adopted

[1]Parts of this Chapter was published in Naghizadeh and Sacchi (2009d)

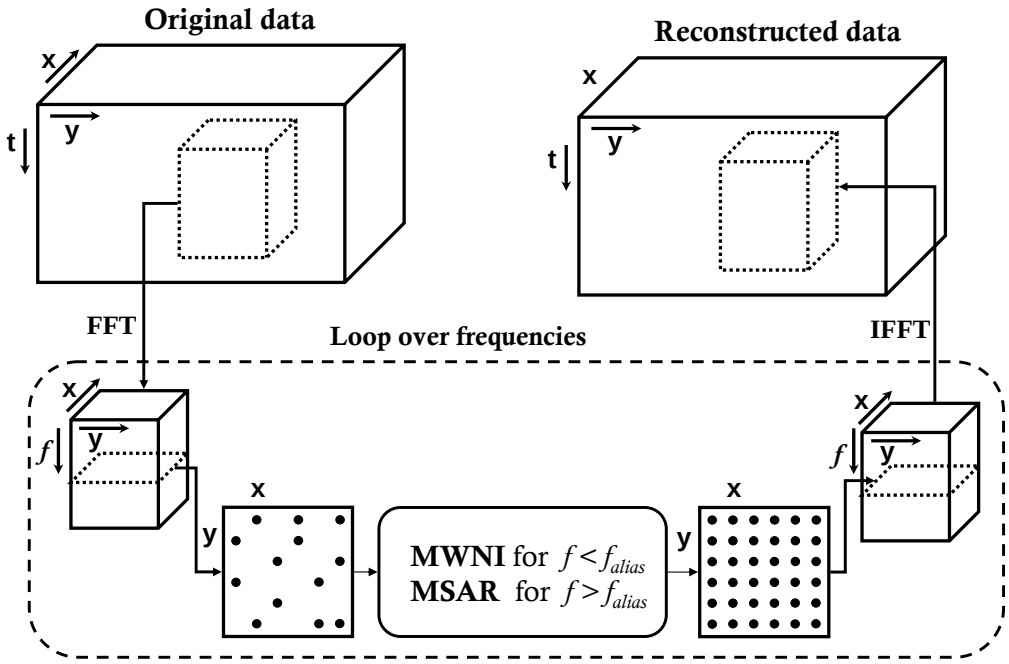

Figure 5.1: Schematic representation for reconstruction of multidimensional data using MWNI or MSAR.

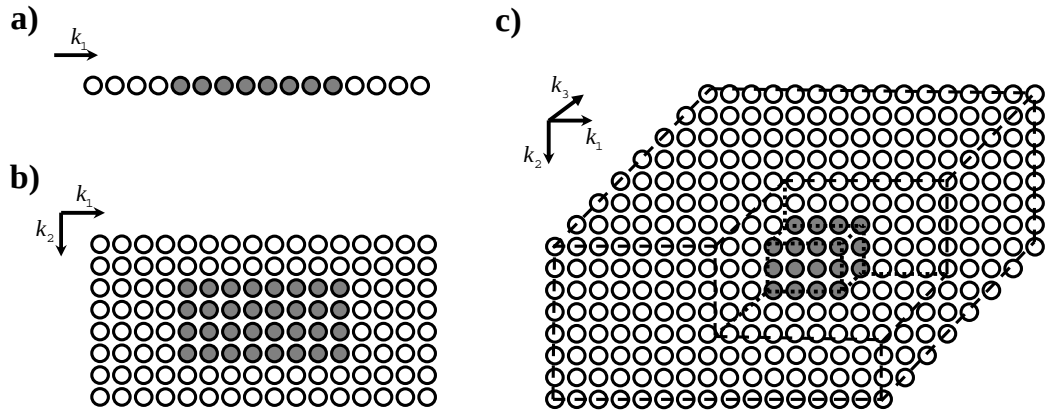

Figure 5.2: Schematic view of a band-limited signal for a) 1D spatial data, b) 2D spatial data and c) 3D spatial data. Grey squares indicate the region of support in the wave number domain.

the minimum weighted norm interpolation (MWNI) method (Liu and Sacchi, 2004; Sacchi and Liu, 2005; Liu et al., 2004). The main operator used in the MWNI method is the Fast Fourier Transform (FFT). There are efficient algorithms for multidimensional application of FFT. By a simple lexicographic ordering of multidimensional data, the flowchart provided in Algorithm 3.1 are used for the multidimensional data (Naghizadeh and Sacchi, 2009b).

Figure 5.2a depicts a schematic representation of the wavenumber domain of a 1D frequency slice. In this case 50% of the samples are unknown. Moving to a 2D frequency slice (Figure 5.2b), and considering a signal that is half band-limited in both k_x and k_y, the number of unknowns are reduced to 25%. The number of unknowns for a 3D frequency slice which is half band-limited in all three wavenumber axes is reduced to 12.5% (Figure 5.2c). Hence, increasing the dimensions of the data can result in more data than unknowns in the *f-k* domain, therefore allowing reconstruction of more sparsely sampled data. This is particularly true when we consider spatial data at low frequencies.

The 1D convolution required for smoothing the spectrum in the MWNI method should be replaced with a multidimensional convolution. Multidimensional convolution (as well as multidimensional autoregressive estimation) can be performed using a 1D convolution algorithm if the original data are properly zero-padded and lexicographically ordered prior to the operation. Appendix B provides the details of such an operation. By utilizing the operators which work with lexicographically ordered data, there is no need to provide new algorithms as the dimension of data increases. This means one set of algorithms provided for the 1D case can be used for the multidimensional situation without any change. I used this strategy throughout my research in order to develop computer programs for the MWNI and MSAR methods.

5.2.2 MSAR estimation of multidimensional prediction filters

Once the low frequencies are reconstructed into a regular grid, the MSAR algorithm can be used to compute the prediction filters. The proof of multidimensional MSAR operation is provided in Appendix A. Here, we expand the theory and formula for the 2D case but generalization to the multidimensional case is straightforward. Let us consider the reconstructed 2D spatial data, $\mathbf{x}$, in the bandwidth $f \in [f_{min_r}, f_{max_r}]$. The components of a 2D prediction filter, P_{m_1,m_2} with order $M_1 \times M_2$, can be estimated using (forward MSAR prediction):

$$x_{n_1,n_2}(f) = \sum_{m_1=0}^{M_1} \sum_{\substack{m_2=0 \text{ if } m_1 \neq 0 \\ m_2=1 \text{ if } m_1=0}}^{M_2} P_{m_1,m_2}(\alpha f) x_{n_1-\alpha m_1, n_2-\alpha m_2}(f), \tag{5.1}$$

where, $\mathbf{x}$ is the 2D data with size $N_1 \times N_2$. The 2D prediction filter can also be estimated using backward prediction:

$$x^*_{n_1,n_2}(f) = \sum_{m_1=0}^{M_1} \sum_{\substack{m_2=0 \text{ if } m_1 \neq 0 \\ m_2=1 \text{ if } m_1=0}}^{M_2} P_{m_1,m_2}(\alpha f) x^*_{n_1+\alpha m_1, n_2+\alpha m_2}(f), \tag{5.2}$$

where $*$ indicates complex conjugate. The solution of an augmented linear system of equations composed of expressions 5.1 and 5.2 gives the estimated 2D prediction filters. Notice that $P_{0,0}$ is considered equal to 1 for the above formulas. Here, forward (equation (5.1)) and backward (equation (5.2)) 2D autoregressive equations are computed by α steps. The parameter $\alpha = 1, 2, ..., \alpha_{max}$ is the step factor used to extract the prediction filter for frequency αf from frequency f.

Figure 5.3 depicts how to find 2D prediction filters for 2D data. The prediction filters operate from the upper-left to lower-right corner. It is also possible to use filters that operate from the upper-right to lower-left corner. The overall strategy is similar to the 1D case. For this specific example, the size of data is equal to 12×12 and the size of the prediction filters is equal to 3×3, and, according to equation (A.6), the maximum α step is equal to 5.

5.2.3 Data reconstruction using multidimensional prediction filters

Once the multidimensional prediction filters of each single frequency are estimated, the next step is to utilize them to reconstruct the missing data samples. Again, one can use prediction filters in the *f-x* domain (MSAR-X). This would follow exactly the same strategy of reordering known and unknown data samples which was explained for the 1D case in Chapter 4. The other alternative is to use the multidimensional AR spectrum of data

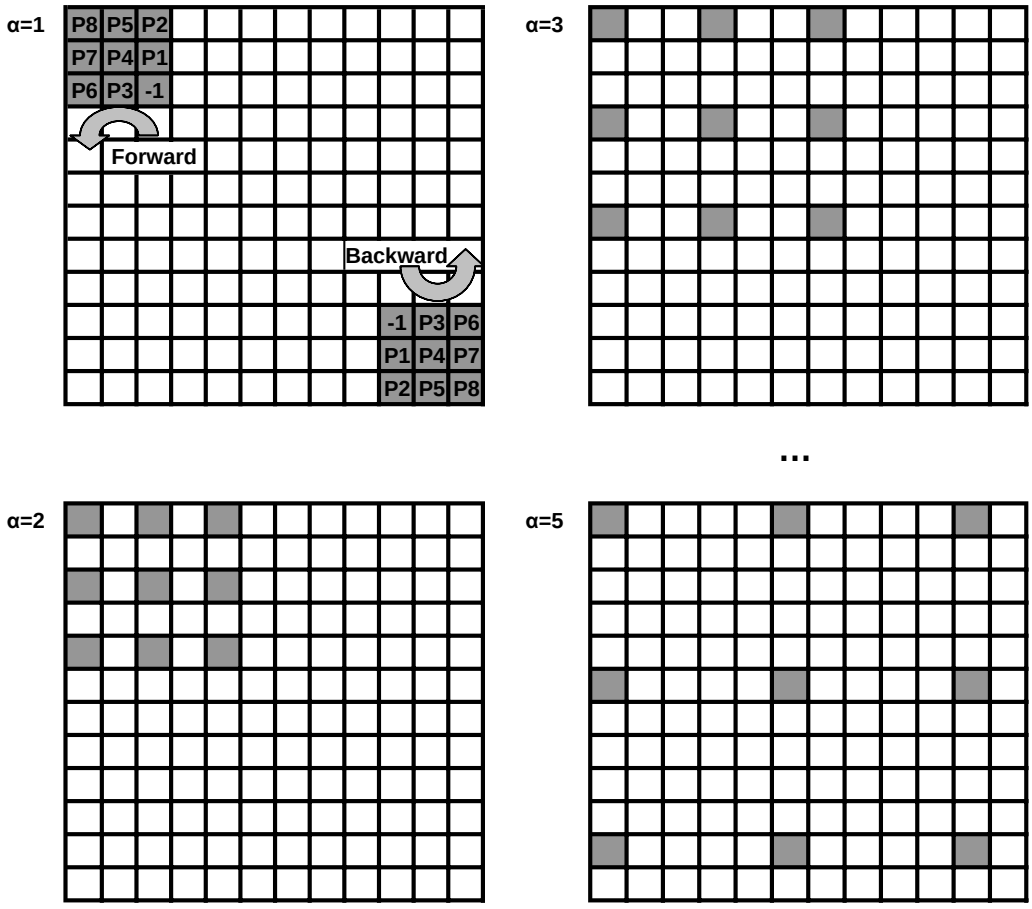

Figure 5.3: Flowchart for the 2D MSAR algorithm. The gray squares show 2D prediction filters. To avoid notational clutter, notice that backward prediction is only shown for $\alpha = 1$ and ignored for higher α steps.

in the *f-k* domain (MSAR-K). It is important to mention that while using MSAR-X is possible, it is computationally very expensive for the multidimensional data. Therefore, for the multidimensional reconstruction of seismic data it is desirable to always use MSAR-K instead of MSAR-X. Any reference to the MSAR method in this chapter indicates using MSAR-K for reconstructing missing samples.

The 2D AR spectrum of data, $S_{pf}(\kappa_1, \kappa_2)$, can be obtained by:

$$S_{pf}(\kappa_1, \kappa_2) = \frac{\sigma_\epsilon^2}{\left|1 - \sum_{m_1=0}^{M_1} \sum_{\substack{m_2=0 \text{ if } m_1 \neq 0 \\ m_2=1 \text{ if } m_1=0}}^{M_2} P_{m_1,m_2} e^{-i2\pi m_1 \kappa_1} e^{-i2\pi m_2 \kappa_2}\right|^2}, \quad -\frac{1}{2} \leq \kappa_1, \kappa_2 \leq \frac{1}{2} \tag{5.3}$$

where κ_1 and κ_2 are the normalized frequencies (wavenumbers) in each direction and σ_ϵ^2 is the noise variance. Generalizing equation (5.3) for higher dimensions is straightforward.

The next step is to build a 2D multi-band-pass filter by identifying the peak locations of the 2D spectrum of prediction filter. In the 2D case the AR spectrum of data is a smooth surface on a 2D grid. A sample is considered a peak location if it is bigger than all of the 8 samples surrounding it. By placing a 2D box function with proper size at the location of identified peaks a 2D multi-band-pass filter (Υ_{pf}) is created which can be used inside a multidimensional Fourier reconstruction method to recover the missing sample.

5.3 Examples

5.3.1 2D synthetic example

The first example considers the application of MSAR for a 2D spatial synthetic data set. The original data are composed of three dipping planes which are aliased in both spatial directions. The data set contains 1200 traces in a spatial grid of 30×40. Figure 5.4a shows a perspective view of the cube containing the synthetic data. The trajectory lines of the three synthetic events can be seen on the boundaries of the cube. Figure 5.4b shows the cube of the original data in which the top view is a time slice at 0.65 (s), the front view is the 21st slice in the Y direction, and the side view is the 17th slice in the X direction. Figure 5.4c shows the *f-k* panel of the data from the front view in Figure 5.4b, or the 21st Y slice of data. This *f-k* representation shows the presence of aliasing as well as random noise in the original data. A data set with missing traces was simulated by first eliminating every other X slices of data and later randomly eliminating 75% of the remaining traces (Figure 5.5a). Therefore, 87.5% of the original traces were eliminated. Figure 5.5b shows the *f-k* panel of missing data from 21st Y slice (Front view of Figure 5.5a).

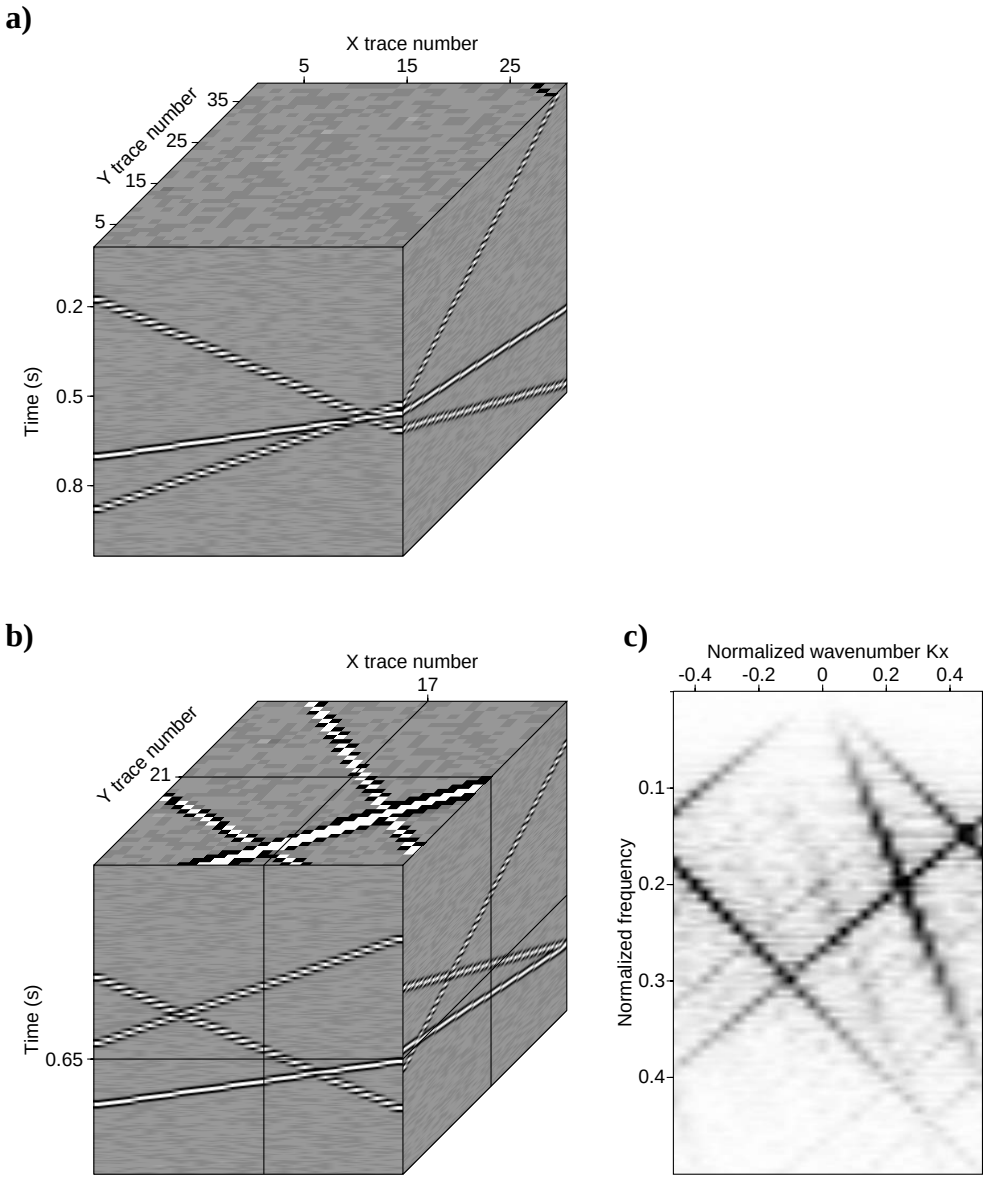

Figure 5.4: a) Original cube of data. b) Slices view from inside the original data. c) The *f-k* representation of the first Y slice of the original data (front view of cube in (b)).

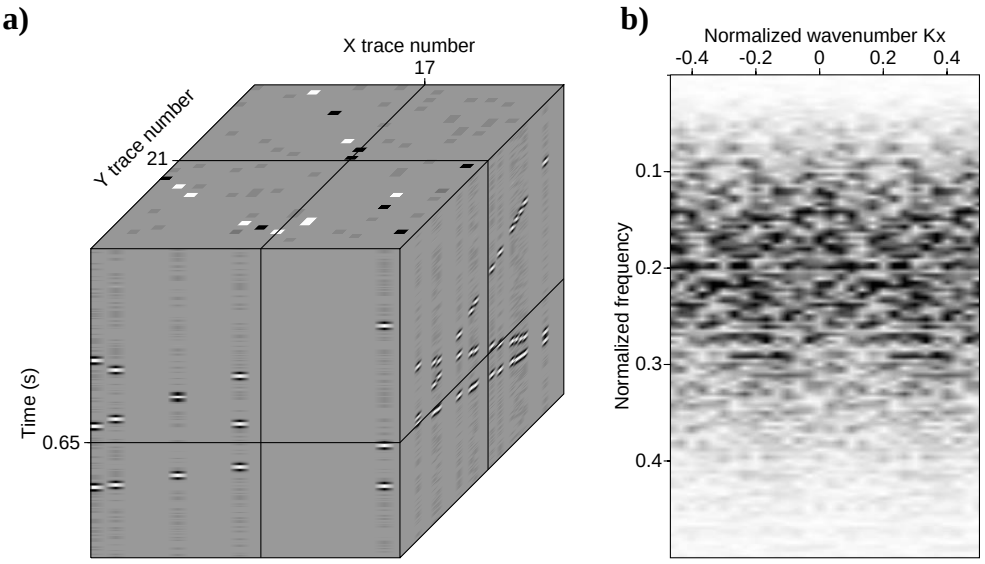

Figure 5.5: a) Slice view of missing data. About 87.5% of traces are missing due to combinational regular and random sampling. b) The *f-k* representation of the first Y slice of the original data (front view of cube in (a)).

Figure 5.6a shows the result of the MWNI method applied to all frequencies. For 3D data, MWNI has to be implemented for each 2D frequency slice of $x - y$ data. Figure 5.6b shows the *f-k* panel of the front view of Figure 5.6a. Applying Fourier reconstruction for all of the frequencies was only able to recover the randomly eliminated traces not the decimated X slices.

In order to apply 2D MSAR reconstruction, the low frequency portion of data from the normalized frequency 0.03 to 0.07 is reconstructed using the MWNI method. Figure 5.7a shows the reconstructed low frequency data. Figure 5.7b shows the *f-k* panel of the front view of Figure 5.7a. In the *f-x-y* domain 2D prediction filters, with size 3×3, can be extracted for all the frequencies using the MSAR routine. The computed prediction filters can be used to estimate the values of missing traces in the spatial direction using either MSAR-X or MSAR-K methods. Figure 5.8a shows the reconstructed data using the MSAR-X algorithm. The *f-k* panel of data in the front view of Figure 5.8a (21st Y slice) is shown in Figure 5.8b. Due to the high percentage of missing data samples MSAR-X fails to recover the original data.

Figure 5.9a shows the reconstructed data using the MSAR-K algorithm. The *f-k* panel of data in the front view of Figure 5.9a (21st Y slice) is shown in Figure 5.9b. The MSAR-K reconstruction results were successful and do not suffer from the shortcomings of the MWNI

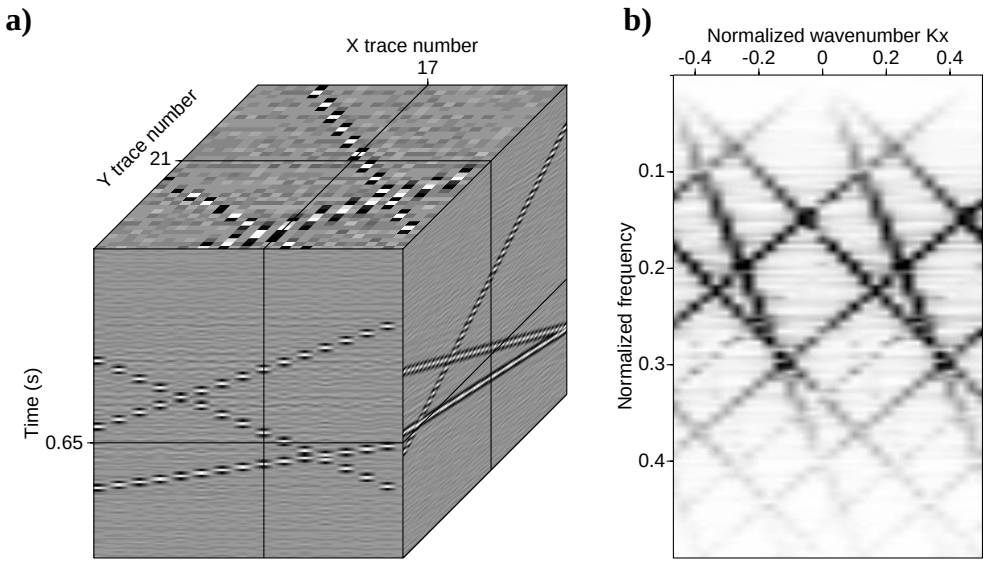

Figure 5.6: a) slice view of the cube of data reconstructed using MWNI. b) The *f-k* representation of the first Y slice of the original data (front view of cube in (a)).

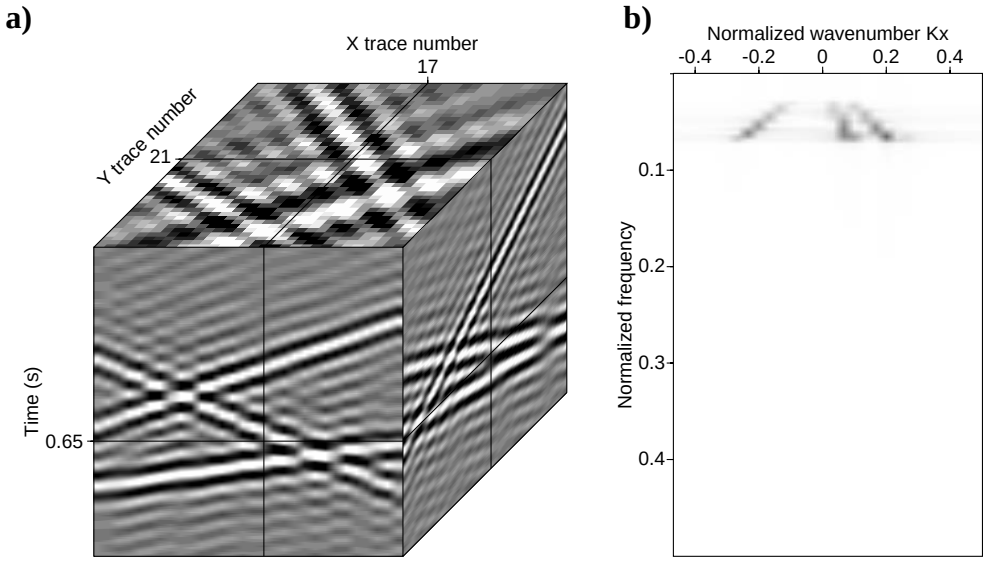

Figure 5.7: a) slice view of the cube of low frequency reconstructed data using MWNI. b) The *f-k* representation of the first Y slice of the original data (front view of cube in (a)).

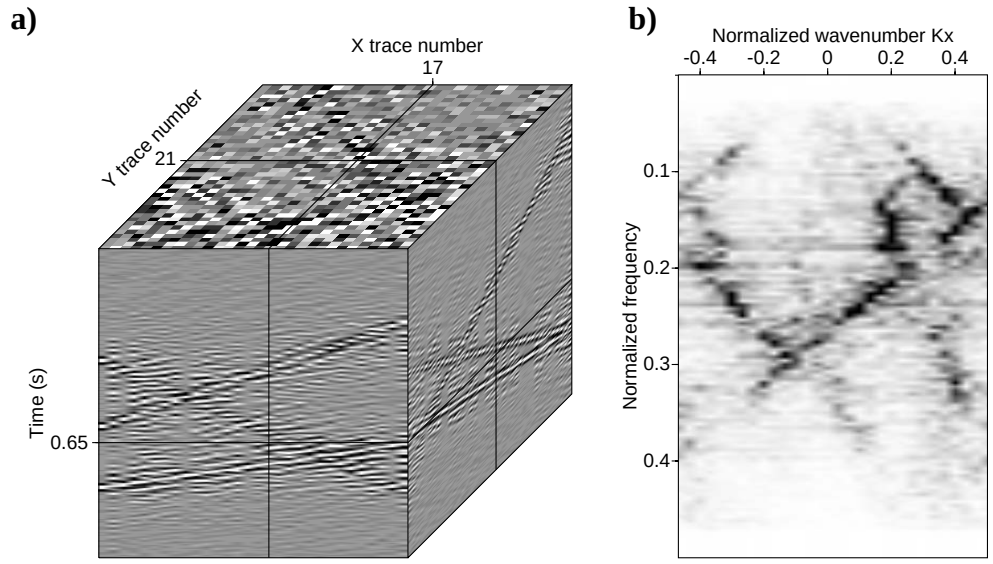

Figure 5.8: a) slice view of the cube of data reconstructed using MSAR-X. b) The *f-k* representation of the first Y slice of the original data (front view of cube in (a)).

and MSAR-X methods. Figure 5.10a shows the 2D AR spectrum of data for the normalized frequency 0.3. Figure 5.10b illustrates the 2D multi-band-pass filter built based on the peak locations of Figure 5.10a. This multi-band-pass filter is used to locate the wavenumbers contained in the original data for normalized frequency 0.3.

5.3.2 2D real data example

To continue testing the MSAR algorithm in higher dimensions, I chose a real data set from the Gulf of Mexico. I picked 19 consecutive shots each with 91 traces, each trace having 900 time samples. I randomly eliminated 60% of the traces to test our algorithm. Figure 5.11 shows the location of available and missing traces. The filled circles indicate the available traces and the cross signs represent missing traces. The data contain several curved events. The MWNI and MSAR method are based on the assumption of linear events (Liu and Sacchi, 2004; Naghizadeh and Sacchi, 2007). In order to satisfy the linearity condition one has to partition data into small windows and reconstruct them separately. It is also important to choose a proper amount of overlap between windows to avoid discontinuities. Here, we reconstruct the data by choosing 19 shots and 20 offsets for each window. Also the next window of offsets is chosen in a way to have 10 overlapped offsets with the previous window.

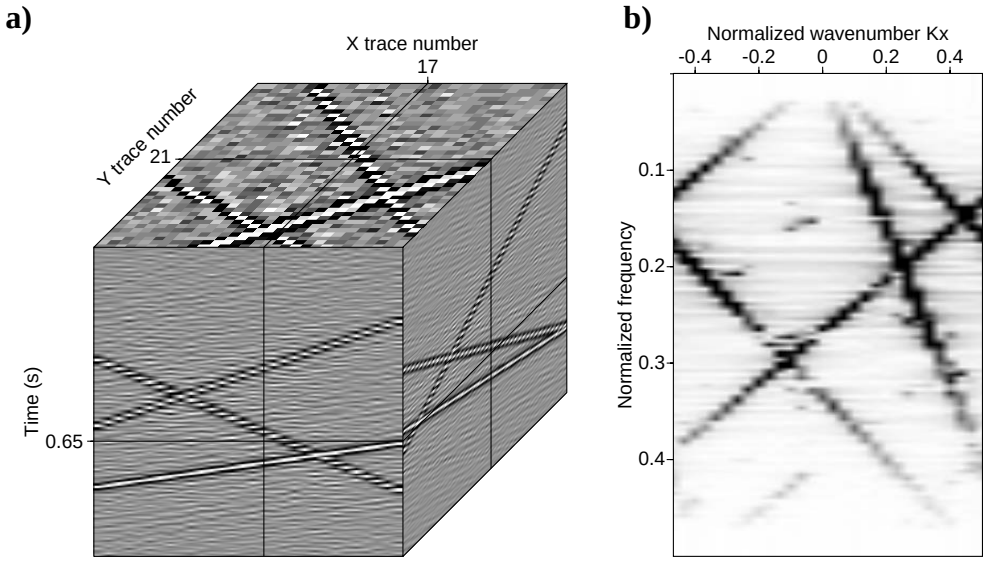

Figure 5.9: a) slice view of the cube of data reconstructed using MSAR-K. b) The *f-k* representation of the first Y slice of the original data (front view of cube in (a)).

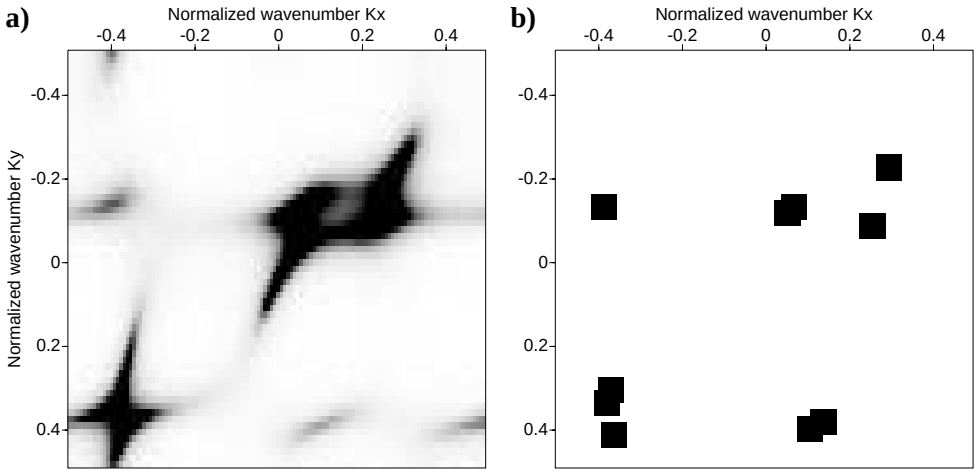

Figure 5.10: a) 2D AR spectrum of data for the normalized frequency 0.3 used for MSAR-K reconstruction in Figure 5.9f. b) 2D multi-band-pass filter built based on the peak location of function in (a).

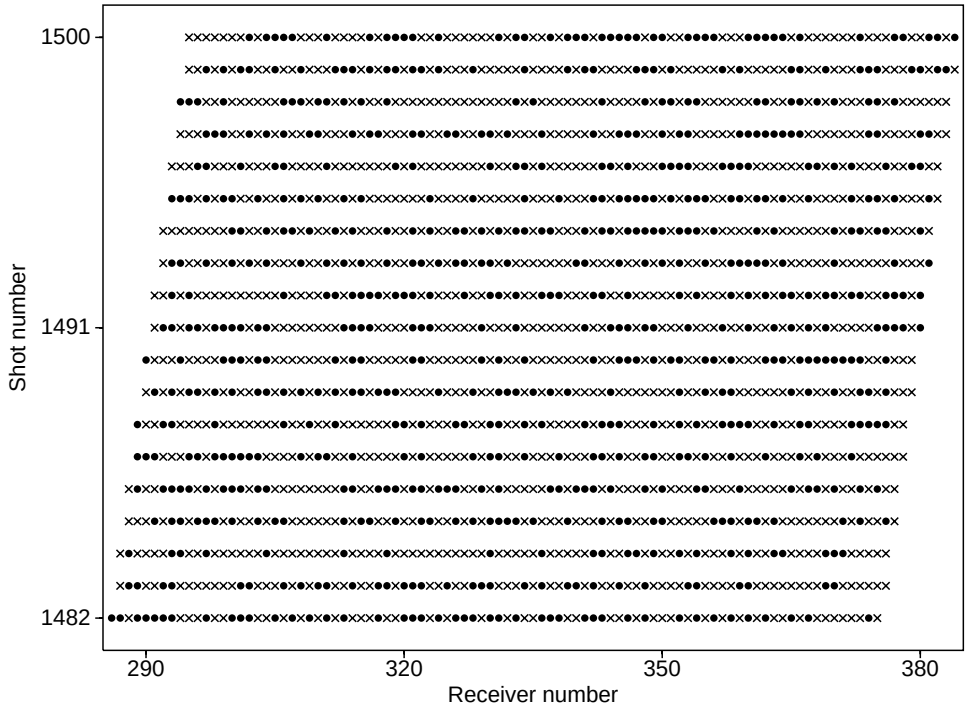

Figure 5.11: Source and receiver positions of the data from the Gulf of Mexico. Filled circles show the available traces and crosses show the missing traces.

Windowing in the time direction can also improve results. The latter was not used for this particular example.

A window of data from shots 1488 to 1491 and from time 3.0 (s) to 5.0 (s) is chosen for illustration purposes. Figures 5.12a and 5.12b show the original and missing data. Figure 5.13a shows the reconstructed low frequency portion of the data using MWNI. I used the MSAR algorithm to extract 4×3 size 2D prediction filters for all frequencies. Finally, the missing traces were reconstructed using the 2D AR spectrum of data in the f-k domain. Figure 5.13b shows the result of the MSAR-K reconstruction. The 2D AR spectrum of data was able to utilize the low frequency information to reconstruct the high frequencies. Figure 5.14 shows the difference section for Shot 1488 of Figure 5.12a. The illustrated sections in Figure 5.14, from 1 to 4, are the original, missing, MSAR-K reconstructed, and difference sections (3 minus 1), respectively.

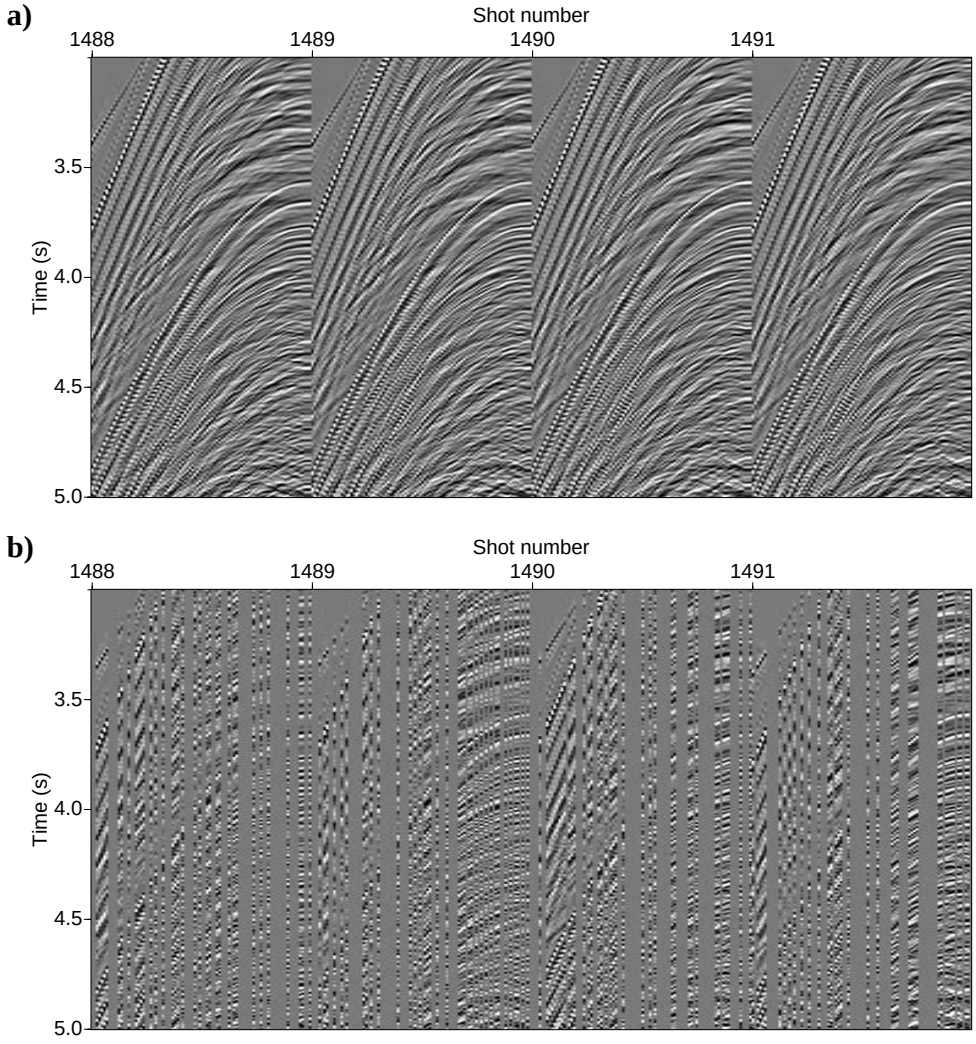

Figure 5.12: A window of data from the Gulf of Mexico. a) Original data. b) Data with 60% randomly missing traces.

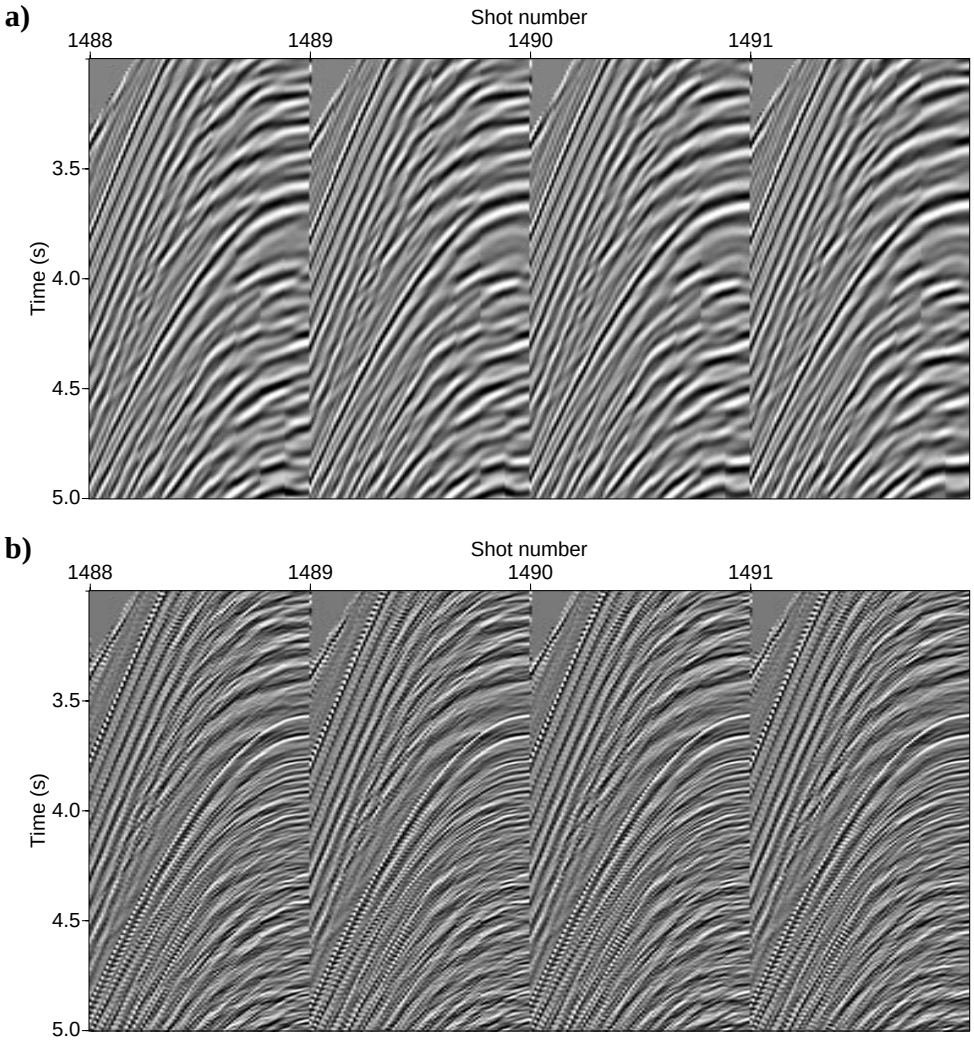

Figure 5.13: a) A window of the low frequency reconstructed data from the Gulf of Mexico using MWNI. b) Reconstructed data using 2D MSAR-K algorithm.

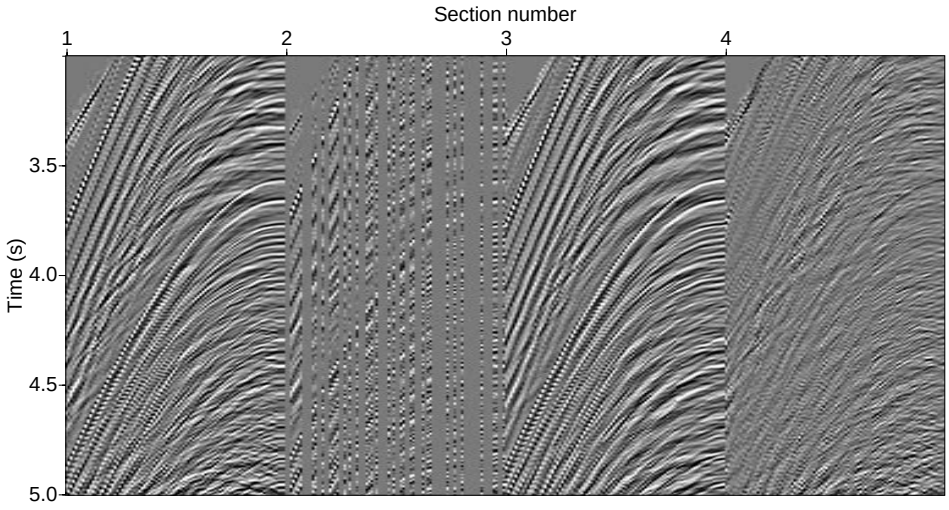

Figure 5.14: Comparison of the original and MSAR-K reconstructed data for Shot Record 1488 of Figure 5.12a. The sections from 1 to 4, are the original data, incomplete data , MSAR-K reconstructed data, and the difference sections (3 minus 1), respectively.

5.3.3 3D real data example

Next, I applied the MSAR-K reconstruction to a 3D real seismic data set. The data were obtained from a 3D survey carried out in the Erskine (Alberta, Canada) area in the Western Canadian Sedimentary Basin. The complete data set contains 150 in-lines and 40 cross-lines with offsets aligned along the in-line direction. Figure 5.15a shows the sparse and irregular sampling geometry in the cross-line and offset for in-line 45. In order to examine the performance of 3D MSAR algorithm, I chose a window from in-lines 35 to 50, cross-lines 4 to 34 and offsets from 50 to 2000 (m) with 50 (m) offset increments. Therefore, the final spatial grid size is $16 \times 31 \times 40$ (in-line, cross-line, offset). The data show little curvature in the offset direction. Therefore, I reconstructed the data without windowing in the offset direction. Figure 5.15b shows the spatial positions after MSAR reconstruction. Figure 5.16a shows the original CMPs for in-line 45 and cross-lines 5-9. The low frequency portion of data are reconstructed using MWNI; the results are shown in Figure 5.16b. Using the MSAR algorithm, $3 \times 3 \times 3$ size 3D prediction filters are extracted for all frequencies. Next the missing data were reconstructed using 3D AR spectrum of data. The MSAR-K reconstruction is shown in Figure 5.16c. The results show the same events, but now with the addition of the higher frequencies. Figure 5.17a shows an original common offset section of the Erskine data at 200 (m). The data are depicted for in-lines 40-49 and for each in-

line the correspondent cross-line section has been shown. Figures 5.17b and 5.17c show the reconstruction of low frequencies using MWNI and the reconstruction of all frequencies using the MSAR-K method, respectively. In Figure 5.17c one can identify continuous dipping events in the cross-line direction. These events were difficult to visualize in the original data (Figure 5.17a).

5.3.4 3D Land data acquired over a salt body

The last example is a 3D real land data acquired in a area where a salt body arises close to the surface. The survey is composed of several cross-lines and in-lines. The data have irregular distributions of offsets and azimuths. The final aim was reconstructing offsets and azimuths into a regular grid in order to get a better image of shallow reflectors. The data are reconstructed using 2D, 3D, and 4D implementations of MWNI and MSAR-K methods. Table 5.1 give a brief overview of the parameters used in reconstruction of the land data. More detailed explanation of the adopted procedures is provided in the following sections.

Table 5.1: Parameters used for reconstruction of land data examples

Example	2D	3D	4D
Spatial dimensions	X-line,Offset	In-line,X-line,Offset	In-line,X-line,Offset,Azimuth
Size of patches	30×80	$15 \times 15 \times 15$	$12 \times 12 \times 12 \times 12$
Overlaps	10×0	$5 \times 5 \times 0$	$5 \times 5 \times 0 \times 0$
Size of AR operator	3×3	$3 \times 3 \times 3$	$3 \times 3 \times 3 \times 3$

2D spatial reconstruction

For the 2D MSAR application, I used cross-lines of only one in-line and regular grid of 80 offsets with an offset interval equal to 25 (m). Also a single azimuth is assumed for all of the offsets to limit the spatial domain of data only to cross-lines and offset. In the case of more than one trace for a specific offset only one trace was kept. The reconstruction is applied using small spatial windows of cross-lines (30 cross-lines) to honor the linearity assumption and overlaps of cross-lines (10 overlapped of cross-lines) to honor the continuity of events. The original data were Normal Move-Out (NMO) corrected and therefore one would expect to have flat events in the offset direction. Figure 5.18 shows the original data after stacking all offsets. Figure 5.19 shows the result of reconstruction of all frequencies using MWNI. Figure 5.20 shows the result of reconstruction of low frequency portion of data using MWNI. The MSAR algorithm uses 3×3 size prediction filters extracted from low frequencies and extended to all frequencies. Figure 5.21 shows the result of MSAR-K reconstruction algorithm. It is clear that both MSAR-K and MWNI methods have partially

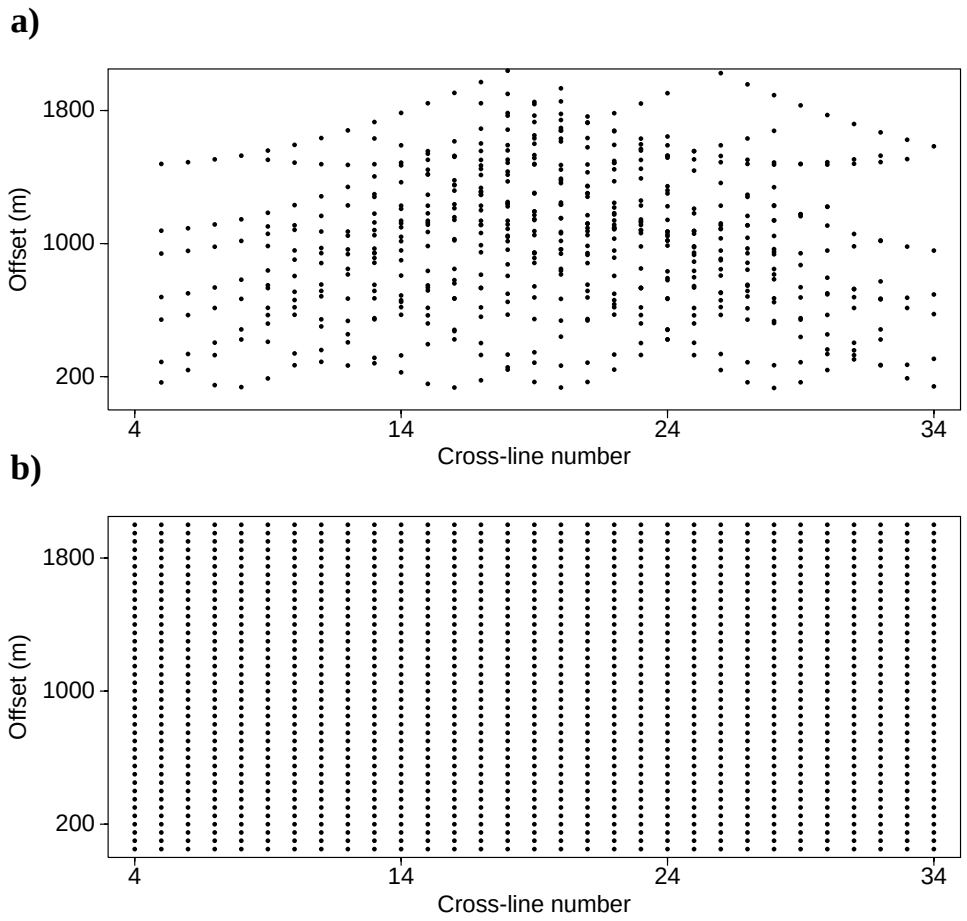

Figure 5.15: a) The location chart of the original available traces for in-line 45 of the Erskine data set. b) The location chart of traces after reconstruction using the MSAR-K method.

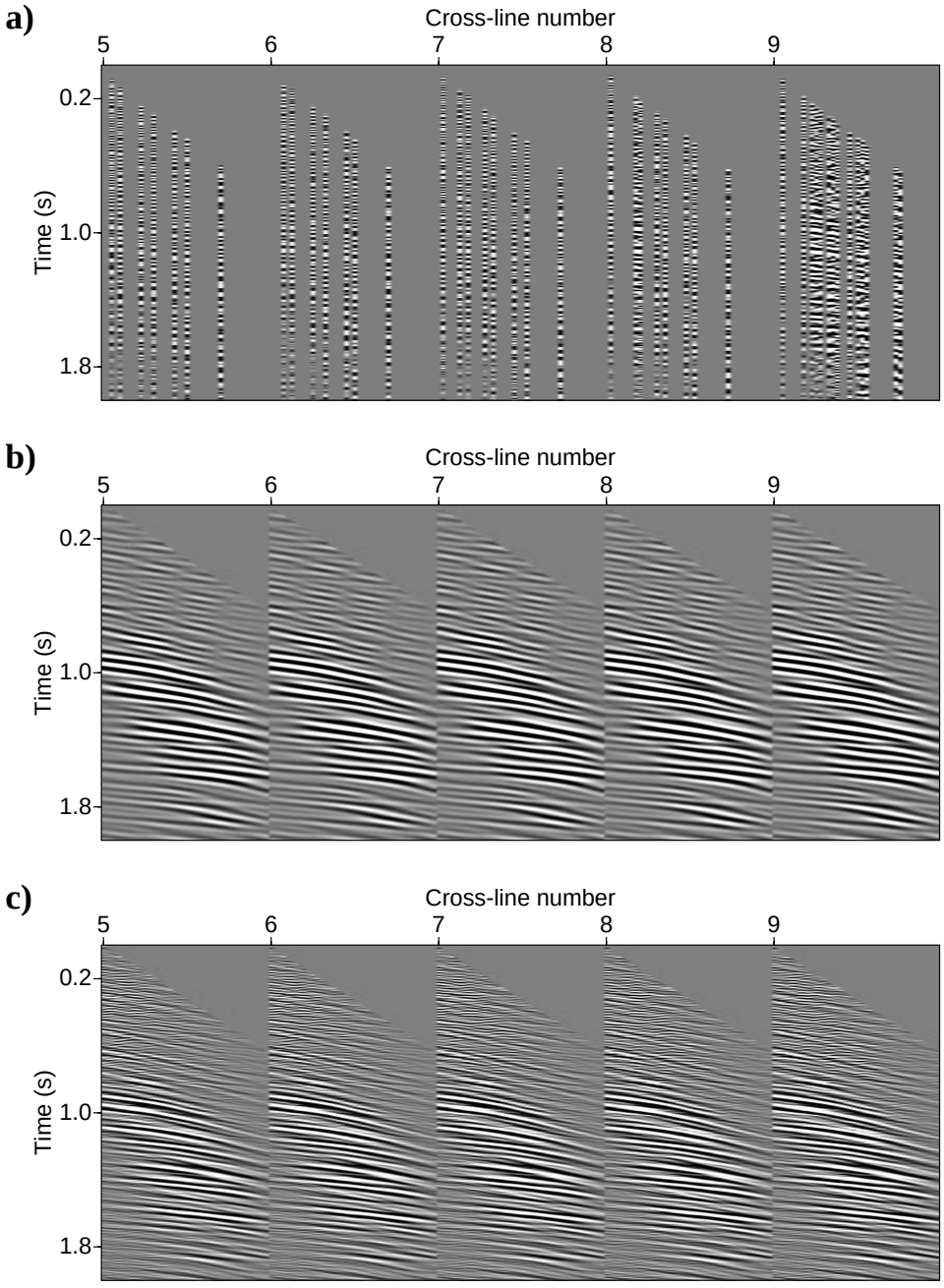

Figure 5.16: a) Original data from in-line 45 and cross-lines 5-9 of the Erskine data set. b) Reconstructed low frequency portion estimated via MWNI. c) Reconstructed data using the MSAR-K method.

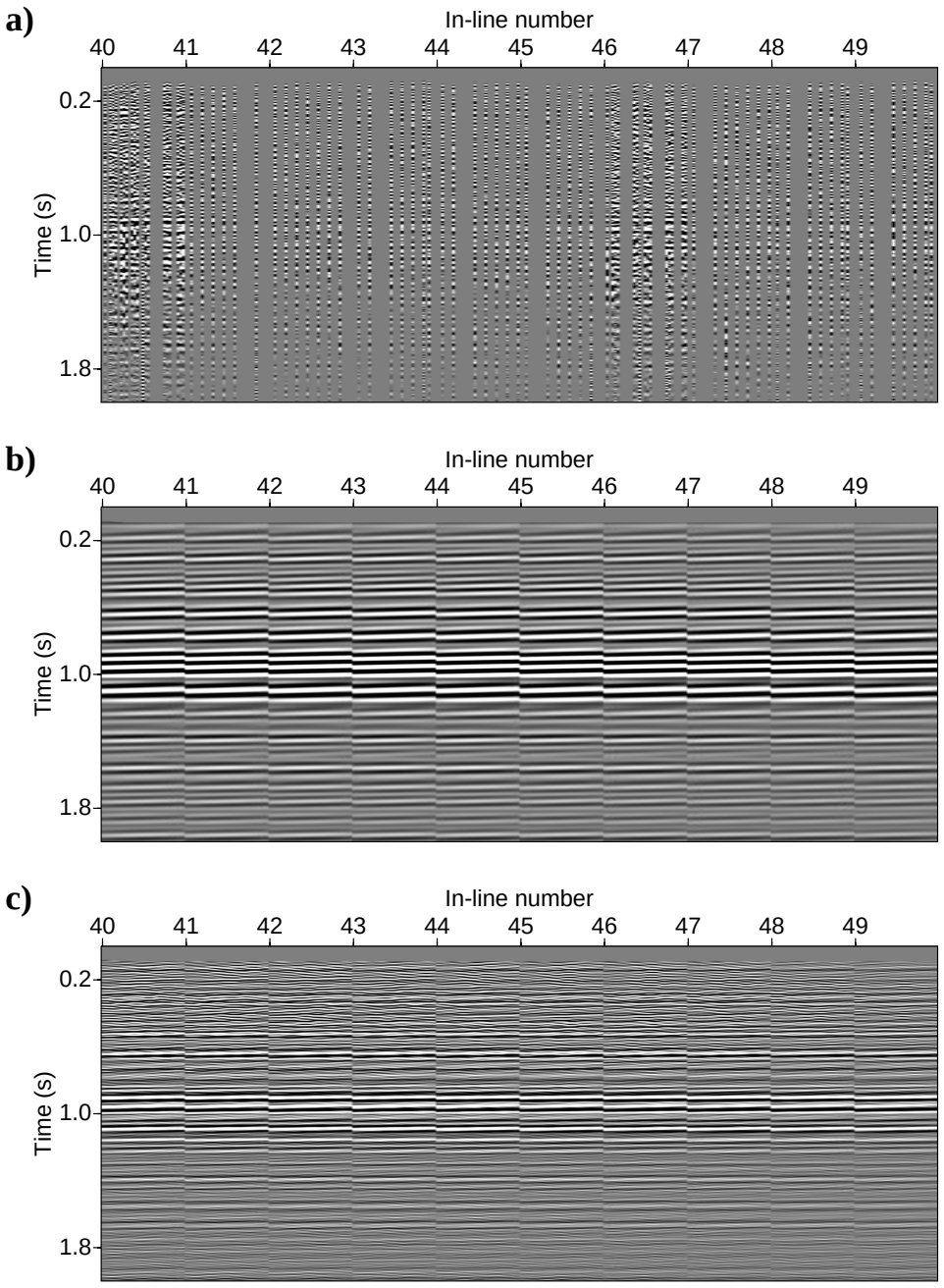

Figure 5.17: a) Original common offset data at offset 200 m from the Erskine data set (in-lines 40-49). b) Reconstructed low frequency portion using MWNI. c) Reconstructed data using the MSAR-K method.

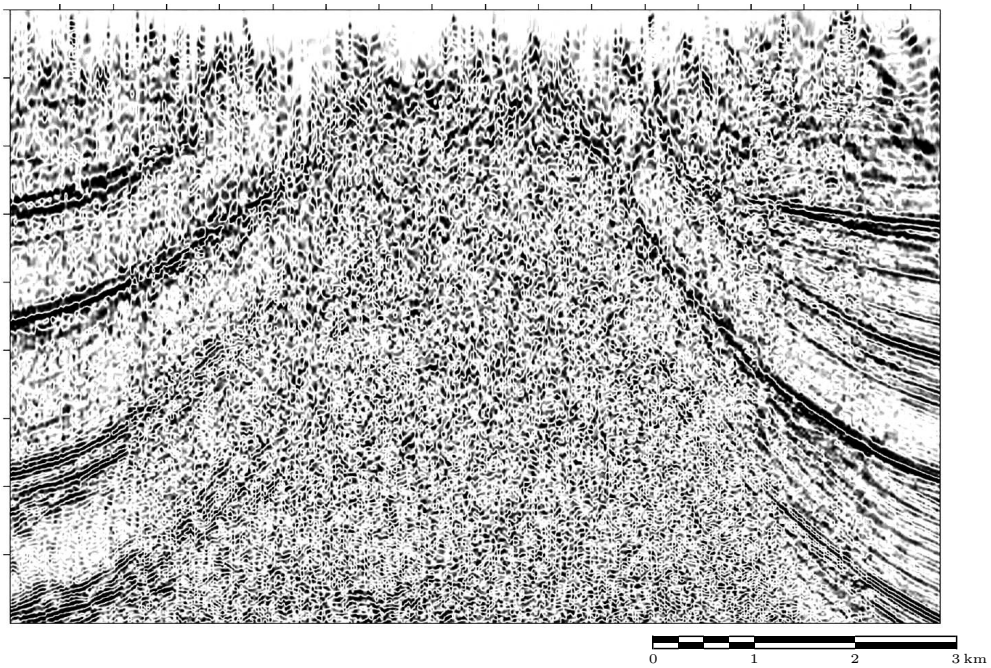

Figure 5.18: A stacked in-line section of the original land data.

eliminated random noise from data. In addition, it is now possible to identify shallow reflectors on the top of salt body (Figures 5.19 and 5.21).

3D spatial reconstruction

To continue the analysis, I created a 3D spatial volume consisting of traces in the in-line and cross-line directions and offset. The size of spatial windows in this case was 15 cross-lines, 15 in-lines, and 15 offsets with 100 (m) offset interval. The next window of data are picked with 5 overlaps in both cross-line and in-line directions. Figure 5.22 shows the original data in one of the in-lines after stacking all offsets. Figures 5.23 and 5.24 show , respectively, the reconstruction of all frequencies and low frequency portion of data using MWNI method for the same inline position in Figure 5.22. The reconstructed low frequency data in Figure 5.24 were used by MSAR algorithm to extract 3D prediction filters of size $3 \times 3 \times 3$ for all the frequencies. Figure 5.25 shows the results of MSAR-K reconstruction method.

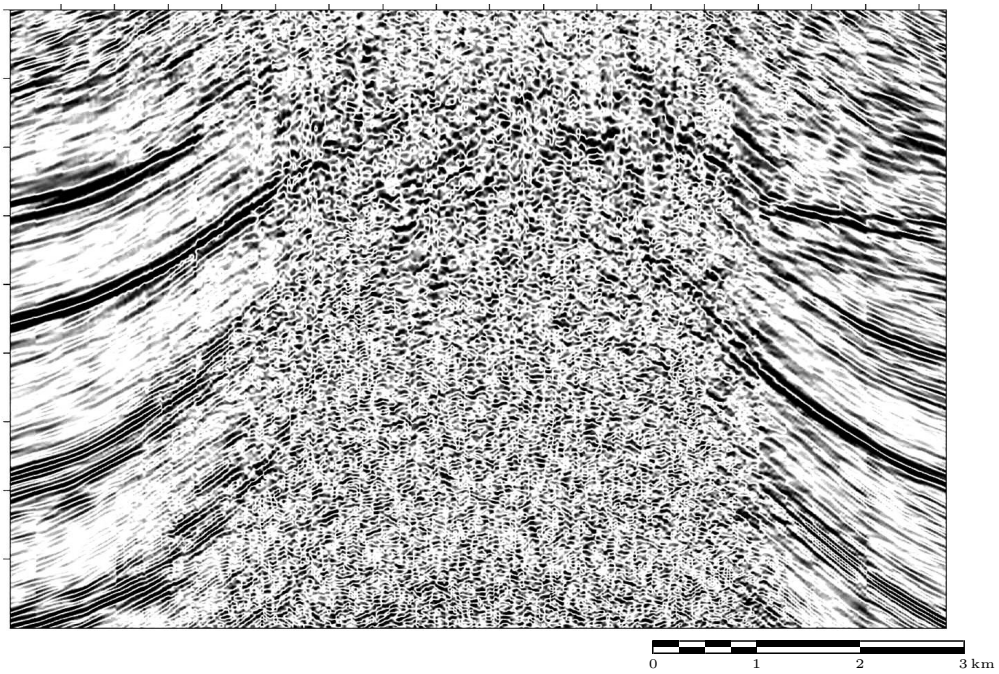

Figure 5.19: 2D MWNI interpolation of data in Figure 5.18.

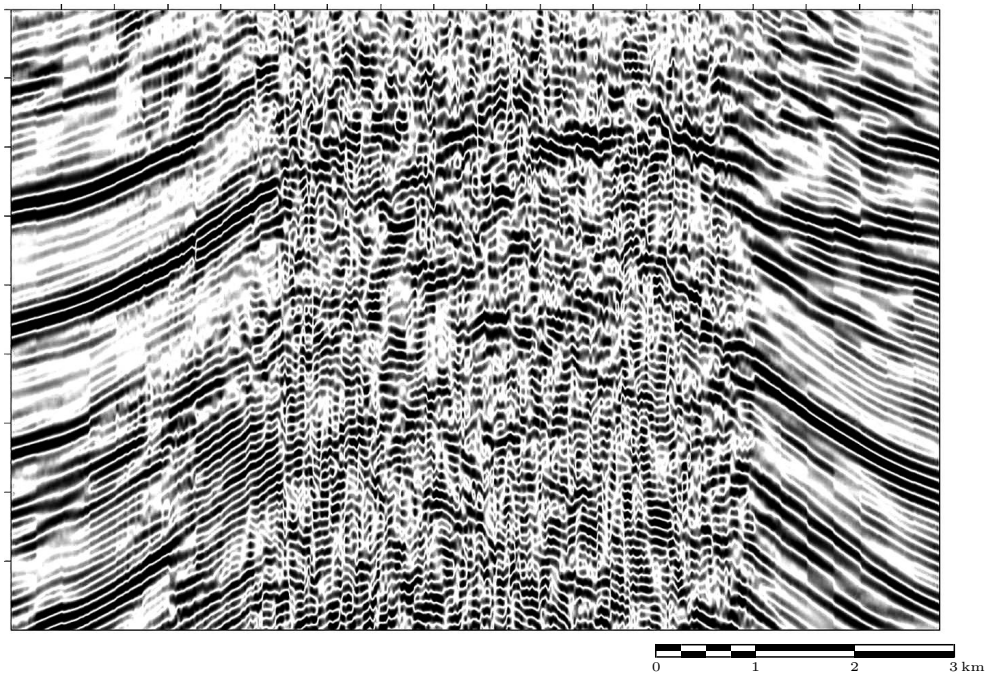

Figure 5.20: Low frequency reconstructed data in Figure 5.18 using 2D MWNI.

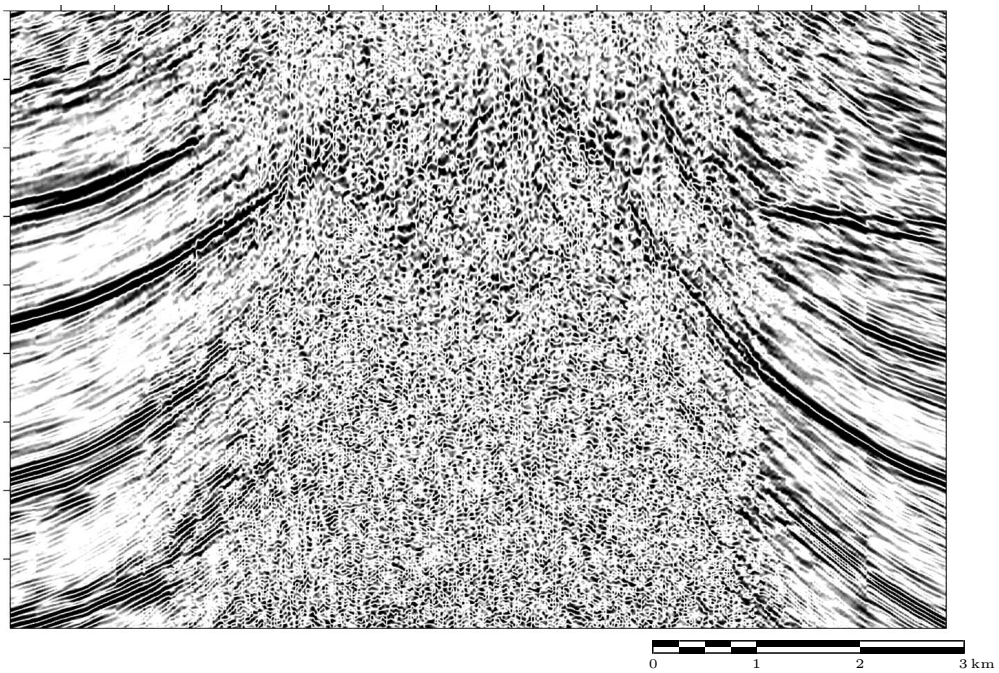

Figure 5.21: 2D MSAR interpolation of data in Figure 5.18.

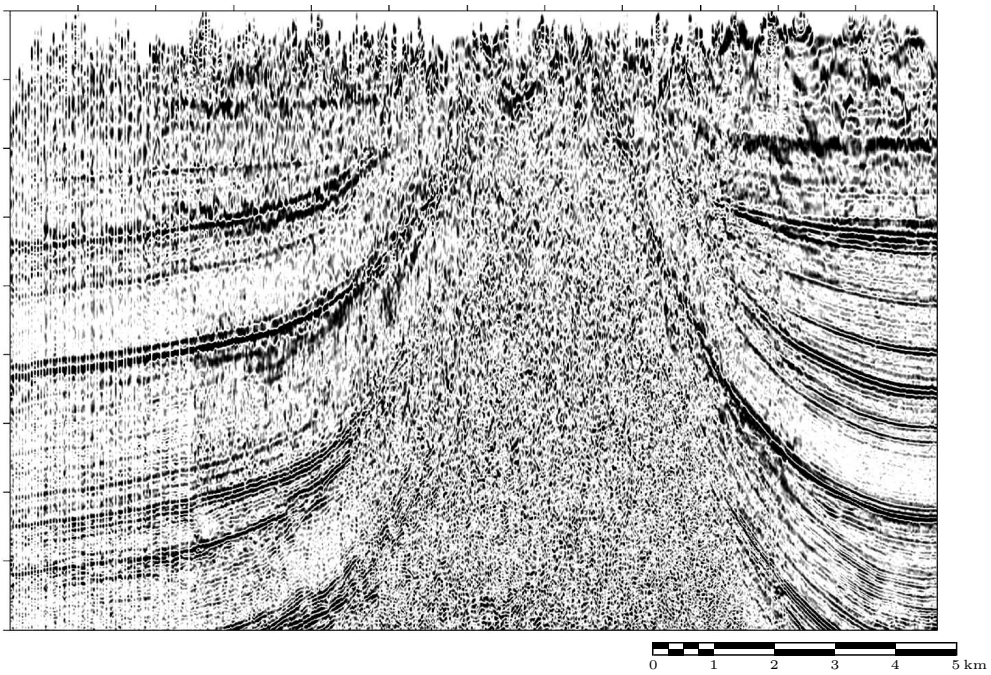

Figure 5.22: A stacked in-line section of the original land data.

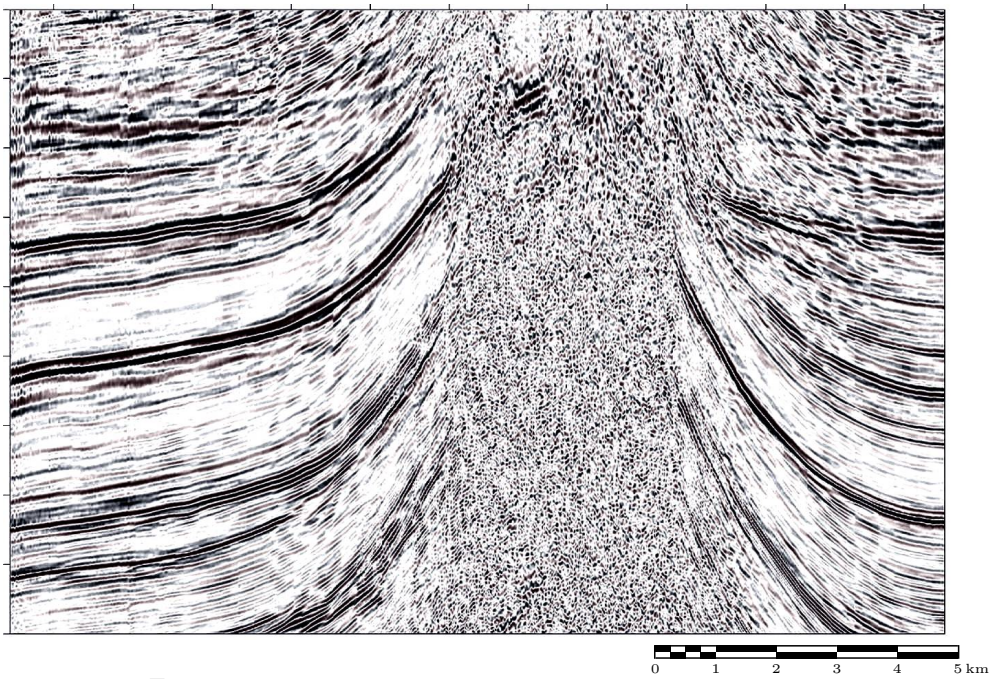

Figure 5.23: 3D MWNI interpolation of data in Figure 5.22.

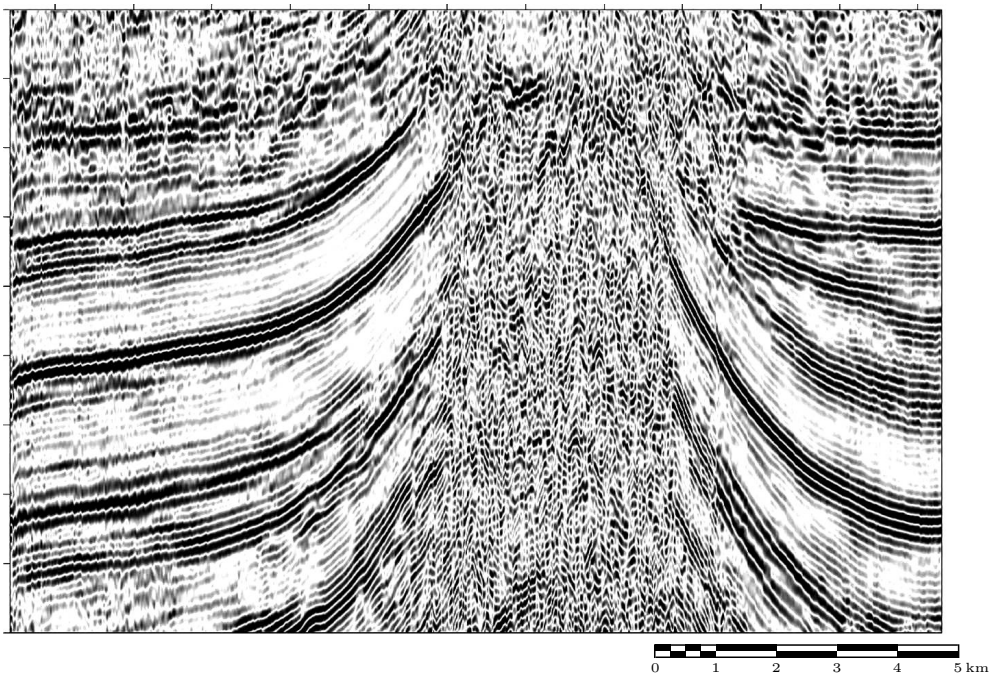

Figure 5.24: Low frequency reconstructed data in Figure 5.22 using 3D MWNI.

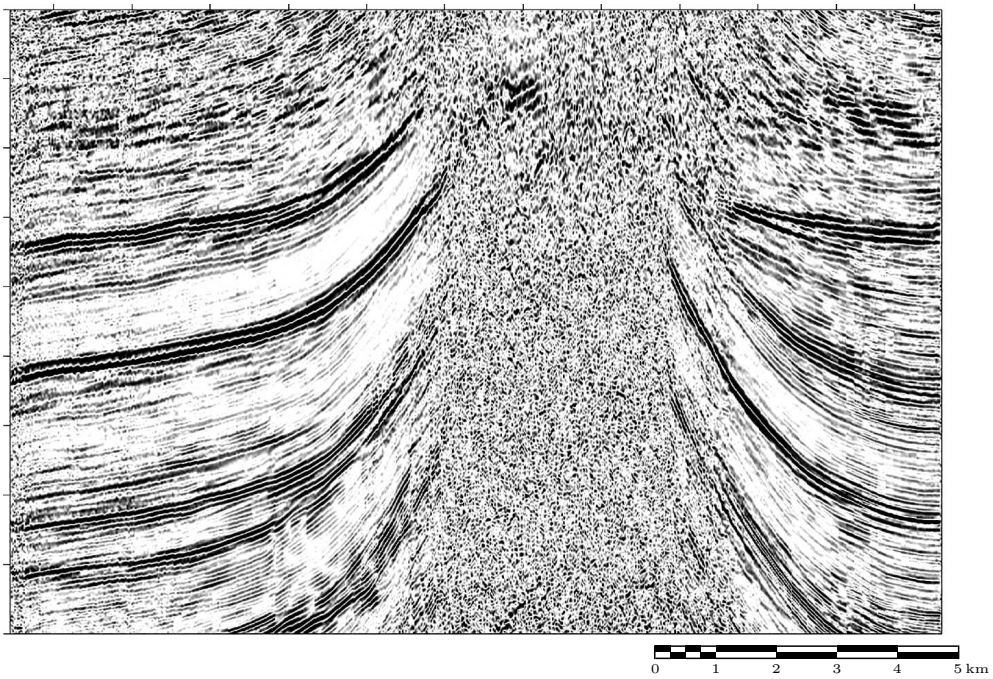

Figure 5.25: 3D MSAR interpolation of data in Figure 5.22.

4D spatial reconstruction

The final test consists of including all 4D [2] spatial directions in the process of data reconstruction. In this stage both offset and azimuth directions are discretized into a regular grid and the available traces are binned into their proper grid locations. The 4D reconstruction is carried out in the spatial windows composed of 12 in-lines, 12 cross-lines, 12 offset (offset interval equal to 100 (m)), and 12 azimuths (azimuth interval equal to 15°). Figure 5.26 shows the reconstruction of all frequencies using MWNI after stacking offset and azimuth direction for the same inline location shown in Figure 5.22. Figure 5.27 shows the reconstruction of the low frequency portion of data using the MWNI method. The MSAR algorithm is used to extract 4D prediction filters with the size $3 \times 3 \times 3 \times 3$ for all the frequencies using only the reconstructed low frequencies in Figure 5.27. Figure 5.28 shows the reconstructed data using 4D MSAR-K reconstruction method.

A comparison between 2D, 3D, and 4D reconstruction of the land data shows that as the number of spatial dimension increases the final stacked sections reveal more of continuous events which were absent in the original stacked section. This is particularly true for shallow reflectors on top of the salt body as well as for strong dipping reflectors on the flank of salt

[2] This is commonly known as 5D interpolation in the seismic data processing industry (Trad, 2008)

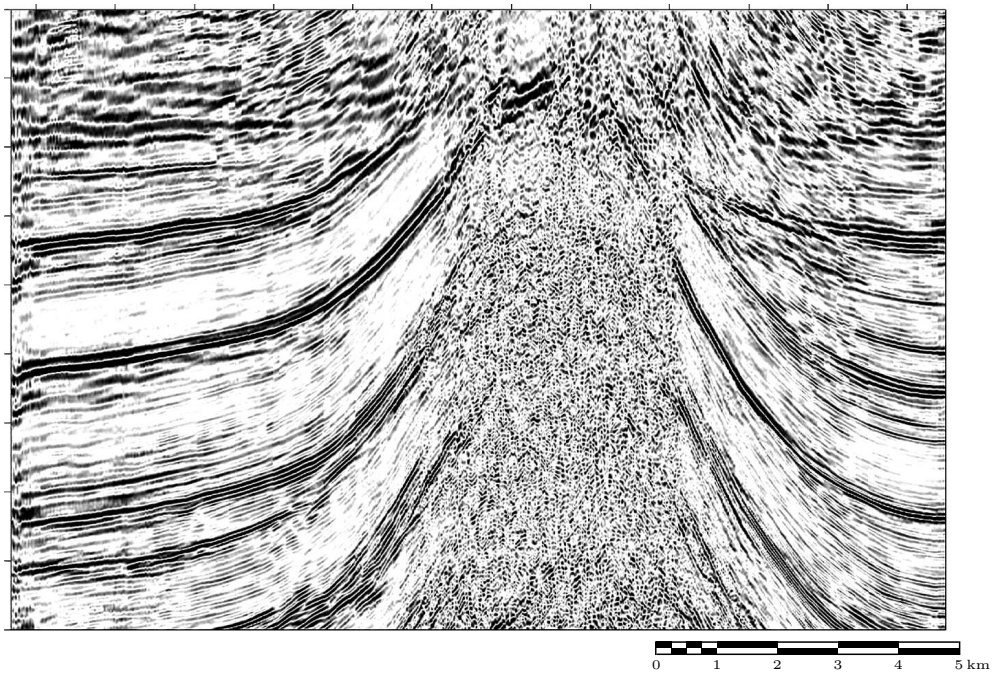

Figure 5.26: 4D MWNI interpolation of data in Figure 5.22.

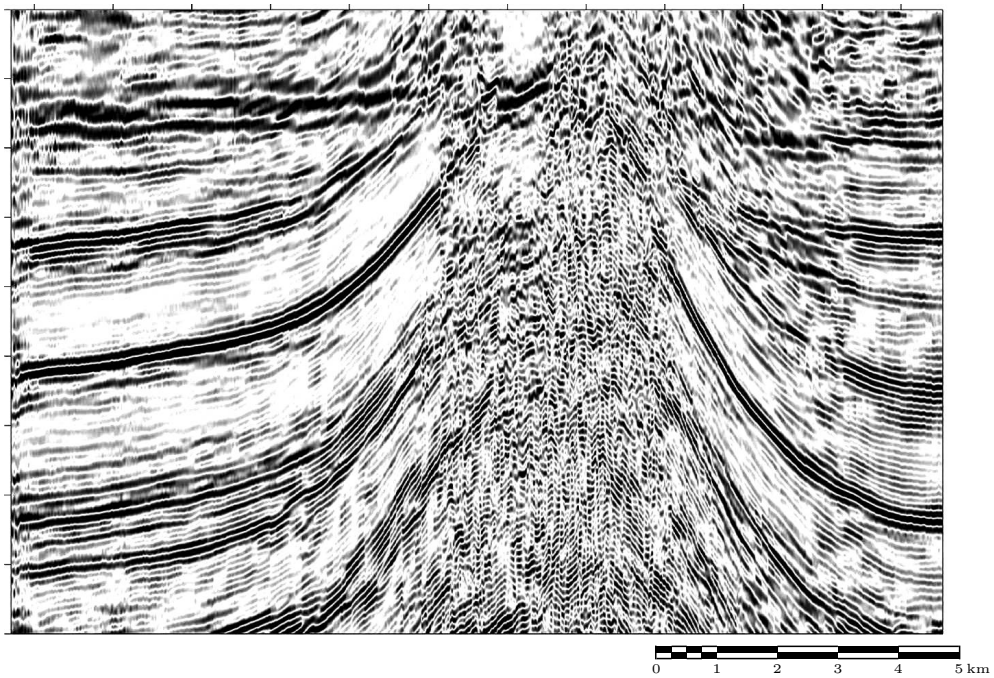

Figure 5.27: Low frequency reconstructed data in Figure 5.22 using 4D MWNI.

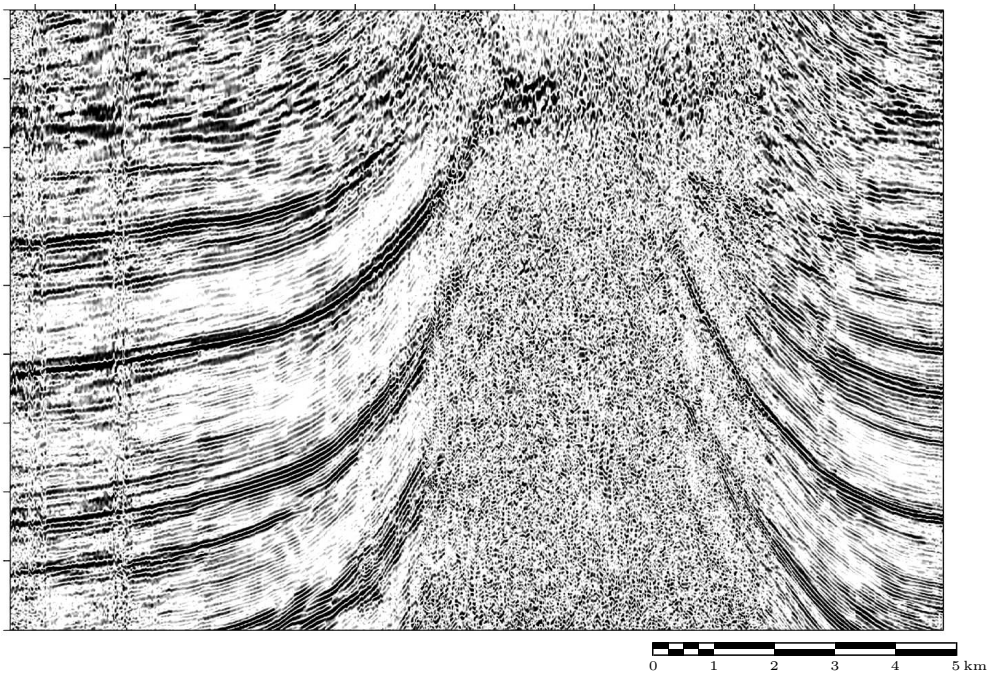

Figure 5.28: 4D MSAR interpolation of data in Figure 5.22.

body. These features are important as they add great value to the seismic interpretation carried out by geophysicists and geologists. In particular, the interpolated/reconstructed data allows for a better delineation of the salt boundaries with the associated potential of reducing the risk at the time of deciding where to drill.

5.4 Conclusions

In this chapter, the MSAR reconstruction method was extended to work with multidimensional data. The method entices the cooperative application of a multidimensional Fourier-based technique, MWNI (Liu and Sacchi, 2004), to reconstruct the unaliased part of the data and a multidimensional MSAR algorithm to reconstruct the high-frequency, and potentially aliased, part of the data. The MSAR algorithm extracts information from low frequencies that were reconstructed via MWNI to reconstruct high frequencies.

The results of 1D and 2D synthetic data reconstruction, and also 2D, 3D, and 4D real data examples, show that MSAR-K is capable of reconstructing multidimensional data. Extracted prediction filters from different low frequencies for a specific high frequency are averaged and used for reconstructing missing spatial samples. The averaging can also serve to attenuate the influence of low frequency noise that can affect the estimation of the prediction filters.

Applying MSAR-K in small spatial windows of data leads to better reconstruction results for records with curved events. Inside small windows of data, events appear as linear events and therefore they can be accurately estimated using prediction filters. Small windows of data also help to reduce the computational costs of the algorithm. The computational cost of estimating missing samples in the spatial domains using prediction filter increases with the dimension of the data.

CHAPTER 6

Adaptive FX Interpolation

6.1 Introduction

In this chapter, a new methodology for the interpolation of curved seismic events will be introduced. The proposed method can be considered as an improvement to the method proposed by Spitz (1991) often called f-x interpolation method. The new method is used for interpolation of nonlinear seismic events without data windowing. Spitz's algorithm considers data as a superposition of events with linear moveout in the time-space (t-x) domain. A superposition of linear events transform into a superposition of complex sinusoids in the f-x domain. Complex sinusoids can be reconstructed via prediction filters (autoregressive operators); this property is used to establish a signal model not only for f-x interpolation (Spitz, 1991) but also for f-x random noise attenuation methods (Canales, 1984; Soubaras, 1994; Sacchi and Kuehl, 2000).

Spitz (1991) showed that the prediction filter estimated from the temporal frequency f can be used to interpolate data at frequency $2\,f$. Prediction filters estimated from the low-frequency (alias-free) portion of the data are used to interpolate the high-frequency aliased data components. The latter makes f-x interpolation with prediction filters a strong contender for interpolating aliased data (Abma and Kabir, 2005).

Several modifications to Spitz's prediction filtering interpolation have been proposed. For instance, Porsani (1999) proposed a half-step prediction filter scheme that makes the interpolation process more efficient. Gulunay (2003) introduced an algorithm with similarities to f-x prediction filtering with a very elegant representation in the frequency-wavenumber f-k domain. Recently, Naghizadeh and Sacchi (2007) proposed a modification of f-x interpolation that allows it to reconstruct data with non-uniformly missing samples.

Seismic interpolation algorithms depend on a signal model. f-x interpolation methods are not an exception to the preceding statement; they assume data composed of a finite number of waveforms with constant dip. This assumption can be validated via windowing. Interpolation methods driven by, for instance, local Radon transforms (Sacchi et al., 2004) and Curvelet frames (Herrmann and Hennenfent, 2008) assume a signal model that consists of events with constant local dip. In addition, they implicitly define operators that are local without the necessity of windowing. This is an attractive property, in particular, when compared to non-local interpolation methods (operators defined on a large spatial aperture) where optimal results are only achievable when the seismic event matches the kinematic signature of the operator. Examples of the latter are interpolation methods based on the hyperbolic/parabolic Radon transforms (Darche, 1990; Trad et al., 2002) and migration operators (Trad, 2003).

As I have already pointed out, f-x methods require windowing strategies to cope with continuous changes in dominant wavenumbers (or dips in t-x). In this chapter I propose a method that avoids the necessity of spatial windows; the prediction filters are automatically updated as lateral variations of dip are encountered. This concept can be implemented in a somewhat cumbersome process that requires classical f-x interpolation in a rolling window. In this chapter I have preferred to use the framework of recursive least squares (Honig and Messerschmidt, 1984; Marple, 1987) to update prediction filters in a recursive fashion. Following Spitz (1991), prediction filters estimated at temporal frequency f are used to reconstruct data at frequency $2f$. I made a fundamental modification to Spitz's method; the interpolation stage of the proposed algorithm uses local filters obtained via adaptive estimation with Exponentially Weighted Recursive Least Squares (EWRLS) (Marple, 1987).

It is important to mention that interpolation with nonstationary t-x prediction filters was proposed by Crawley et al. (1999). Their algorithm evaluates local t-x prediction filters and missing data; the estimation process is stabilized by imposing smoothness to adjacent local prediction filters. Our implementation, on the other hand, is in the f-x domain and the estimation process is implemented via EWRLS.

This chapter is organized as follows. First, I provide the theory for adaptive f-x interpolation and pay particular attention to a recursive implementation of adaptive interpolation via EWRLS. I then evaluate the algorithm with synthetic and real data examples and discuss the parameter selection problem associated to our algorithm.

6.2 Theory

6.2.1 Problem definition

Let's consider spatial data in the f-x domain. The data at one monochromatic temporal frequency f are indicated by the length-N discrete signal $\mathbf{x} = [x_1, x_2, x_3, \ldots, x_N]^T$. We assume local prediction filters of length M. Forward and backward prediction equations are written as follows:

$$x_{M+n} = p_1(n)x_{M+n-1} + p_2(n)x_{M+n-2} + \ldots p_M(n)x_n + f_n \tag{6.1}$$

$$x^*_n = p_1(n)x^*_{n+1} + p_2(n)x^*_{n+2} + \ldots p_M(n)x^*_{n+M} + b_n \tag{6.2}$$

where $\mathbf{p}(n) = [p_1(n), p_2(n) \ldots p_M(n)]^T$ denotes the adaptive prediction filter at spatial sample n. The quantities denoted f_n and b_n indicate the innovation terms of the forward and backward processes. Adaptive prediction filtering can be viewed as a non-stationary autoregressive model. In other words, an autoregressive model with time (space)-variant coefficients. It is important to point out that such a model can also be used to estimate evolutionary spectra for time-frequency (space-wavenumber) analysis (Priestly, 1988).

Adaptive prediction filters are estimated by minimizing the following weighted forward and backward error functions:

$$J_f(n) = \sum_{i=1}^{n} \lambda^{n-i} |\, x_{i+M} - \sum_{k=1}^{M} p_k(n)x_{i+M-k} \,|^2, \tag{6.3}$$

$$J_b(n) = \sum_{i=n}^{N-M} \lambda^{i-n} |\, x^*_i - \sum_{k=1}^{M} p_k(n)x^*_{i+k} \,|^2, \tag{6.4}$$

where $0 < \lambda < 1$ is the forgetting factor. This parameter is used to de-emphasize the contribution of data samples far away from the estimation point n.

Expressions 6.3 and 6.4 can be independently solved to estimate adaptive prediction filters. In the numerical implementation I have adopted forward modeling (equation 6.1) to estimate the prediction filters. Backward modeling equations (equation 6.2) were used to initialize our algorithm. The latter is explained in a forthcoming section. It is important to stress, however, that one could have defined a numerical implementation where backward prediction is utilized to estimate adaptive prediction filters with forward prediction to initialize the algorithm.

Defining the following auxiliary vector $\mathbf{u}(i) = [x_{i+M-1}, x_{i+M-2}, \ldots, x_i]^T$ and scalar $d(i) =$

x_{i+M}, the solution that minimizes the error function in equation 6.3 is given by

$$\begin{aligned} \mathbf{p}(n) &= (\sum_{i=1}^{n} \lambda^{n-i} \mathbf{u}(i)\mathbf{u}(i)^H)^{-1} \sum_{i=1}^{n} \lambda^{n-i} \mathbf{u}(i)d(i) \\ &= [\Phi(n)]^{-1} \boldsymbol{\psi}(n)\,, \end{aligned} \tag{6.5}$$

where

$$\Phi(n) = \sum_{i=1}^{n} \lambda^{n-i} \mathbf{u}(i)\mathbf{u}(i)^H \tag{6.6}$$

$$\boldsymbol{\psi}(n) = \sum_{i=1}^{n} \lambda^{n-i} \mathbf{u}(i)d(i)\,. \tag{6.7}$$

6.2.2 Adaptive estimation via EWRLS

One possible solution of the adaptive prediction problem given by equation 6.3 involves solving 6.5 for each spatial position n. The latter will require the inversion of the matrix $\Phi(n)$ at every spatial position n. I will circumvent the inversion of $\Phi(n)$ by adopting a recursive scheme where $\mathbf{p}(n)$ is obtained from $\mathbf{p}(n-1)$ and the data point $x(n)$.

I derive the EWRLS algorithm following (Honig and Messerschmidt, 1984) and modify it for complex algebra. The prediction filter at estimation point n is given by

$$\mathbf{p}(n) = [\boldsymbol{\Phi}(n)]^{-1} \boldsymbol{\psi}(n)\,. \tag{6.8}$$

Similarly, the prediction filter at spatial position $n-1$ is given by

$$\mathbf{p}(n-1) = [\boldsymbol{\Phi}(n-1)]^{-1} \boldsymbol{\psi}(n-1)\,. \tag{6.9}$$

Now an update rule is required to go from 6.9 to 6.8 where we bypass the inversion of the matrices $\boldsymbol{\Phi}(n-1)$ and $\boldsymbol{\Phi}(n)$. Let's first write the matrix $\boldsymbol{\Phi}(n)$ and the vector $\boldsymbol{\psi}(n)$ in terms of $\boldsymbol{\Phi}(n-1)$ and $\boldsymbol{\psi}(n-1)$

$$\begin{aligned} \boldsymbol{\Phi}(n) &= \sum_{i=1}^{n} \lambda^{n-i} \mathbf{u}(i)\mathbf{u}(i)^H \\ &= \lambda \sum_{i=1}^{n-1} \lambda^{n-1-i} \mathbf{u}(i)\mathbf{u}(i)^H + \mathbf{u}(n)\mathbf{u}(n)^H \\ &= \lambda \boldsymbol{\Phi}(n-1) + \mathbf{u}(n)\mathbf{u}(n)^H, \end{aligned} \tag{6.10}$$

$$\begin{aligned}\boldsymbol{\psi}(n) &= \sum_{i=1}^{n} \lambda^{n-i}\mathbf{u}(i)d(i) \\ &= \lambda \sum_{i=1}^{n-1} \lambda^{n-1-i}\mathbf{u}(i)d(i) + \mathbf{u}(n)d(n) \\ &= \lambda\boldsymbol{\psi}(n-1) + \mathbf{u}(n)d(n)\,. \end{aligned} \tag{6.11}$$

We now use the matrix inversion lemma (also called the Sherman-Morrison-Woodbury formula (Hager, 1989)). If $\mathbf{A}$ and $\mathbf{B}$ are $K \times K$ positive definite matrices, $\mathbf{D}$ is a $L \times L$ matrix, and $\mathbf{C}$ is a $K \times L$ such that

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^{H} \tag{6.12}$$

then the inverse of the matrix $\mathbf{A}$ is given by the following expression

$$\mathbf{A}^{-1} = \mathbf{B} + \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^{H}\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^{H}\mathbf{B}\,. \tag{6.13}$$

The proof of the above expression is provided in Appendix C. The following substitutions $\mathbf{A}_{M\times M} = \boldsymbol{\Phi}(n)$, $\mathbf{B}_{M\times M} = \lambda^{-1}\boldsymbol{\Phi}^{-1}(n-1)$, $\mathbf{C}_{M\times 1} = \mathbf{u}(n)$, and $\mathbf{D}_{1\times 1} = 1$ are used to apply the inversion lemma to equation 6.10

$$\boldsymbol{\Phi}^{-1}(n) = \lambda^{-1}\boldsymbol{\Phi}^{-1}(n-1) - \frac{\lambda^{-2}\boldsymbol{\Phi}^{-1}(n-1)\mathbf{u}(n)\mathbf{u}(n)^{H}\boldsymbol{\Phi}^{-1}(n-1)}{1 + \lambda^{-1}\mathbf{u}(n)^{H}\boldsymbol{\Phi}^{-1}(n-1)\mathbf{u}(n)}. \tag{6.14}$$

We now define $\mathbf{R}(n) = \boldsymbol{\Phi}^{-1}(n)$, and introduce a new vector $\boldsymbol{\omega}(n)$

$$\boldsymbol{\omega}(n) = \frac{\lambda^{-1}\mathbf{R}(n-1)\mathbf{u}(n)}{1 + \lambda^{-1}\mathbf{u}(n)^{H}\mathbf{R}(n-1)\mathbf{u}(n)}. \tag{6.15}$$

Now by multiplying both sides of equation 6.14 by $\mathbf{u}(n)$ and, after some mathematical manipulations, expression 6.15 simplifies to $\boldsymbol{\omega}(n) = \mathbf{R}(n)\mathbf{u}(n)$. Now equation 6.14 can be written as

$$\mathbf{R}(n) = \lambda^{-1}\mathbf{R}(n-1) - \lambda^{-1}\boldsymbol{\omega}(n)\mathbf{u}(n)^{H}\mathbf{R}(n-1)\,. \tag{6.16}$$

We now derive the update equation for $\mathbf{p}(n)$. Starting from equation 6.8 we have:

$$
\begin{aligned}
\mathbf{p}(n) &= [\mathbf{\Phi}(n)]^{-1}\boldsymbol{\psi}(n) \\
&= \mathbf{R}(n)\boldsymbol{\psi}(n) \\
&= \mathbf{R}(n)[\lambda\boldsymbol{\psi}(n-1) + \mathbf{u}(n)d(n)] \\
&= \mathbf{R}(n)[\lambda\mathbf{\Phi}(n-1)\mathbf{p}(n-1) + \mathbf{u}(n)d(n)] \\
&= \mathbf{R}(n)[(\mathbf{\Phi}(n) - \mathbf{u}(n)\mathbf{u}(n)^H)\mathbf{p}(n-1) + \mathbf{u}(n)d(n)] \\
&= \mathbf{p}(n-1) - \mathbf{R}(n)\mathbf{u}(n)\mathbf{u}(n)^H\mathbf{p}(n-1) + \mathbf{R}(n)\mathbf{u}(n)d(n) \\
&= \mathbf{p}(n-1) + \mathbf{R}(n)\mathbf{u}(n)[d(n) - \mathbf{u}(n)^H\mathbf{p}(n-1)] \\
&= \mathbf{p}(n-1) + \mathbf{R}(n)\mathbf{u}(n)\alpha(n) \\
&= \mathbf{p}(n-1) + \boldsymbol{\omega}(n)\alpha(n) \qquad (6.17)
\end{aligned}
$$

where we have defined the scalar variable

$$\alpha(n) = d(n) - \mathbf{u}(n)^H\mathbf{p}(n-1)\,. \qquad (6.18)$$

Utilizing the expressions 6.15, 6.16, 6.17, and 6.18, the adaptive estimation of prediction filters can be summarized as follows

$$
\begin{aligned}
\text{Let } \mathbf{R}(n-1) &= \Phi^{-1}(n-1) \\
\text{Update } \mathbf{p} \text{ and } \mathbf{R} & \\
\boldsymbol{\omega}(n) &= \frac{\lambda^{-1}\mathbf{R}(n-1)\mathbf{u}(n)}{1 + \lambda^{-1}\mathbf{u}(n)^H\mathbf{R}(n-1)\mathbf{u}(n)} \\
\alpha(n) &= d(n) - \mathbf{u}(n)^H\mathbf{p}(n-1) \qquad (6.19) \\
\mathbf{p}(n) &= \mathbf{p}(n-1) + \boldsymbol{\omega}(n)\alpha(n) \\
\mathbf{R}(n) &= \lambda^{-1}\mathbf{R}(n-1) - \lambda^{-1}\boldsymbol{\omega}(n)\mathbf{u}(n)^H\mathbf{R}(n-1)\,.
\end{aligned}
$$

6.2.3 Initialization

It is evident from equation 6.19 that in order to initiate the recursive algorithm $\mathbf{p}(1)$ and $\mathbf{R}(1)$ are required. Reliable estimates of these parameters are obtained using backward prediction modeling.

Denoting $c(i) = x_i^*$ and $\mathbf{v}(i) = [x_{i+1}^*, x_{i+2}^*, \ldots, x_{i+M}^*]^T$, the solution for equation 6.4 at

estimation point $n = 1$ is given by

$$\mathbf{\Phi}(1) = \sum_{i=1}^{n} \lambda^{i-1} \mathbf{v}(i)\mathbf{v}(i)^H, \tag{6.20}$$

$$\boldsymbol{\psi}(1) = \sum_{i=1}^{n} \lambda^{i-1} \mathbf{v}(i)c(i)\,. \tag{6.21}$$

The required initial parameters are obtained by

$$\mathbf{R}(1) = \mathbf{\Phi}^{-1}(1) \tag{6.22}$$

$$\mathbf{p}(1) = \mathbf{R}(1)\boldsymbol{\psi}(1)\,. \tag{6.23}$$

Equation 6.22 is the only inversion needed by the algorithm. After initializing the algorithm with $\mathbf{R}(1)$ and $\mathbf{p}(1)$, the recursive solution (equation 6.19) is used to estimate adaptive prediction filters for all spatial positions.

6.2.4 Interpolation via adaptive prediction filters

In order to interpolate the data we consider spatial samples at frequency f with their associated prediction filters estimated from frequency $f/2$ (Spitz, 1991). Consider, for instance, a prediction filter of length $M = 3$, the equations for adaptive forward and backward prediction associated to the i-th filter are given by

$$\begin{pmatrix} p_3(i) & p_2(i) & p_1(i) & -1 & 0 & 0 & 0 \\ 0 & p_3(i) & p_2(i) & p_1(i) & -1 & 0 & 0 \\ 0 & 0 & p_3(i) & p_2(i) & p_1(i) & -1 & 0 \\ 0 & 0 & 0 & p_3(i) & p_2(i) & p_1(i) & -1 \\ 0 & 0 & 0 & -1 & p_1^*(i) & p_2^*(i) & p_3^*(i) \\ 0 & 0 & -1 & p_1^*(i) & p_2^*(i) & p_3^*(i) & 0 \\ 0 & -1 & p_1^*(i) & p_2^*(i) & p_3^*(i) & 0 & 0 \\ -1 & p_1^*(i) & p_2^*(i) & p_3^*(i) & 0 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} x_i \\ x_{\frac{2i+1}{2}} \\ x_{i+1} \\ x_{\frac{2(i+1)+1}{2}} \\ x_{i+2} \\ x_{\frac{2(i+2)+1}{2}} \\ x_{i+3} \end{pmatrix} \approx \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{6.24}$$

The rational indexes represent the desired samples to be estimated by the interpolation scheme. Notice that in the system 6.24 we have not included the innovation/error terms seen in expressions 6.1-6.2. We have, however indicated the presence of the small innovation term by including $\approx \mathbf{0}$.

Equations similar to 6.24, for all possible samples i, lead to an over-determined system from where one can estimate the unknown data. Let's $\mathbf{G}^i$ be the above $[2(M+1)] \times [2M+1]$ matrix. Matrices for all possible spatial samples i are combined in one augmented system of equations containing all the known and unknown data

$$\begin{pmatrix} \mathbf{G}^1 & \mathbf{0}_{[2(M+1)]\times[2(N-M-1)]} \\ \cdots & \cdots \quad \cdots \\ \mathbf{0}_{[2(M+1)]\times[2(i-1)]} & \mathbf{G}^i \quad \mathbf{0}_{[2(M+1)]\times[2(N-M-i)]} \\ \cdots & \cdots \quad \cdots \\ \mathbf{0}_{[2(M+1)]\times[2(N-M-1)]} & \mathbf{G}^{N-M} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_{\frac{3}{2}} \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_{N-1} \\ x_{\frac{2N-1}{2}} \\ x_N \end{pmatrix} \approx \mathbf{0}_{[2(M+1)(N-M)]\times[1]} \,. \tag{6.25}$$

The size of the augmented matrix in equation 6.25 is $[2(M+1)(N-M)] \times [2N-1]$. By dividing data samples into vectors containing known samples

$$\mathbf{x}_k = [x_1, x_2, x_3, \ldots, x_N\,]^T$$

and unknown samples

$$\mathbf{x}_u = [x_{3/2}, x_{5/2}, x_{7/2}, \ldots, x_{(2N-1)/2}\,]^T \,,$$

equation 6.25 reduces to the following over-determined system

$$\mathbf{A}\mathbf{x}_u \quad \approx \quad \mathbf{B}\mathbf{x}_k \,. \tag{6.26}$$

The matrices $\mathbf{A}$ and $\mathbf{B}$ depend on adaptive prediction filters estimated with EWRLS. The last system of equations is solved via the method of least squares

$$\mathbf{x}_u \quad = \quad (\mathbf{A}^H\,\mathbf{A}\,)^{-1}\,\mathbf{A}^H\,\mathbf{B}\,\mathbf{x}_k \,. \tag{6.27}$$

In the numerical examples I have used the method of conjugate gradients to solve for the unknown samples $\mathbf{x}_u$. An important advantage of using the method of conjugate gradients is that the number of iterations required for convergence to $\mathbf{x}_u$ can be truncated to gain computational efficiency. In the following examples I have used a maximum number of iterations $\approx N/4$.

6.3 Synthetic and Field Data Examples

6.3.1 Synthetic data

I first examine the performance of *f-x* adaptive interpolation with synthetic examples. Amplitude spectra are portrayed in terms of normalized frequency and normalized wavenumber. Normalized frequency and wavenumber axes are obtained by considering $\Delta t = 1$ and $\Delta x = 1$, respectively. This means that in order to obtain frequency axes in Hz and Cycles/m one must divide the normalized ones by Δt and Δx, respectively.

The first synthetic data set is composed of three linear events. Figure 6.1a shows the synthetic data. The data are interpolated using adaptive prediction filters of length $M = 3$ and forgetting factors $\lambda = 1$ (Figure 6.1b) and $\lambda = 0.25$ (Figure 6.1c). Figures 6.2a, 6.2b and 6.2c show the *f-k* domain representation of Figures 6.1a, 6.1b and 6.1c, respectively. When $\lambda = 1$ the estimation of the prediction filter considers all the data with equal weight regardless of their proximity to the estimation point. In other words, there is no need of adaptability to local dip and the EWRLS algorithm identifies without effort the correct prediction filters required to interpolate the data. In this case the algorithm works for both $\lambda = 1$ and $\lambda = 0.25$. A small improvement in performance is obtained for $\lambda = 1$. The latter corresponds to the stationary filter case where adaptability to dip is not required as expected for this particular example.

To continue with the analysis, I now propose to examine an example where the dip of the waveform varies with space. A synthetic section composed of hyperbolic events is provided in Figure 6.3a. The *f-k* spectrum is shown in Figure 6.3c. I decimate the original gather to create the decimated data portrayed in Figure 6.3b. The *f-k* spectrum of the decimated data is provided in Figure 6.3d. The decimated section of Figure 6.3b was interpolated with forgetting factors $\lambda = 1$ (Figure 6.4a) and $\lambda = 0.15$ (Figure 6.4b). In contrast to the previous example, a forgetting factor $\lambda = 1$ results in an extremely low quality interpolation. By choosing $\lambda = 0.25$ the prediction filters are allowed to adapt to the local dip of the data and the algorithm successfully interpolates the data. Figures 6.4c and 6.4d show the *f-k* spectra of Figures 6.4a and 6.4b, respectively.

It is insightful to compare the performance of *f-x* adaptive interpolation to classical *f-x* interpolation (Spitz, 1991). The decimated synthetic data shown in Figure 6.3b is interpolated using classical *f-x* interpolation. We first applied the interpolation to the full aperture and then to small overlapping spatial windows. *f-x* interpolation with Spitz's method using a single predictor error obtained from the full data aperture is portrayed in Figure 6.5a. This example is quite unfair to Spitz's *f-x* interpolation because the dip of the reflection is rapidly varying with offset. Figure 6.5b portrays Spitz's interpolation when the method is

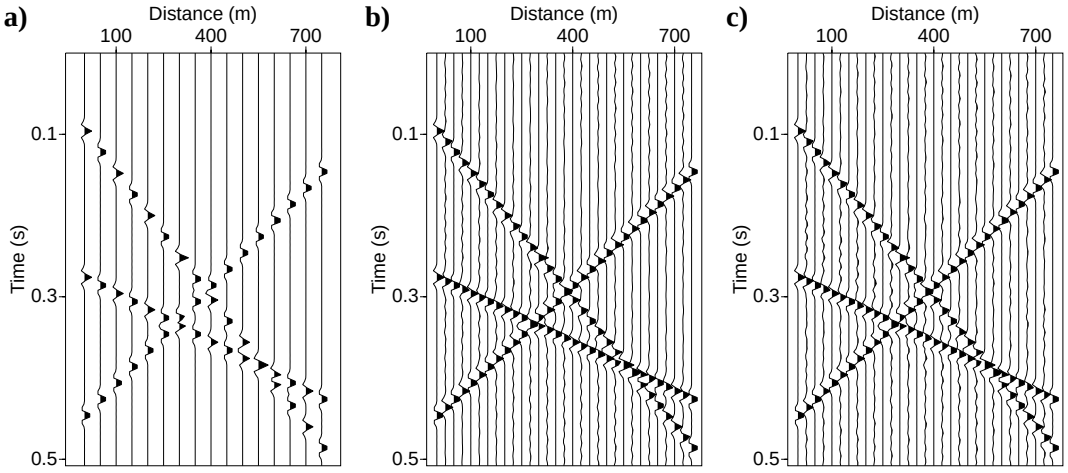

Figure 6.1: a) Original data. b) Interpolated data using *f-x* adaptive prediction filtering with forgetting factors $\lambda = 1$ and c) $\lambda = 0.25$. Prediction filters of length $M = 3$ were used for both figures.

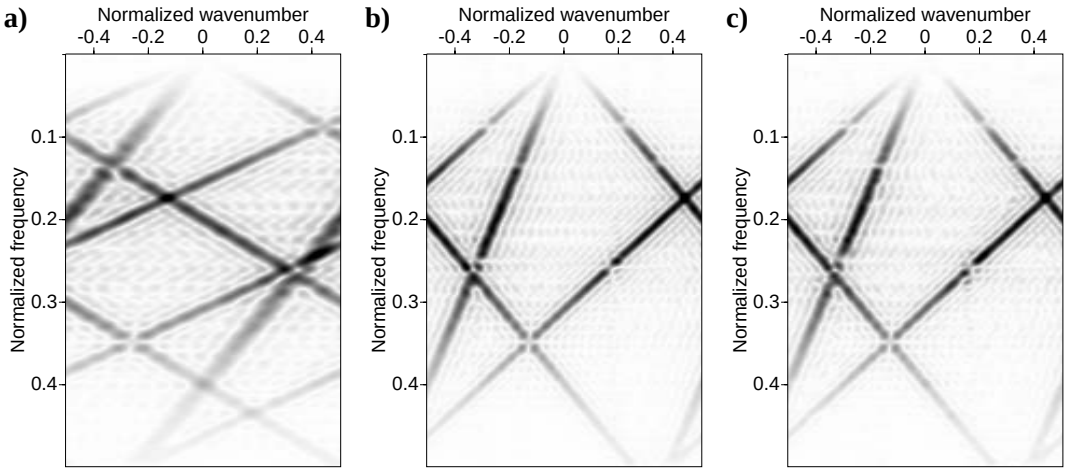

Figure 6.2: a),b) and c) are the *f-k* spectra of the data portrayed in Figures 6.1a, 6.1b and 6.1c, respectively.

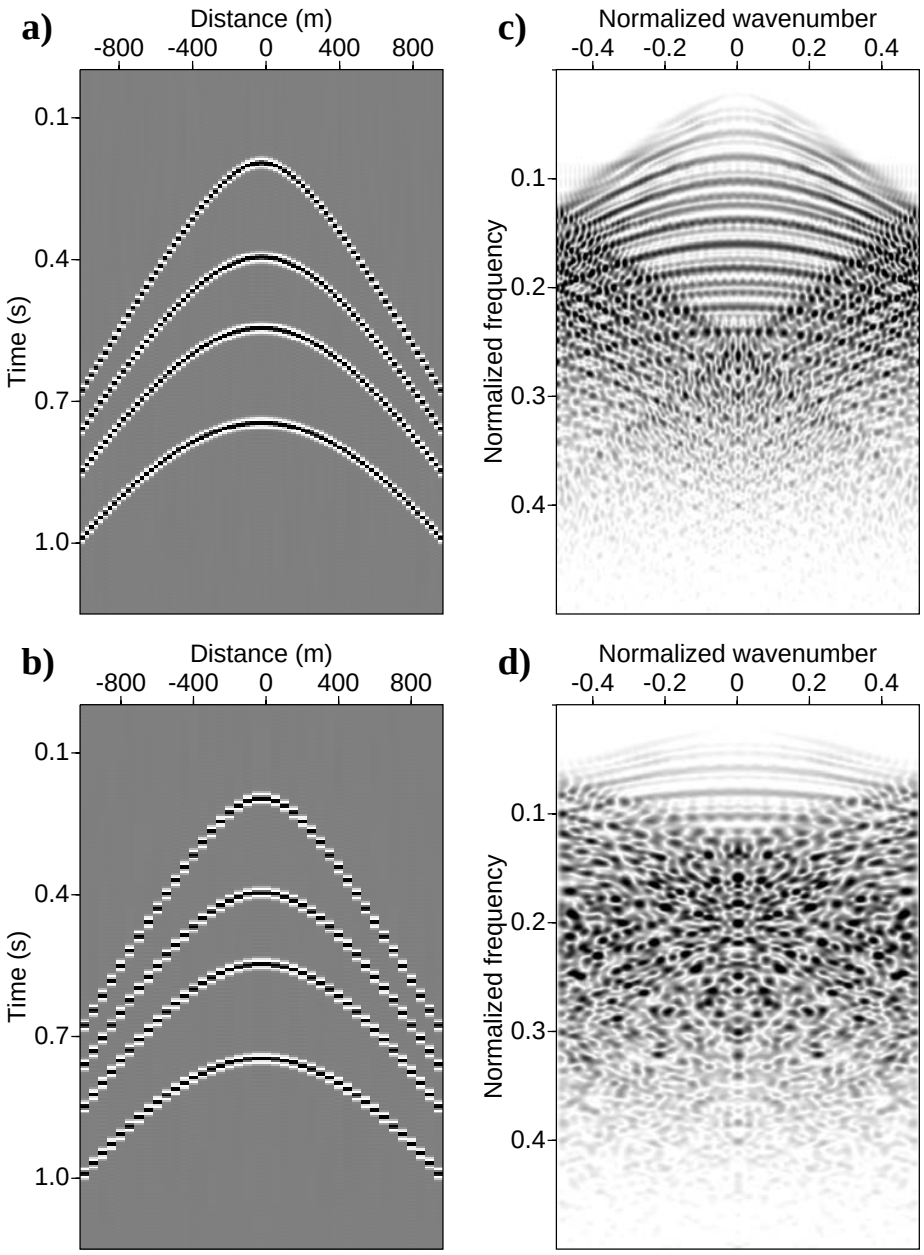

Figure 6.3: a) Original data, b) decimated data. c) and d) are the *f-k* spectra of (a) and (b), respectively.

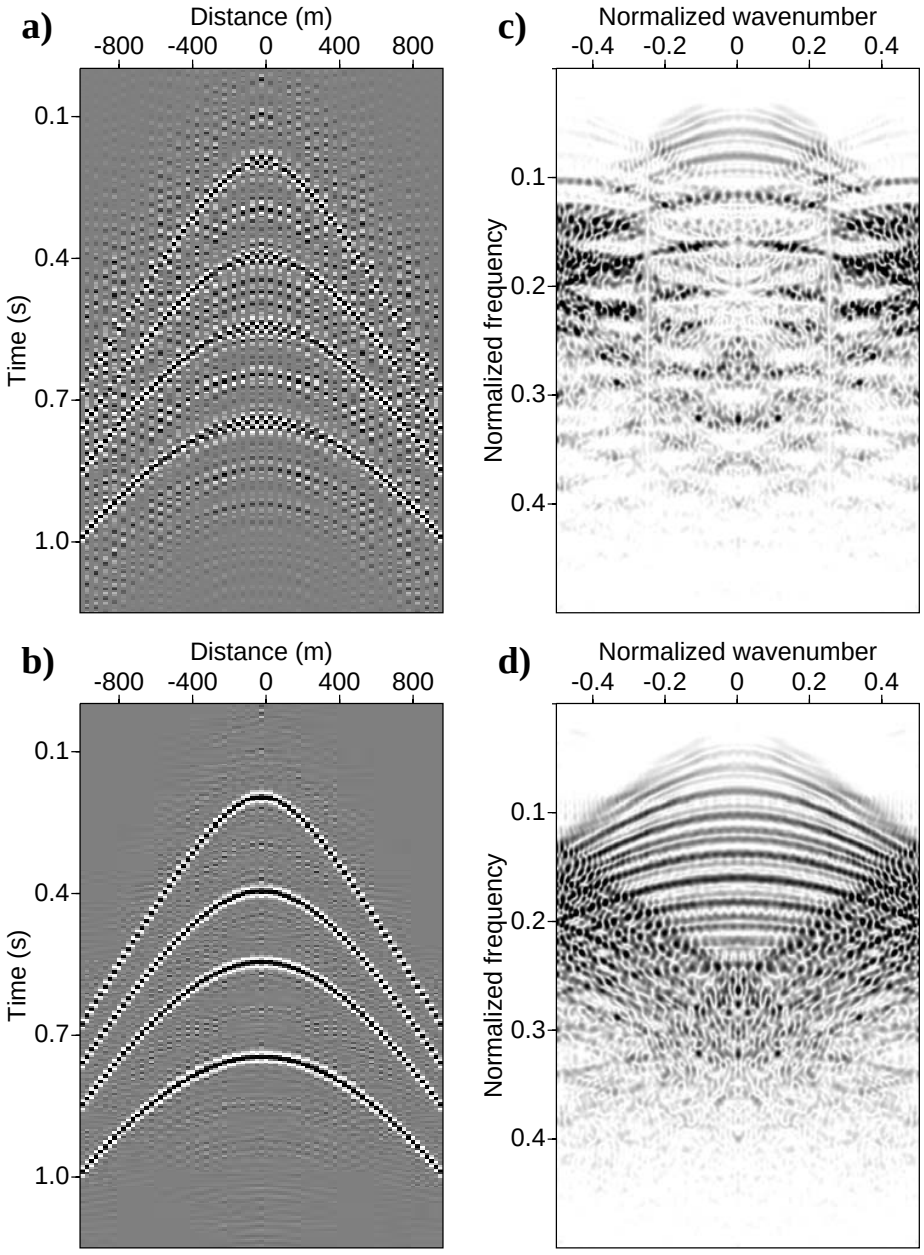

Figure 6.4: a) Interpolated data via adaptive *f-x* interpolation with $\lambda = 1$. b) Interpolated data via adaptive *f-x* interpolation with $\lambda = 0.15$. The length of the prediction filter for all frequencies and panels is $M = 4$. The *f-k* spectra of (a) and (b) are provided in c) and d), respectively.

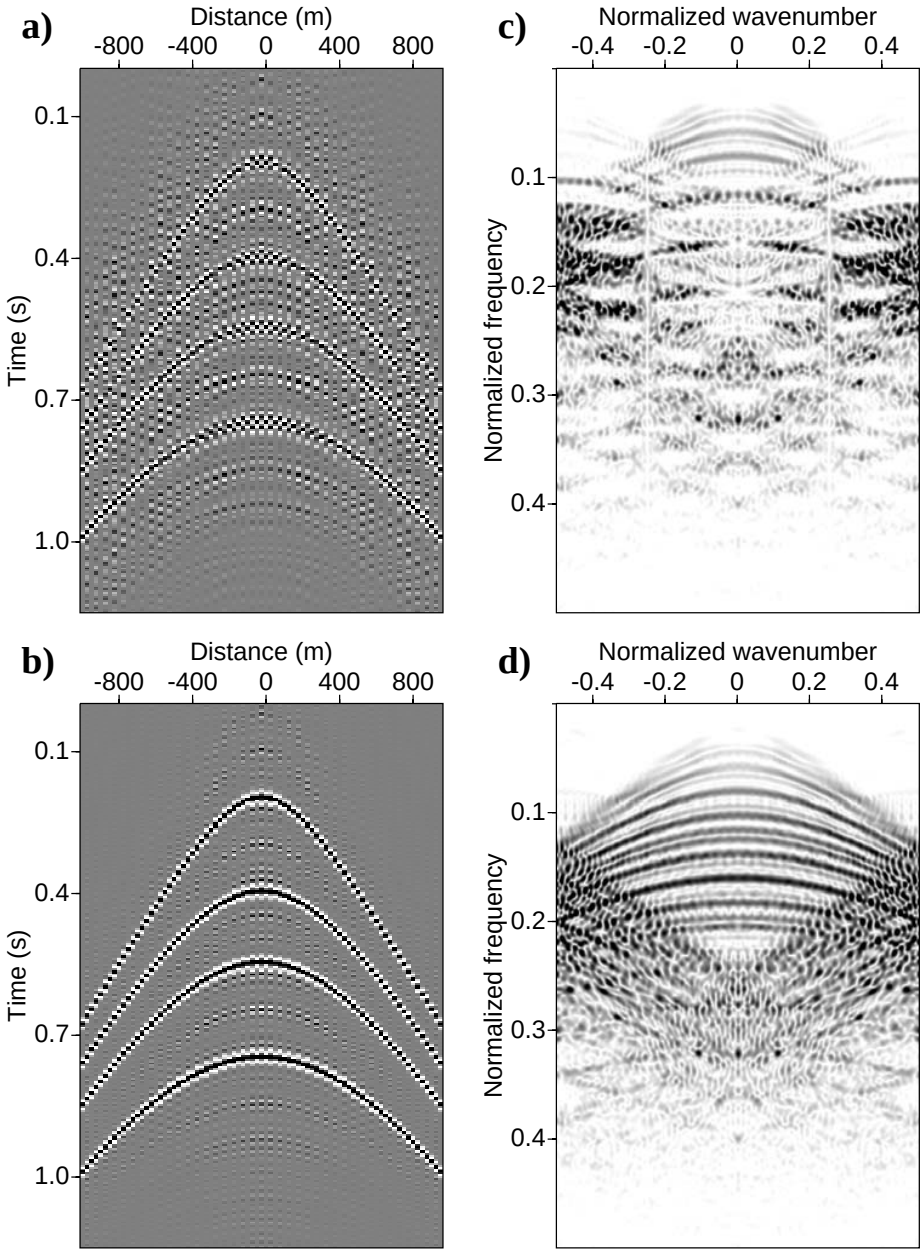

Figure 6.5: a) Interpolation with Spitz's *f-x* interpolation. Prediction filters were estimated from the full aperture. The latter violates the constant dip assumption. b) Spitz's *f-x* applied on small overlapping spatial windows of 7 traces. The method can now successfully interpolate the data. c) and d) provide the *f-k* spectra of (a) and (b), respectively.

applied on small overlapping spatial windows of 7 traces. Figures 6.5c and 6.5d show the *f-k* spectra of Figures 6.5a and 6.5b, respectively.

The interpolated data in Figures 6.4b and 6.5b as well as the original data (Figure 6.3a) contain a significant amount of alias. In order to de-alias the data one can successively apply adaptive *f-x* interpolation. Figure 6.6a shows the result of adaptive *f-x* interpolation applied to the original data (Figure 6.3a) using $\lambda = 0.3$. The data in Figure 6.6a is interpolated again using $\lambda = 0.6$ and the result is depicted in Figure 6.6b. Notice that we have increased λ since the local dip variability has decreased after the first interpolation. Figures 6.6c and 6.6d show the de-aliased *f-k* spectra of Figures 6.6a and 6.6b.

The last synthetic example consists of parabolic events with conflicting dips (Figure 6.7a). The original data were decimated to obtain the decimated data in Figure 6.7b. The decimated data are interpolated using adaptive *f-x* interpolation with $\lambda = 0.3$. The result is shown in Figure 6.7c. Figures 6.8a, 6.8b and 6.8c provide the *f-k* spectra of the original, decimated and interpolated data, respectively. This example shows that adaptive *f-x* interpolation can also resolve conflicting space-variant dips. It is important to stress that significant amount of aliased energy is visible in the original data. The adaptive *f-x* interpolation has properly resolved the alias as indicated by the *f-k* panels.

6.3.2 Marine data example

I first test *f-x* adaptive interpolation on a marine shot gather from the Gulf of Mexico. The entire gather was interpolated but only a small data window from 1.8s to 3.0s is illustrated. The original data are provided in Figure 6.9a. The decimated data used to test the interpolation are displayed in Figure 6.9b. The interpolated data and the interpolation error are provided in Figures 6.9c and 6.9d, respectively. A forgetting factor $\lambda = 0.2$ and prediction filters of length $M = 4$ were used for this example. For completeness, I also provide *f-k* spectra of the original data (Figure 6.10a), the decimated data (Figure 6.10b) and the interpolated data (Figure 6.10c).

Figure 6.11a shows a near offset section from the Gulf of Mexico. The section was interpolated using adaptive *f-x* interpolation with $\lambda = 0.2$ and prediction filters of length $M = 4$. The final interpolation is shown in Figure 6.11b. It is evident that curved diffracted events were properly interpolated. Similar results were obtained using classical *f-x* interpolation with small overlapping windows of 7 traces.

At this point a few comments about computational efficiency are in order. It is difficult to provide a comparative measure of computational cost for these examples. I have adopted the method of conjugate gradients for the data reconstruction stage for both the adaptive and non-adaptive algorithms. Therefore, the computational cost is heavily controlled by

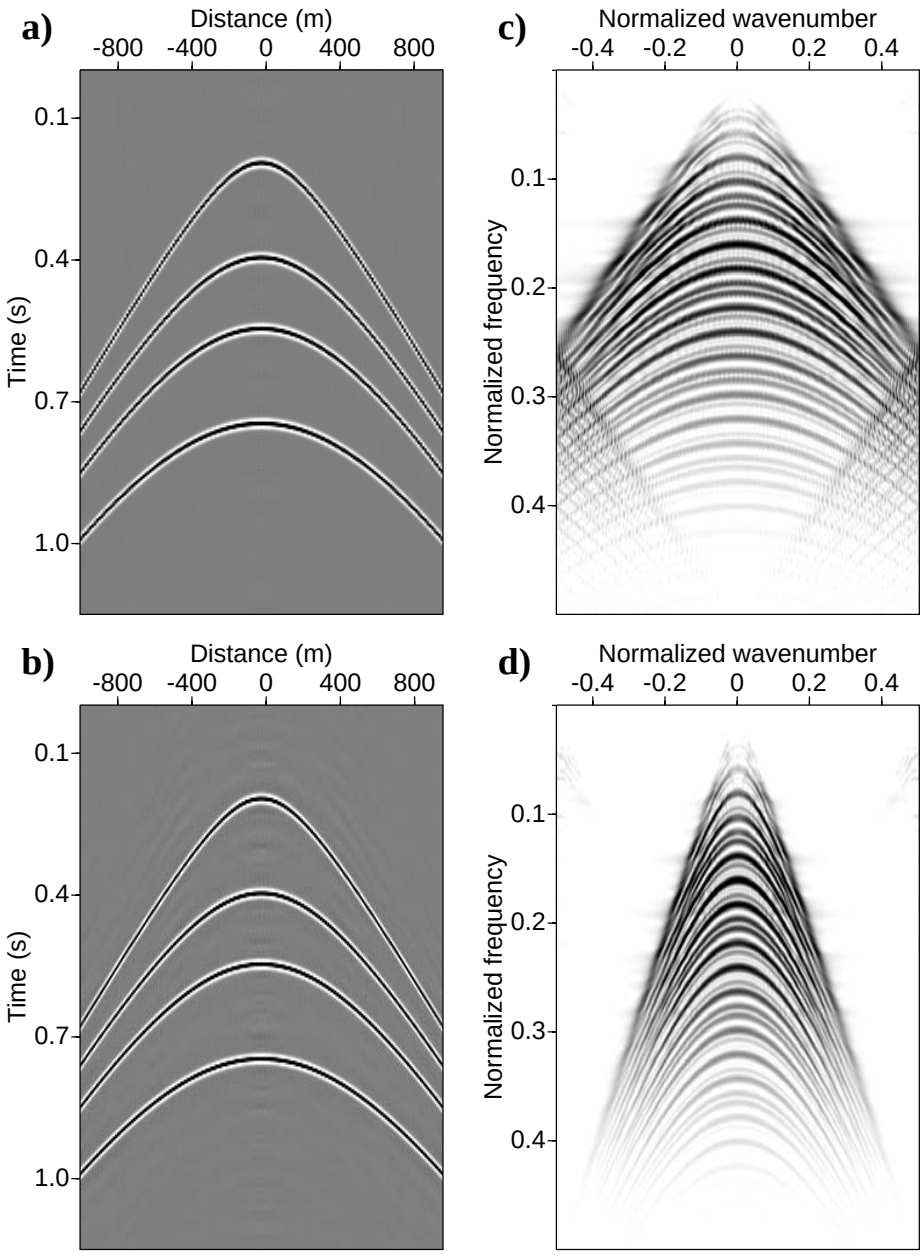

Figure 6.6: a) Adaptive *f-x* interpolation of the original data in Figure 6.3a with $\lambda = 0.3$ b) Adaptive *f-x* interpolation of a) with $\lambda = 0.6$. c) and d) are the *f-k* spectra of (a) and (b), respectively.

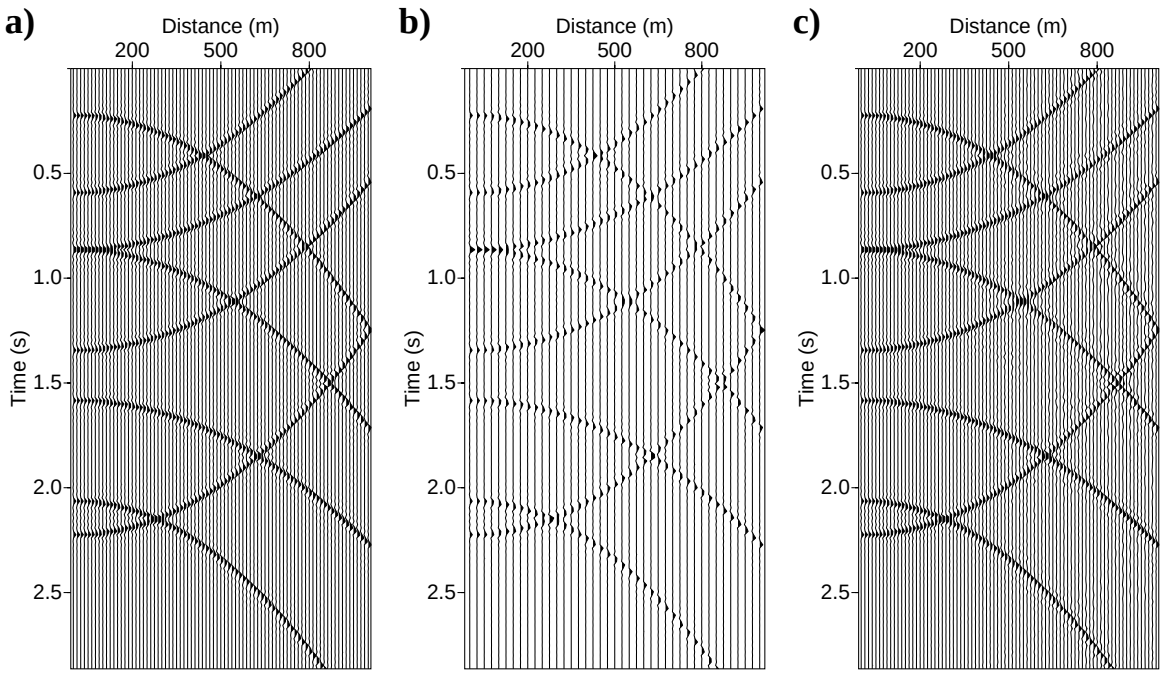

Figure 6.7: Synthetic example with conflicting dips. a) Original data. b) Decimated data. c) Interpolated data using adaptive *f-x* interpolation with forgetting factor $\lambda = 0.3$ and prediction filter length $M = 4$.

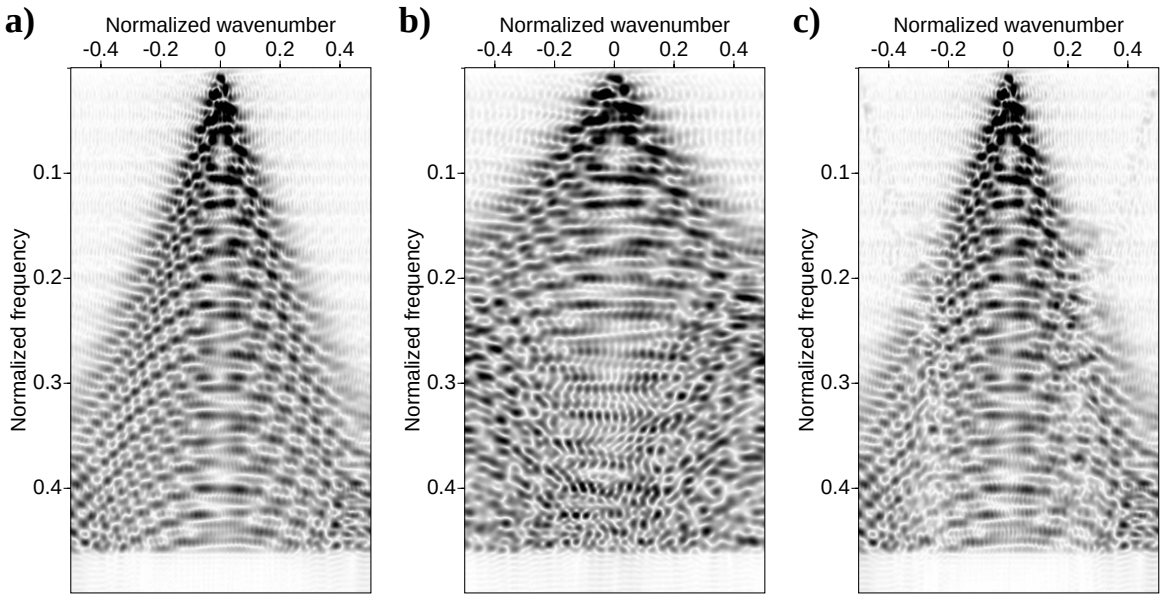

Figure 6.8: a), b) and c) provide the *f-k* spectra of the data portrayed in Figures 6.7a, 6.7b and 6.7c, respectively.

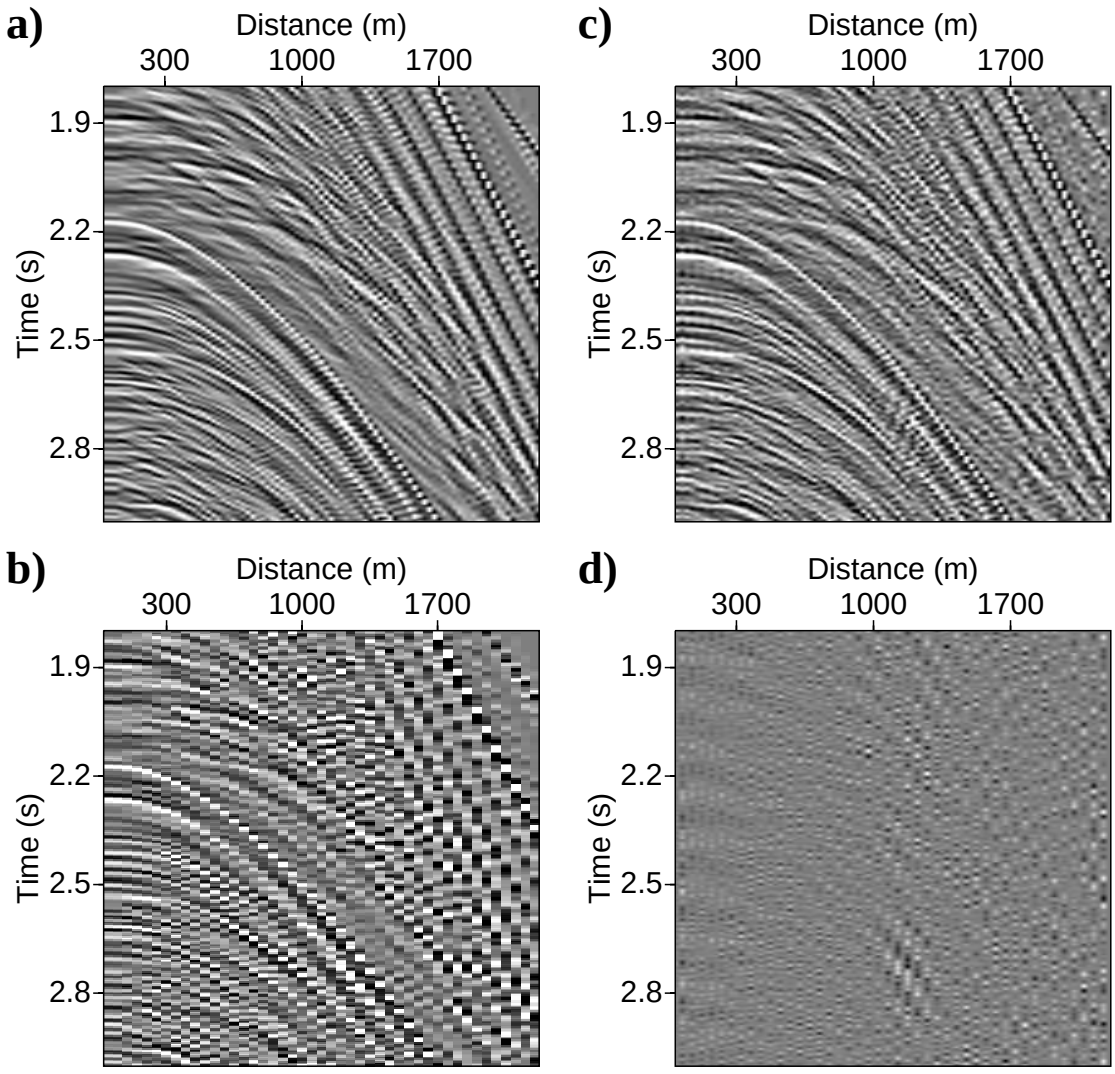

Figure 6.9: a) Marine common source gather from a data set from the Gulf of Mexico. b) Decimated data. c) *f-x* interpolation with forgetting factor $\lambda = 0.2$ and prediction filter length $M = 4$. d) The interpolation error, original minus interpolated data.

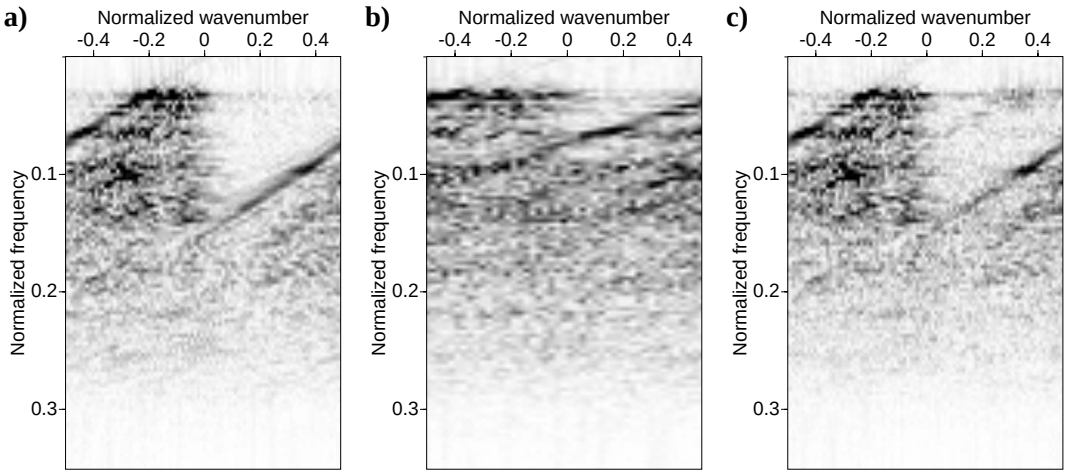

Figure 6.10: a),b) and c) are the *f-k* spectra of Figures 6.9a, 6.9b and 6.9c, respectively.

the number of iterations required by the conjugate gradients method to achieve convergence at a given frequency. The simulations indicate that there is no significant difference in the computational cost of the adaptive algorithm using EWRLS and the non-adaptive Spitz method using overlapping windows.

6.4 Parameter Selection

For optimal results an automatic process for the selection of the forgetting factor λ and filter length M is required. I have adopted the following heuristic strategy. I first decimate the data, then 3 temporal frequencies are used to compute the average reconstruction error for different values of M and λ. The minimum reconstruction error provides optimal values M_{dec} and λ_{dec} for the decimated data. When the algorithm is used to interpolate the original data I use $M = M_{dec}$ and $\lambda = \lambda_{dec}^{1/2}$. Using $\lambda = \lambda_{dec}^{1/2}$ stresses that the forgetting factor should be larger than the one for the decimated data. The above described strategy was adopted for parameter selection in the synthetic and real data examples shown in this article.

It is also important to address the problem of noisy data. As the SNR decreases one should increase the forgetting to allow for more data to participate in the estimation of the adaptive filter. One needs to realize, however, that by increasing λ we loose adaptability to changes in local dip. In other words, robustness to the presence of noise and adaptability to local dip are not attainable at the same time. Nevertheless, the real data examples show that *f-x*

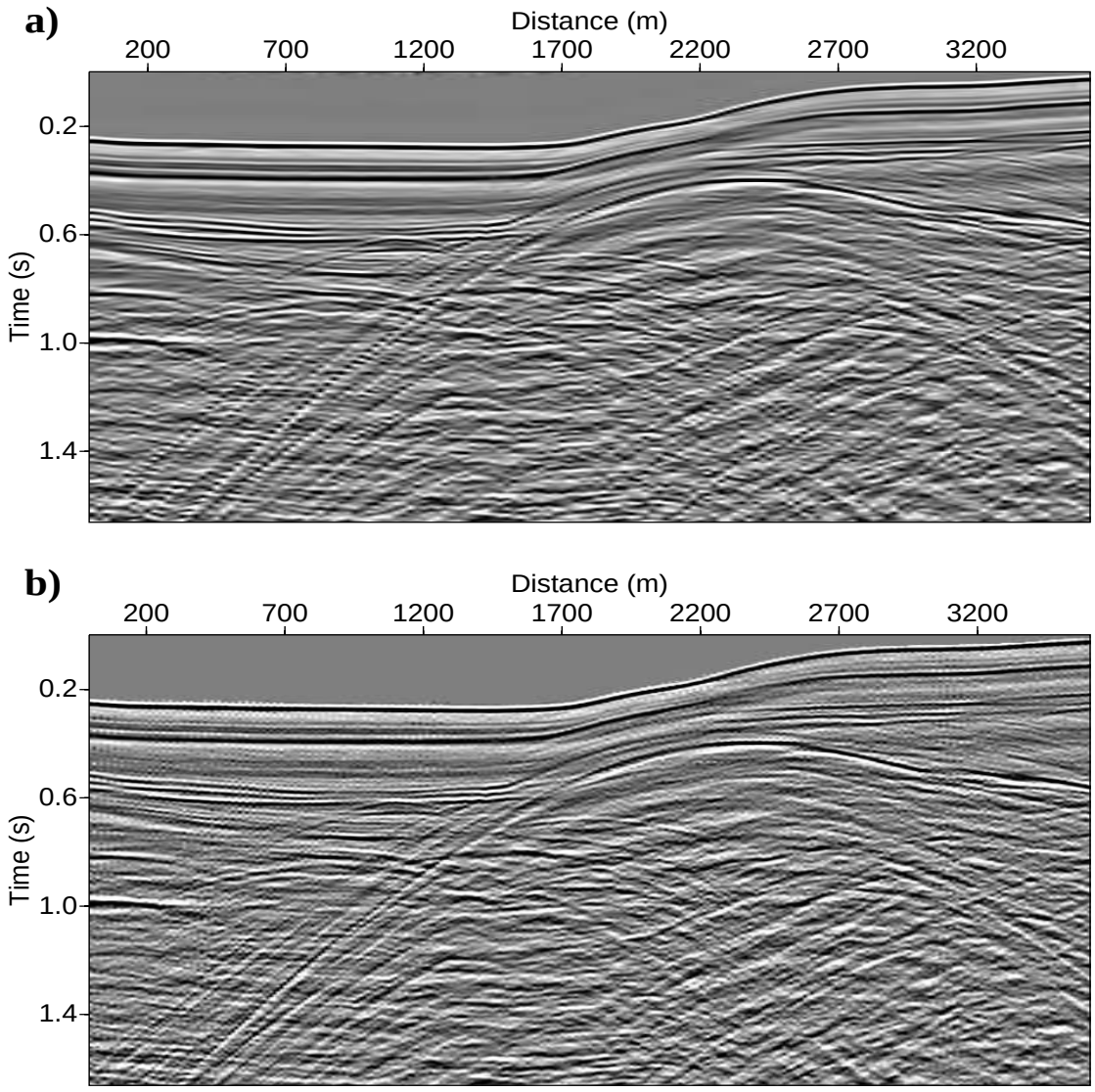

Figure 6.11: Real data example portraying the interpolation of a near offset section from the Gulf of Mexico. a) Original section. b) Interpolated section using adaptive *f-x* interpolation with forgetting factor $\lambda = 0.2$ and prediction filter length $M = 4$.

adaptive interpolation can handle less than ideal data quite well.

6.5 Conclusions

In this chapter I introduced an efficient and easy-to-implement method to interpolate seismic records composed of curved events. I consider the problem of interpolating waveforms with variable dip by re-writing *f-x* interpolation as an adaptive process. The method eliminates the need of selecting window parameters (window size and amount of overlapping between adjacent windows).

The proposed adaptive *f-x* interpolation algorithm is robust under strong changes of curvature. In addition, the method performs quite well in the presence of conflicting dips with alias as illustrated by the examples. Adaptive *f-x* interpolation depends on two parameters: operator length (as in the classical *f-x* interpolation scheme) and an extra parameter, the forgetting factor, that controls adaptability to changes in local dip. I have also proposed an heuristic method to determine the operator length and forgetting factor.

CHAPTER 7

Conclusions

This thesis studies the reconstruction of multidimensional seismic data. The main goal was to investigate the advantages and limitations of methods which are based on signal processing approaches and, introduce methodologies that provide robust and efficient alternatives for seismic data reconstruction. The focus was on Fourier reconstruction and autoregressive (AR) interpolation methods.

The chief contributions of this dissertation to the field of seismic data reconstruction can be summarized as follow:

- Analysis of discrete sampling functions and a formulation that elucidates the conditions under which Fourier methods can succeed in recovering missing data (**Chapter 3**).
- A novel reconstruction method named multistep autoregressive (MSAR) reconstruction was provided (**Chapter 4**). The MSAR method is a combination of Fourier reconstruction (for the low frequency portion of data) and autoregressive interpolation methods. MSAR does not suffer from the shortcoming of the stand-alone application of each of these methods. The reconstruction of missing data samples can be either performed in the *f-x* domain (MSAR-X) or in the *f-k* domain (MSAR-K).
- An adaptive *f-x* interpolation method is introduced to interpolate data with spatially varying wavenumbers (**Chapter 6**). The proposed adaptive interpolation method eliminates the need for spatial windowing of the data.

Each of the above methodologies has their own advantages and limitations. The optimal reconstruction method is to be chosen based on the sampling function and the nature of the

original data. I will expand more on the conditions needed for successful application of the proposed methods.

7.1 Fourier reconstruction

In this thesis I utilized the Minimum Weighted Norm Interpolation (MWNI) method (Liu and Sacchi, 2004) among the many variants of Fourier reconstruction methods that have been proposed by the seismic signal processing community. Using a band-limiting operator (low-pass filter) in the wavenumber domain, MWNI performs like band-limited Fourier reconstruction. By iteratively weighting the Fourier coefficients, the MWNI method behaves like a sparse Fourier reconstruction method (Sacchi et al., 1998).

Advantages

- The MWNI method is the best candidate for reconstructing randomly sampled data provided that the original data have a simple (sparse) Fourier representation (linear events in the t-x domain).
- MWNI with band-limitation can effectively reconstruct seismic data if the data wavenumber domain is band-limited. For instance, applying Normal Move-Out (NMO) correction can lead to a band-limited offset-wavenumber domain.
- The performance of multidimensional Fourier reconstruction is superior to the 1D application if both of the conditions mentioned above are met.
- For regular spatial sampling of two or more spatial dimensions using a chessboard pattern design, band-limited MWNI can reconstruct aliased spatial dimensions utilizing the information from the unaliased dimensions. This is also explored by Trad (2008).

Limitations

- The major failure of Fourier reconstruction comes with regularly sampled and aliased data. The alias in this case interfere with the original spectrum of data. Therefore, MWNI or band-limited MWNI methods can not discriminate between true and artificial events.
- A proper spatial (and sometimes temporal) windowing strategy is required for MWNI method in order to satisfy the linear event assumption. The presence of curved events will jeopardize the performance of MWNI.

7.2 MSAR reconstruction

MSAR reconstruction is an effort to use the low frequencies for an aliased-free reconstruction of the high frequencies. The low frequency portion of data is spatially band-limited and can be reconstructed using band-limited Fourier reconstruction methods. The multistep autoregressive algorithm provides the opportunity to estimate the prediction filters for high frequencies by just using a band of low frequency data. The estimated prediction filters can be used in the f-x domain to reconstruct the data (MSAR-X). One can use the AR parametric spectrum of the data to build band-pass filters that can be used by a Fourier reconstruction algorithm (MSAR-K). Overall, the performance of MSAR-K is superior to MSAR-X as well as any stand-alone Fourier reconstruction method.

Advantages

- The MSAR method is capable of eliminating sampling artifacts in the Fourier domain and, therefore, it is not effected by the type of sampling used to acquired the data.

Limitations

- MSAR requires the original events in the t-x domain to be linear. Consequently, spatial windowing of nonlinear events is needed.
- The seismic source wavelet should be invariant in the window of analysis (no dispersion).
- The MSAR method does require quite a complex algorithm and its extension to N-D is not an easy task.

7.3 Adaptive f-x interpolation

Curved seismic events in the t-x domain transform into non-stationary signals in the f-x domain. Prediction filters as they were proposed by Spitz (1991) are powerful spectrum estimations for stationary signals (linear events in the t-x domain). Adaptive f-x interpolation aims to find a local prediction filter for each spatial location.

Advantages

- Adaptive *f-x* interpolation does not need any spatial windowing for optimal reconstruction.
- It performs quite well for data with laterally variant dips as well as spatillay variant wavelets.
- The performance of the proposed method depends on two factors: the length of the prediction filter and a forgetting factor. One can estimate these parameters automatically from the original data and therefore obtain a fully-automated interpolation method.

Limitations

- The proposed adaptive interpolation method is only applicable for regularly sampled data. Irregularly missing samples in the original data can lead to an inaccurate estimation of the prediction filters.
- At the moment, the method has only been applied to 2D data (1D spatial prediction). However, theoretically it can also be extended to the multidimensional case.
- The method could become quite demanding on memory if extended to the multidimensional case.

7.4 Final remarks and recommendations

Real seismic data depends on 4 spatial dimensions and time or frequency. To fully exploit the benefit of the algorithms presented in this thesis, one should work with 4 spatial dimensions at the time. This increases the number of data points leading to a more reliable reconstruction. It is also true that working with 5D data volumes (4 spatial dimensions and frequency) introduces operational problems that can only be addressed by industrial strategies that are often beyond the scope of the type of research carried out at a university. A typical 3D survey consists of a 5D data volume composed of several terabytes of data. The algorithms proposed in this thesis have proven their applicability to multidimensional data reconstruction with synthetic and field data examples. More work, however, is needed to implement the algorithms in an industrial production mode. Particular attention needs to be paid to input/output strategies since I believe this is the real bottle neck of the seismic reconstruction problem.

My research has mainly focused on reconstruction algorithms and I have not studied the problem of preconditioning the data prior to reconstruction. In particular, land data are severely contaminated with coherent noise (surface waves) that needs to be removed prior to reconstruction. It is often the case that noise removal algorithms require well-sampled data. On the other hand, coherent noise could deteriorate the reconstruction if it is not properly canceled. It is clear that more research is needed to tackle the regularization problem and the coherent noise suppression problem simultaneously. This should be an important research avenue for improving the work presented in this thesis. Land seismic data are also affected by elevation statics (receivers and sources are often placed on an irregular surface). Handling the static problem in conjunction with the regularization problem is not an easy task and consequently, these problems are treated separately. To my knowledge little has been made in attempts to integrate reconstruction, coherent noise removal and static correction under a single procedure for seismic signal processing. I believe that future research on seismic data reconstruction for land data should concentrate on integrating the aforementioned problems.

Bibliography

Abma, R. and N. Kabir (2005). Comparison of interpolation algorithms. *The Leading Edge 24*(10), 984–989.

Cadzow, J. A. and K. Ogino (1981). Two-dimensional spectral estimation. *IEEE Transactions on Acoustics, Speech, and Signal processing 29*(3), 396–401.

Canales, L. L. (1984). Random noise reduction. *54th Annual International Meeting, SEG, Expanded Abstarcts*, Session:S10.1.

Claerbout, J. (1998). Multidimensional recursive filters via a helix. *Geophysics 63*(5), 1532–1541.

Claerbout, J. F. and S. Fomel (2006). Image estimation by examples. http://sepwww.stanford.edu/sep/prof/gee/lal/paper_html/node1.html.

Crawley, S., R. Clapp, and J. Clearbout (1999). Interpolation with smoothly nonstationary prediction-error filters. *69th Annual International Meeting, SEG, Expanded Abstarcts*, 1154–1157.

Dahlquist, G. and A. Bjorck (Eds.) (2006). *Numerical Methods in Scientific Computing*. Philadelphia: Society for Industrial and Applied Mathematics.

Darche, G. (1990). Spatial interpolation using a fast parabolic transform. *60th Annual International Meeting, SEG, Expanded Abstarcts*, 1647–1650.

Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory 52*(4), 1289–1306.

Duijndam, A. J. W., M. A. Schonewille, and C. O. H. Hindriks (1999). Reconstruction of band-limited signals, irregularly sampled along one spatial direction. *Geophysics 64*(2), 524–538.

Eldar, Y. C. (2006). Mean-squared error sampling and reconstruction in the presence of noise. *IEEE Transactions on Signal Processing 54*(12), 4619–4633.

Feuer, A. and G. C. Goodwin (2005). Reconstruction of multidimensional bandlimited signals from nonuniform and generalized samples. *IEEE Transactions on Signal Processing 53*(11), 4273–4282.

Fomel, S. (2003). Seismic reflection data interpolation with differential offset and shot continuation. *Geophysics 68*(2), 733–744.

Franz, S., S. K. Mitraa, and G. Doblingerc (2003). Frequency estimation using warped discrete fourier transform. *Signal Processing 83*, 1661–1671.

Gulunay, N. (2003). Seismic trace interpolation in the Fourier transform domain. *Geophysics 68*(1), 355–369.

Hager, W. W. (1989). Updating the inverse of a matrix. *SIAM Review 31*(2), 221–239.

Haining, R. (Ed.) (2003). *Spatial Data Analysis: Theory and Practice*. Cambridge University Press.

Hennenfent, G. and F. J. Herrmann (2006). Application of stable signal recovery to seismic data interpolation. *76th Annual International Meeting, SEG, Expanded Abstarcts*, 2797–2801.

Hennenfent, G. and F. J. Herrmann (2007). Random sampling: new insights into the reconstruction of coarsely-sampled wavefields. *77th Annual International Meeting, SEG, Expanded Abstarcts*, 2575–2579.

Hennenfent, G. and F. J. Herrmann (2008). Simply denoise: Wavefield reconstruction via jittered undersampling. *Geophysics 73*(3), V19–V28.

Herrmann, F. J. and G. Hennenfent (2008). Non-parametric seismic data recovery with curvelet frames. *Geophysical Journal International 173*, 233–248.

Herrmann, F. J., D. Wang, G. Hennenfent, and P. P. Moghaddam (2008). Curvelet-based seismic data processing: A multiscale and nonlinear approach. *Geophysics 73*(1), A1–A5.

Hestenes, M. R. and E. Stiefel (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards 49*(6), 409–436.

Honig, M. and D. Messerschmidt (1984). *Adaptive Filters: Structures, Algorithms and Applications*. Boston, MA: Kluwer.

Kumaresan, R. and D. W. Tufts (1981). A two-dimensional technique for frequency-wavenumber estimation. *Proceedings of the IEEE 69*(11), 1515–1517.

Lau, S., P. J. Sherman, and L. B. White (2002). Asymptotic statistical properties of AR spectral estimators for processes with mixed spectra. *IEEE Transactions on Information Theory 48*(4), 909–917.

Liu, B. (2004). *Multi-dimensional Reconstruction of Seismic Data*. Ph. D. thesis, University of Alberta.

Liu, B. and M. D. Sacchi (2004). Minimum weighted norm interpolation of seismic records. *Geophysics 69*(6), 1560–1568.

Liu, B., M. D. Sacchi, and D. Trad (2004). Simultaneous interpolation of 4 spatial dimensions. *74th Annual International Meeting, Soc. of Expl. Geophys.*, 2009–2012.

Ma, C. (2004). Spatial autoregression and related spatio-temporal models. *Journal of Multivariate Analysis 88*, 152–162.

Makur, A. and S. K. Mitra (2001). Warped discrete-fourier transform: Theory and applications. *IEEE Transactions on Circuits and SystemsI: Fundamental Theory and Applications 48*(9), 1086–1093.

Malcolm, A. E., M. V. de Hoop, and J. H. LeRousseau (2005). The applicability of dip moveout/azimuth moveout in the presence of caustics. *Geophysics 70*(1), S1–S17.

Markel, J. and A. Gray (Eds.) (1976). *Linear Prediction of Speech*. Springer-Verlag.

Marple, L. (1980). A new autoregressive spectrum analysis algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing 28*(4), 342–349.

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

McClellan, J. H. (1982). Multidimensional spectral estimation. *Proceedings of the IEEE 70*(9), 1029–1039.

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

Naghizadeh, M. and M. D. Sacchi (2008). Sampling functions and sparse reconstruction methods. *EAGE Conference, Rome, Italy*.

Naghizadeh, M. and M. D. Sacchi (2009a). *f-x* adaptive seismic-trace interpolation. *Geophysics 74*(1), V9–V16.

Naghizadeh, M. and M. D. Sacchi (2009b). Robust reconstruction of aliased data using autoregressive spectral estimates. *EAGE Conference, Amsterdam, Netherlands*.

Naghizadeh, M. and M. D. Sacchi (2009c). Sampling considerations for band-limited fourier reconstruction of aliased seismic data. *EAGE Conference, Amsterdam, Netherlands*.

Naghizadeh, M. and M. D. Sacchi (2009d). Seismic data reconstruction using multidimensional prediction filters. *Geophysical Prospecting In Press*, doi: 10.1111/j.1365–2478.2009.00805.x.

Narasimhan, S. V. and S. Veena (Eds.) (2005). *Signal Processing: Principles and Implementation*. Alpha Science International Ltd.

Oppenheim, A. V. and R. W. Schafer (Eds.) (1974). *Digital Signal processing*. Prentice-Hall Inc.

Papoulis, A. (Ed.) (1991). *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill.

Picinbono, B. and M. Benidir (1986). Some properties of lattice autoregressive filters. *IEEE Transactions on Acoustics, Speech and Signal Processing 34*(2), 342–349.

Porsani, M. (1999). Seismic trace interpolation using half-step prediction filters. *Geophysics 64*(5), 1461–1467.

Priestly, M. B. (Ed.) (1988). *Non-linear and non-stationary time series analysis*. London: Academic Press.

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

Ronen, J. (1987). Wave-equation trace interpolation. *Geophysics 52*(7), 973–984.

Russ, J. A., D. W. Casbeer, and A. L. Swindlehurst (2004). STAP detection using space-time autoregressive filtering. *Radar Conference, 2004. Proceedings of the IEEE*, 541–545.

Saad, Y. (Ed.) (2004). *Iterative Methods for Sparse Linear Systems*. Philadelphia: Society for Industrial and Applied Mathematics.

Sacchi, M. and H. Kuehl (2000). *FX ARMA* filters. *70th Annual International Meeting, SEG, Expanded Abstarcts*, 2092–2095.

Sacchi, M. and B. Liu (2005). Minimum weighted norm wavefield reconstruction for AVA imaging. *Geophysical Prospecting 53*(6), 787–801.

Sacchi, M. D. and T. J. UIrych (1996). Estimation of the discrete fourier transform, a linear inversion approach. *Geophysics 61*(4), 1128–1136.

Sacchi, M. D., T. J. Ulrych, and C. J. Walker (1998). Interpolation and extrapolation using a high-resolution discrete fourier transform. *IEEE Transaction on Signal Processing 46*(1), 31–38.

Sacchi, M. D., D. J. Verschuur, and P. M. Zwartjes (2004). Data reconstruction by generalized deconvolution. *74th Annual International Meeting, SEG, Expanded Abstarcts*, 1989–1992.

Schonewille, M. A., R. Romijn, A. J. W. Duijndam, and L. Ongkiehong (2003). A general reconstruction scheme for dominant azimuth 3D seismic data. *Geophysics 68*(6), 2092–2105.

Slepian, D. (1976). On bandwidth. *Proceedings of IEEE 64*(3), 292–300.

Soubaras, R. (1994). Signal-preserving random noise attenuation by the F-X projection. *64th Annual International Meeting, SEG, Expanded Abstarcts*, 1576–1579.

Spitz, S. (1990). 3-D seismic interpolation in the F-XY domain. *SEG Technical Program Expanded Abstracts*, 1641–1643.

Spitz, S. (1991). Seismic trace interpolation in the *F-X* domain. *Geophysics 56*(6), 785–794.

Sroubek, F., G. Cristobal, and J. Flusser (2007). A unified approach to superresolution and multichannel blind deconvolution. *IEEE Trans. Image Processing 16*(9), 2322–2332.

Stolt, R. H. (2002). Seismic data mapping and reconstruction. *Geophysics 67*(3), 890–908.

Strohmer, T. (1997). Computationally attractive reconstruction of bandlimited images from irregular samples. *IEEE Transactions on Image Processing 6*(4), 540–548.

Takalo, R., H. Hytti, and H. Ihalainen (2005). Tutorial on univariate autoregressive spectral analysis. *Journal of Clinical Monitoring and Computing 19*, 401–410.

Tarantola, A. (Ed.) (2005). *Inverse Problem Theory and Methods for Model Parameter Estimation*. Philadelphia: Society for Industrial and Applied Mathematics.

Thomson, D. J. (1982). Spectrum estimation and harmonic analysis. *Proceedings of IEEE 70*(9), 1055–1096.

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

Trad, D. (2008). Five dimensional seismic data interpolation. *78th Annual International Meeting, SEG, Expanded Abstarcts*, 978–981.

Trad, D., T. J. Ulrych, and M. D. Sacchi (2002). Accurate interpolation with high-resolution time-variant radon transforms. *Geophysics 67*(2), 644–656.

Tsaig, Y. and D. L. Donoho (2006). Extensions of compressed sensing. *Signal Processing 86*, 549–571.

Ulrych, T. J. and R. W. Clayton (1976). Time series modelling and maximum entropy. *Phys. Earth Planetary Interiors 12*, 188–200.

Unser, M. (2000). Sampling - 50 years after shannon. *Proceedings of the IEEE 88*(4), 569–587.

Vaidyanathan, P. P. (Ed.) (2008). *The Theory of Linear Prediction*. Morgan & Claypool.

Vandewalle, P., L. Sbaiz, J. Vandewalle, and M. Vetterli (2007). Super-resolution from unregistered and totally aliased signals using subspace methods. *IEEE Transactions on Signal Processing 55*(7), 3687–3703.

Wang, Y. (2002). Seismic trace interpolation in the f-x-y domain. *Geophysics 67*(4), 1232–1239.

Welch, P. D. (1967). The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Trans. Audio Electroacoust. 15*(2), 70–73.

Wiggins, R. A. and S. D. Miller (1972). New noise-reduction technique applied to long-period oscillations from the Alaska earthquake. *Bull. Seism. Soc. Am. 62*, 471–479.

Zwartjes, P. and A. Gisolf (2006). Fourier reconstruction of marine-streamer data in four spatial coordinates. *Geophysics 71*(6), V171–V186.

Zwartjes, P. and M. D. Sacchi (2007). Fourier reconstruction of nonuniformly sampled, aliased seismic data. *Geophysics 72*(1), V21–V32.

Zwartjes, P. M. (2005). *Fourier reconctruction with sparse inversion*. Ph. D. thesis, Delft University of Technology.

APPENDIX A

Proof of multistep autoregressive operators

In this appendix I prove the multistep autoregressive (MSAR) method for a 2D case. A multidimensional, as well as a 1D proof, can be easily derived by following the same approach. Consider a cube of 3D data in the t-x-y domain composed of K dipping events (plane waves). In the f-x-y domain each waveform can be characterized by its ray parameter vector $\mathbf{p}_k$ that can be decomposed into two components p_{x_k} and p_{y_k}. For a given frequency $n\Delta f$, we have a 2D spatial signal, G, that can be represented as

$$\begin{aligned} G(l,m,n) &= \sum_{k=1}^{K} A_k(n\Delta f) e^{-i2\pi(n\Delta f)(l\Delta x p_{x_k} + m\Delta y p_{y_k})} \\ &= \sum_{k=1}^{K} A_k(n\Delta f) e^{-i\Theta_k(l,m,n)}, \end{aligned} \tag{A.1}$$

where Θ_k and A_k represent the amount of phase shift and amplitude of the wavelet for the kth event, respectively. Δx and Δy are the sampling intervals in the spatial directions and Δf is the sampling interval on the frequency axis. The integer values l, m and n are used to indicate the sample numbers in x, y, and f directions, respectively. Since the signal G consists of a finite superposition of harmonics, it can be well represented by the following 2D autoregressive operators of size $R \times S$ (Kumaresan and Tufts, 1981; Cadzow and Ogino,

1981; McClellan, 1982):

$$G(l,m,n) = \sum_{r=0}^{R} \sum_{\substack{s=0 \text{ if } r\neq 0 \\ s=1 \text{ if } r=0}}^{S} P(r,s,n\Delta f) \times G(l-r, m-s, n)\,. \tag{A.2}$$

Now consider the case where $\Delta x' = \alpha\Delta x$, $\Delta y' = \alpha\Delta y$ and $\Delta f' = \Delta f/\alpha$ where α is an integer. Rewriting equation A.2 for $\Delta x'$, $\Delta y'$ and $\Delta f'$ leads to

$$G(\alpha l, \alpha m, \frac{n}{\alpha}) = \sum_{r=0}^{R} \sum_{\substack{s=0 \text{ if } r\neq 0 \\ s=1 \text{ if } r=0}}^{S} P'(r,s,n\frac{\Delta f}{\alpha}) \times G(\alpha(l-r), \alpha(m-s), \frac{n}{\alpha}). \tag{A.3}$$

In addition, the following expression can be derived for the phase function:

$$\begin{aligned} \Theta_k(\alpha l, \alpha m, \frac{n}{\alpha}) &= 2\pi(\frac{n\Delta f}{\alpha})(l\alpha\Delta x p_{x_k} + m\alpha\Delta y p_{y_k}) \\ &= 2\pi(n\Delta f)(l\Delta x p_{x_k} + m\Delta y p_{y_k}) \\ &= \Theta_k(l,m,n)\,. \end{aligned} \tag{A.4}$$

It is important to stress that in equations (A.2) and (A.3) the amplitude of the signal, A_k, is canceled out because of the presence of G on both sides of the equations. Therefore, expressions (A.2), (A.3) and (A.4) lead to the following

$$P'(r,s,n\frac{\Delta f}{\alpha}) = P(r,s,n\Delta f)\,. \tag{A.5}$$

Equation (A.5) suggests that prediction filters extracted from frequency f/α using an α step auto-regressive operator can be used to reconstruct the spatial data at frequency f. For 2D spatial data of size $L \times M$ and a 2D autoregressive operator of order $R \times S$, the maximum number of steps (α) that one can use is determined by the following expression

$$\alpha_{max} = \min\left[\lfloor\frac{L-\frac{R+1}{2}}{R}\rfloor, \lfloor\frac{M-\frac{S+1}{2}}{S}\rfloor\right], \tag{A.6}$$

where, $\lfloor\rfloor$ means truncation to the closest smaller integer number. For the multidimensional spatial reconstruction problem one needs to use multidimensional autoregressive operators in the same way as discussed for the 2D case.

APPENDIX B

Multidimensional convolution using 1D operators

In this appendix, I will present a simple algorithm to perform multidimensional convolution using one-dimensional convolution operators. The proposed algorithm can be easily used to perform the multidimensional autoregressive operation needed by the MSAR reconstruction algorithm. The technique introduced here is very similar to the helix coordinate representation introduced by Claerbout (1998).

Let's start with the two-dimensional (2D) case. First, I will assume the convolution of $\mathbf{X}_{M\times N}$ with $\mathbf{Y}_{K\times L}$. The result of ordinary 2D discrete convolution, $\mathbf{Z}_{(M+K-1)\times(N+L-1)}$, is obtained give by

$$Z(i,j) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} X(m,n).Y(i-m,j-n) \tag{B.1}$$

where $0 \leq i < M+K-1$ and $0 \leq j < N+L-1$. Next, the following two matrices are introduced

$$\mathbf{X}'_{(M+K-1)\times(N+L-1)} = \begin{pmatrix} \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \mathbf{Y}'_{(M+K-1)\times(N+L-1)} = \begin{pmatrix} \mathbf{Y} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}. \tag{B.2}$$

Now one can order in lexicographic form the vectors $\mathbf{X}''$ and $\mathbf{Y}''$ from the matrices $\mathbf{X}'$ and $\mathbf{Y}'$, respectively. The vectors are truncated in a way that their last sample is the last sample of the original data $\mathbf{X}$ and $\mathbf{Y}$, respectively. This means the lengths of $\mathbf{X}''$ and $\mathbf{Y}''$ are given by $(N-1)\times(M+K-1)+M$ and $(L-1)\times(M+K-1)+K$, respectively. I have adopted column-wise lexicographic ordering but the row-wise lexicographic ordering

a)

1	5	9
2	6	10
3	7	11
4	8	12

b)

1	9	17	25	33	41	49
2	10	18	26	34	42	50
3	11	19	27	35	43	51
4	12	20	28	36	44	52
5	13	21	29	37	45	53
6	14	22	30	38	46	54
7	15	23	31	39	47	55
8	16	24	32	40	48	56

c)

1	12	23	34	45	56	67	78	89
2	13	24	35	46	57	68	79	90
3	14	25	36	47	58	69	80	91
4	15	26	37	48	59	70	81	92
5	16	27	38	49	60	71	82	93
6	17	28	39	50	61	72	83	94
7	18	29	40	51	62	73	84	95
8	19	30	41	52	63	74	85	96
9	20	31	42	53	64	75	86	97
10	21	32	43	54	65	76	87	98
11	22	33	44	55	66	77	88	99

d)

1	12	23	34	45	56	67	78	89
2	13	24	35	46	57	68	79	90
3	14	25	36	47	58	69	80	91
4	15	26	37	48	59	70	81	92
5	16	27	38	49	60	71	82	93
6	17	28	39	50	61	72	83	94
7	18	29	40	51	62	73	84	95
8	19	30	41	52	63	74	85	96
9	20	31	42	53	64	75	86	97
10	21	32	43	54	65	76	87	98
11	22	33	44	55	66	77	88	99

e)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

f)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	...	71	72	73	74

Figure B.1: A scheme showing the 2D convolution of two matrices using 1D convolution. a) and b) are the two signals to be convolved. c) and d) Signals after zero-padding. e) and f) Signals after lexicographic ordering and truncation. The 1D convolution of (e) and (f) followed by reshaping is equivalent to the 2D convolution of (a) and (b).

can be used as well. The 1D convolution of $\mathbf{X}^{''}$ and $\mathbf{Y}^{''}$ will produce $\mathbf{Z}^{''}$. It is interesting to see that the length of $\mathbf{Z}^{''}$ is

$$\begin{aligned} l_{Z''} &= l_{X''} + l_{Y''} - 1 \\ &= [(N-1)\times(M+K-1)+M] + [(L-1)\times(M+K-1)+K] - 1 \\ &= (M+K-1)\times(N-1+L-1)+M+K-1 \\ &= (M+K-1)\times(N+L-1), \end{aligned} \tag{B.3}$$

which, surprisingly, is equal to the size of the matrix $\mathbf{Z}$. Therefore, the last step is to reshape the vector $\mathbf{Z}^{''}$ (column-wise in this case) into a $(M+K-1)\times(N+L-1)$ size matrix.

Figure B.1 shows a simple cartoon of the proposed algorithm. The aim is to convolve $\mathbf{X}_{4\times3}$, shown by black cells (Figure B.1a), with $\mathbf{Y}_{8\times7}$ shown by white cells (Figure B.1b). The numbers inside the cells represent the index of the data sample. Figures B.1c and B.1d represent Figures B.1a and B.1b after zero-padding. The gray cells represent the zero values. Figures B.1e and B.1f ($\mathbf{X}^{''}$ and $\mathbf{Y}^{''}$) are created by column-wise lexicographic ordering of Figures B.1c and B.1d ($\mathbf{X}^{'}$ and $\mathbf{Y}^{'}$). Notice that the last samples of $\mathbf{X}^{''}$ and $\mathbf{Y}^{''}$ are 26 and 74, respectively. The 1D convolution of Figures B.1e and B.1f, followed by reshaping leads to a matrix of size 11×9. This is the desired 2D convolution.

The above strategy can be easily extended for higher dimensions. Figure B.2 shows how one can reduce a 3D convolution to a 1D convolution. Here, $\mathbf{X}_{2\times2\times3}$, shown by black cells (Figure B.2a), is to be convolved with $\mathbf{Y}_{3\times3\times3}$, shown by white cells (Figure B.2b). Figures B.2c and B.2d represent Figures B.2a and B.2b after zero-padding, respectively. Figures B.2e and B.2f show the column-wise lexicographically ordered versions of Figures B.2c and B.2d, respectively. By convolving Figure B.2e with Figure B.2f we obtain a vector with length equal to $38+43-1=80$. Reshaping the resulting 1D convolution into a 3D data set of size $4\times4\times5$ gives the result for the desired 3D convolution.

To further improve the proposed algorithm one can use sparse 1D convolution algorithms to speed up the operation time. Sparse 1D convolution algorithms are designed to avoid multiplying and summing zero entries. For instance, in Figures B.1e and B.2e the signal contains an excessive amount of zeros (gray cells). Multiplying them by the correspondent value of the other signal (Figures B.1f and B.2f) and adding them during convolution does not contribute to the final result. Therefore, it is desired to avoid operations that involve zeros. The solution is to save the indices of the non-zero values in Figures B.1e and B.2e (black cells) and rewrite a 1D convolution algorithm that only uses the non-zero entries. The latter will improve the computational efficiency of the 1D convolution algorithm.

Linear prediction or autoregressive modeling for multidimensional data processing can also be implemented using one-dimensional operators. In essence, linear prediction entails pre-

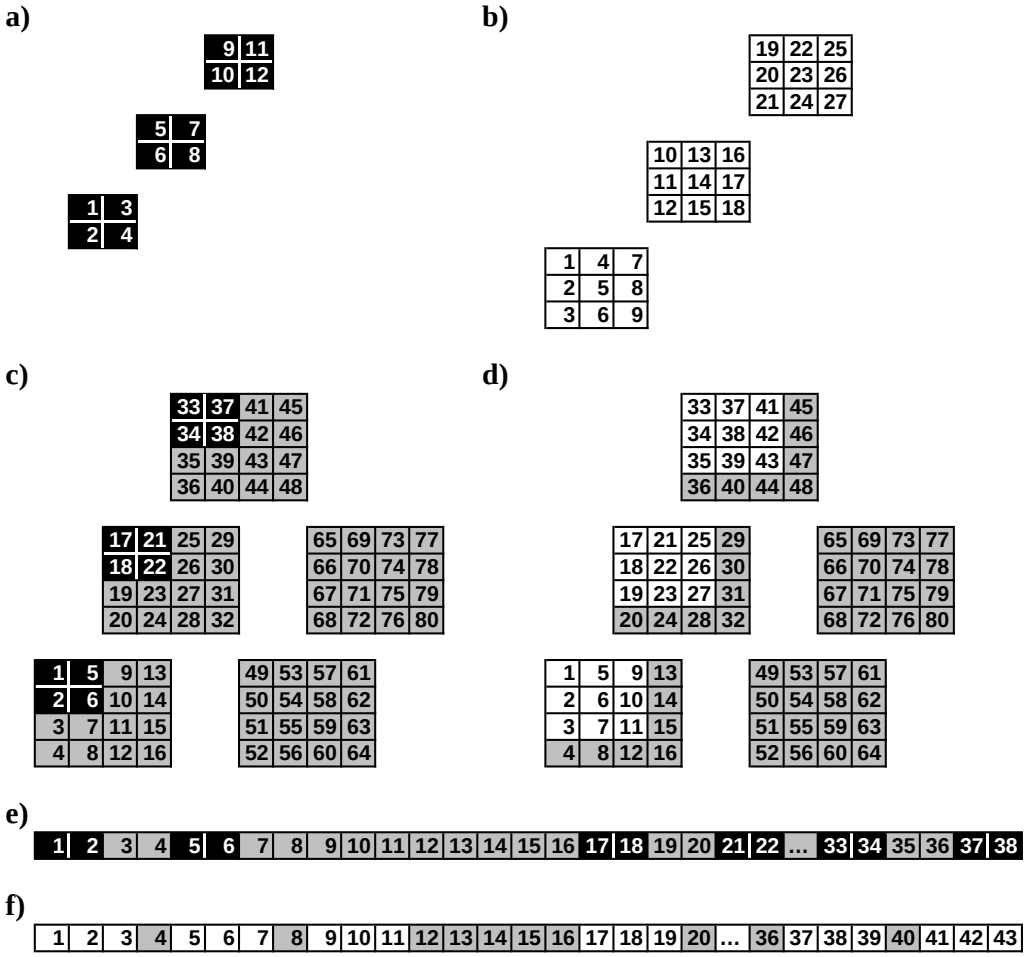

Figure B.2: A scheme showing the 3D convolution of two cube of data using 1D convolution. a) and b) are the two signals to be convolved. c) and d) Signals after zero-padding. e) and f) Signals after lexicographic ordering and truncation. The 1D convolution of (e) and (f) followed by reshaping is equivalent to the 3D convolution of (a) and (b).

dicting an observation by linearly transforming the nearby observation via convolution. Therefore, when working with multidimensional data one can transform multidimensional linear prediction into 1D linear prediction as well (Naghizadeh and Sacchi, 2009c). The MSAR code developed in the package `SeisRec` (Appendix D) uses 1D convolution to perform ND convolution. The idea is to have a single algorithm capable of handing 1D, 2D, 3D and 4D spatial reconstruction.

APPENDIX C

Proof of matrix inversion lemma

Consider an augmented matrix X composed of the matrices $\mathbf{A}_{m\times m}$, $\mathbf{B}_{m\times n}$, $\mathbf{C}_{n\times m}$, and $\mathbf{D}_{n\times n}$ as follows

$$\mathbf{X} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}. \tag{C.1}$$

Let's assume that the inversion of matrix $\mathbf{X}$ is matrix $\mathbf{Y}$ which is composed of the matrices $\mathbf{E}_{m\times m}$, $\mathbf{F}_{m\times n}$, $\mathbf{G}_{n\times m}$, and $\mathbf{H}_{n\times n}$. Then, we have

$$\mathbf{X}^{-1} = \mathbf{Y} \Rightarrow \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{pmatrix}. \tag{C.2}$$

The multiplication of a matrix and its inverse should give an identity matrix, namely $\mathbf{XY} = \mathbf{I}_{(m+n)\times(m+n)}$, or

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} \begin{pmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{pmatrix} = \begin{pmatrix} \mathbf{I}_{m\times m} & \mathbf{0}_{m\times n} \\ \mathbf{0}_{n\times m} & \mathbf{I}_{n\times n} \end{pmatrix}. \tag{C.3}$$

Expression C.3 can be expanded as

$$\mathbf{XY} = \begin{pmatrix} \mathbf{AE} + \mathbf{BG} = \mathbf{I}_{m\times m} & \mathbf{AF} + \mathbf{BH} = \mathbf{0}_{m\times n} \\ \mathbf{CE} + \mathbf{DG} = \mathbf{0}_{n\times m} & \mathbf{CF} + \mathbf{DH} = \mathbf{I}_{n\times n} \end{pmatrix}. \tag{C.4}$$

where, from Expressions $\mathbf{0}_{n\times m}$ and $\mathbf{0}_{m\times n}$ we obtain, respectively

$$\mathbf{G} = -\mathbf{D}^{-1}\mathbf{CE}, \tag{C.5}$$

$$\mathbf{F} = -\mathbf{A}^{-1}\mathbf{BH}. \tag{C.6}$$

where, $\mathbf{D}$ and $\mathbf{A}$ have to be nonsingular matrices. Now, by substituting (C.5) into the $\mathbf{I}_{m\times m}$ term of Expression (C.4) we have:

$$\mathbf{E} = (\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}, \tag{C.7}$$

and similarly by substituting (C.6) into the $\mathbf{I}_{n\times n}$ term of Expression (C.4) we get:

$$\mathbf{H} = (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}. \tag{C.8}$$

By replacing Expressions (C.5), (C.6), (C.7) and (C.8) into Expression C.2 we obtain

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^{-1} = \begin{pmatrix} (\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & -\mathbf{A}^{-1}\mathbf{B}\mathbf{H} \\ -\mathbf{D}^{-1}\mathbf{C}\mathbf{E} & (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{pmatrix}$$

which further simplifies to

$$\begin{pmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{pmatrix} = \begin{pmatrix} (\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{pmatrix} \tag{C.9}$$

Now, consider the following multiplication of matrices:

$$\begin{aligned} \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} \begin{pmatrix} \mathbf{I} & -\mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} &= \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B} \end{pmatrix} \begin{pmatrix} \mathbf{I} & -\mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B} \end{pmatrix}. \end{aligned} \tag{C.10}$$

By inverting both side of the Expression (C.10) we have

$$\begin{pmatrix} \mathbf{I} & -\mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{pmatrix}$$

and using matrix multiplication properties

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{I} & -\mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{pmatrix}.$$

With further simplification and remembering that $\mathbf{X}^{-1} = \mathbf{Y}$, we have

$$\begin{pmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{pmatrix} = \begin{pmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{pmatrix} \tag{C.11}$$

Equating $\mathbf{E}$ terms from expressions (C.9) and (C.11) we obtain a general formula for matrix inversion lemma:

$$(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1}. \tag{C.12}$$

For the specific case of $\mathbf{C} = \mathbf{B}^H$ Expression (C.12) reduces to

$$(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{B}^H)^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{B}^H\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^H\mathbf{A}^{-1}. \tag{C.13}$$

This is the exact expression used for deriving the recursive estimation of prediction filters in Chapter 6. The latter is also called the Sherman-Morrison-Woodbury formula (see for instance Hager (1989)).

APPENDIX D

The reconstruction package SeisRec

Many of the results presented in this thesis were computed with the package `SeisRec`. This is a C package that I have created in order to perform multi-dimensional seismic data reconstruction. `SeisRec` can handle N-D cubes of irregularly sampled data (input) and produce a regularly sampled data cube. It works with aliased data and it is flexible enough to accommodate to 1D, 2D, 3D, and 4D spatial interpolation problems. Table D.1 describes some of the key features of `SeisRec`. For completeness I have added a reference to the MATLAB package that I have written for adaptive interpolation. The latter, at present time, has not been included into `SeisRec`. The `SeisRec` package also contains synthetic and real data examples that permit to reproduce many of the results in this thesis.

Table D.1: List of computer programs developed for this thesis

	1D	2D	3D	4D	Language	Package
MWNI	✓	✓	✓	✓	C	SeisRec
MSAR-X	✓	✓	✓	✓	C	SeisRec
MSAR-K	✓	✓	✓	✓	C	SeisRec
ADAPTIVE f-x	✓				MATLAB	
Gulunay F-K	✓	✓	✓	✓	C	SeisRec