



M.Sc. Thesis

Wavenumber-frequency spectrum estimation of ambient seismic noise

Yukang Tu

Abstract

In oil and gas exploration, seismic arrays are deployed by geophysicists to image the subsurface. For passive seismic applications, the data recorded by the sensor array may contain velocity and angle information of the propagating seismic wave. This information can be used to infer the properties of material in different earth layers. In order to find the velocity and arrival angle, beamforming algorithms are applied to estimate the wavenumber-frequency spectrum for the seismic signals. The propagating seismic wave field consists of body waves and surface waves. In some applications, surface waves are interpreted as noise, thus filtering is required to remove the surface waves before or during the implementation of beamforming algorithms. In this thesis, we first introduce a data model. Then several beamforming algorithms based on the data model are discussed, and the performance of the different algorithms is evaluated. Capon beamforming as adopted in seismics has limitations. Robust Capon beamforming which can overcome these limitations is explained in the thesis. For filtering of the surface waves, we propose to first reconstruct the irregularly sampled spatial signal into a uniform array, then design a velocity filter to remove the unwanted low-speed noise (surface waves).

Wavenumber-frequency spectrum estimation of ambient seismic noise

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

by

Yukang Tu

born in Shaoxing, P.R.China

This work was performed in:

Exploratory Research (EPT-RXX)

Shell International Exploration and Production B.V.

Kessler Park 1, 2288 GS, Rijswijk, The Netherlands



Delft University of Technology

Copyright © 2009 Circuits and Systems Group
All rights reserved.

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
TELECOMMUNICATIONS

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled “**Wavenumber-frequency spectrum estimation of ambient seismic noise**” by **Yukang Tu** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 30th/June/2009

Chairman:

Prof.dr.ir. Alle-Jan Van Der Veen

Advisor:

Dr.ir. Xander Campman

Committee Members:

Prof.dr.ir. Geert Leus

Prof.dr.ir. A. Gisolf

Abstract

In oil and gas exploration, seismic arrays are deployed by geophysicists to image the subsurface. For passive seismic applications, the data recorded by the sensor array may contain velocity and angle information of the propagating seismic wave. This information can be used to infer the properties of material in different earth layers.

In order to find the velocity and arrival angle, beamforming algorithms are applied to estimate the wavenumber-frequency spectrum for the seismic signals. The propagating seismic wave field consists of body waves and surface waves. In some applications, surface waves are interpreted as noise, thus filtering is required to remove the surface waves before or during the implementation of beamforming algorithms.

In this thesis, we first introduce a data model. Then several beamforming algorithms based on the data model are discussed, and the performance of the different algorithms is evaluated. Capon beamforming as adopted in seismics has limitations. Robust Capon beamforming which can overcome these limitations is explained in the thesis. For filtering of the surface waves, we propose to first reconstruct the irregularly sampled spatial signal into a uniform array, then design a velocity filter to remove the unwanted low-speed noise (surface waves).

Acknowledgments

I would like to thank Dr Xander Campman, my supervisor in Shell, for his advice and daily supervision on my thesis project. I appreciate his generous and kindness, and his constructive suggestions always kept me motivated during my internship.

I would also like to thank Prof Alle-Jan Van Der Veen for his advices, and I am grateful for him to spend much time on my thesis project, lots of discussions we had with him are very helpful.

Finally I am very thankful to Prof Geert Leus and Prof A. Gisolf, for their participation in my thesis commitee, and their comments on my work are all appreciated.

Yukang Tu

Delft, The Netherlands

30th/June/2009

Contents

Abstract	v
Acknowledgments	vii
List of figures	xi
1 Introduction	1
1.1 Seismic array processing	1
1.1.1 Beamforming	2
1.1.2 Filtering	3
1.2 Objectives	3
1.2.1 Thesis goals	3
1.2.2 Thesis organization	3
2 Data Model	5
2.1 Introduction	5
2.2 Frequency Domain Data Model	7
2.2.1 Fourier Transform theory	7
2.2.2 Formation of the Data Model	8
2.2.3 Practical Issues	10
2.3 Conventional Beamforming	12
2.3.1 Beamforming theory	12
2.3.2 Implementation	13
2.3.3 Simulations	14
2.4 Conclusion	15
3 Robust Capon Beamforming	17
3.1 Introduction	17
3.2 Standard Capon Beamforming	17

3.2.1	Problem Statement	17
3.2.2	Solution	18
3.2.3	Numerical Results	19
3.3	Robust Capon Beamforming	20
3.3.1	Problems of Capon Beamforming	20
3.3.2	Derivation of the Robust Capon Beamforming(RCB) algorithm	20
3.3.3	Simulation Results	26
3.4	Conclusion	30
4	Reconstruction and filtering	33
4.1	Problem Statement	33
4.2	Solution	34
4.2.1	Data Model	34
4.2.2	Estimate the wavenumber-frequency spectrum	36
4.2.3	Reconstruct the spatial signal in a uniform grid	39
4.2.4	Filter the seismic signal	39
4.2.5	Practical issue: Implementation of the FRSI method	41
4.2.6	Simulations	43
4.3	Conclusion	45
5	Conclusion and Future Work	51
5.1	Recommendations for Future Work	52
	Bibliography	53

List of Figures

1.1	An example seismic array and its beamforming output	2
2.1	Decomposition of the body wave slowness vector \vec{u}	6
2.2	An example 2D non-uniform array	8
2.3	Segmentation and windowing of the received time series	11
2.4	Conventional beamforming for an irregular array with 25 sensors	16
3.1	Capon beamforming for an irregular array with 25 sensors	19
3.2	Beampattern for a 10by10 square array using the RCB and SCB method in Case 1	27
3.3	Power estimation for a 10by10 square array using the RCB and SCB mehtod in Case 1	28
3.4	Beampattern and Power estimation using RCB and SCB in Case 2	31
3.5	Power estimation for a 10by10 square array using RCB and SCB in Case 3	32
4.1	The general steps for filtering of seismic signals	34
4.2	The basic scheme of reconstruction	36
4.3	An example 2D velocity filter	40
4.4	The estimated wavenumber-frequency spectrum using FRSI	44
4.5	Reconstruction of an irregular array to a uniform array	46
4.6	Reconstruction and filtering of seismic signal	47
4.7	Reconstruction algorithm (FRSI) applied to real data	48
4.8	Conventional beamforming to a 20by20 square array	49
4.9	Robust Capon Beamforming to a 20by20 square array	49

Introduction

1.1 Seismic array processing

In oil and gas exploration, geophysicists use seismic arrays to image the subsurface. Seismic waves, travelling at different velocities inside or at the surface of the Earth, give information about the Earth's structure. For instance, the velocity of a seismic wave changes when it travels through different Earth layers, thus can help geophysicists to image the Earth and find out whether oil can have accumulated in a certain area or not.

Seismic waves are recorded at the Earth's surface by arrays of sensors. In explorations, seismic arrays with several kilometers range are deployed in potential oil fields. The source signal is generated at the Earth's surface, and the reflected data from different Earth layers are recorded when a source goes off. The sampled data at each array sensor is stored in computers for further processing. The temporal sampling interval is within several mili-seconds.

Besides exploration, seismic arrays are also used in the research field of global seismology e.g. for monitoring earthquake activities. In this case, sampling intervals are much larger and arrays can keep recording data for a long period. Although different types of arrays are deployed in different applications, the basic methodology for array processing is the same. An example array in Tibet for earthquake applications is shown in Figure 1.1(a).

In this thesis, we will focus on passive seismic applications where only ambient seismic noise is recorded by the array. For this application, a small size array (e.g., with 25 sensors) is deployed in a certain field for several days.

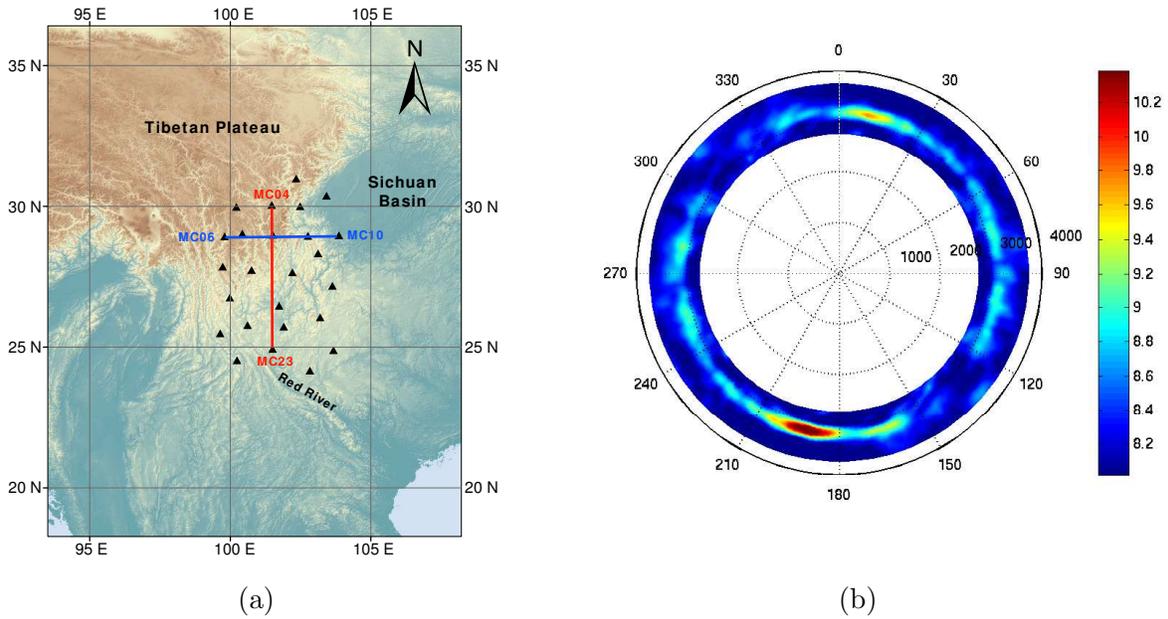


Figure 1.1: An example seismic array and its beamforming output: (a) Tibet seismic array for monitoring earthquakes (b) Beamforming result. (From Yao et al. (2009))

1.1.1 Beamforming

Beamforming is an array processing technique for estimating characteristics of seismic waves. In essence one estimates the wavenumber-frequency spectrum. The basic principle of beamforming is to combine the data collected from all sensors, and then apply multichannel processing algorithms to estimate the parameters of interest. In seismic applications, slowness and angle are the parameters of interest. An example beamforming result for the Tibet array is plotted in Figure 1.1(b). The radius represents the apparent velocity, while the angle is plotted along the circumference of the plot. In this example, ambient noise recorded during one month was beamformed. As can be seen from the figure, a strong source is present with an arrival angle within 180 and 210 degree, and velocity around 3200m/s , pointing to a source in the south-India Ocean [Yao et al. (2009)]. Note in our latter beamforming simulations, we use radius to represent slowness (inverse of speed) instead of velocity.

1.1.2 Filtering

Body waves and surface waves are different seismic wave types. Body waves travel from inside the Earth to the Earth's surface with relatively high speed, while surface waves travel along the Earth's surface with lower speed. In order to improve the beamforming performance, filtering is required to separate body waves and surface waves. Based on the fact that body waves are faster than surface waves, a velocity filter (or f-k filter) can be designed to filter out one type of waves. Most of the passive seismic arrays are irregular arrays, in other words, the propagating wave is non-uniformly sampled in the spatial domain. The design of digital velocity filters for non-uniformly sampled signals is very complex. For this reason, we propose to first reconstruct the non-uniformly sampled spatial signal in a uniform array, and then design a velocity filter to remove the unwanted signals.

1.2 Objectives

1.2.1 Thesis goals

The objectives of this thesis are to tackle two problems.

- Estimate the velocities and arrival angles for seismic signals.
- Filter out one type of seismic waves (body wave or surface wave).

1.2.2 Thesis organization

Chapter 2 introduces the data model. Based on Fourier theory, we show that a time delay causes a phase shift in the frequency domain. We propose a data model by combining all the source signals and noises, and the data model is represented in matrix form. Conventional beamforming is also introduced in this chapter.

Chapter 3 discusses two advanced beamforming algorithms: Capon and Robust Capon beamforming. Capon beamforming is a method which has better resolution and

interference rejection capabilities, while the robust Capon method further improves the performance of Capon's method and extends its applicability.

Chapter 4 discusses reconstruction and filtering. First a new data model for reconstruction purposes will be introduced. Then several reconstruction methods are discussed. After the non-uniformly sampled spatial wave field has been reconstructed in a uniform array, a 2D velocity filter will be designed and implemented. Finally beamforming algorithms will be applied to the filtered data.

Chapter 5 states the conclusion and future work. The advantages and limitations of the introduced algorithms will be discussed, and some recommendations for future work will be presented in this chapter.

2.1 Introduction

Seismic waves consist of body waves and surface waves. A main difference between them is the travelling velocity: body waves are much faster than surface waves. Based on this, a velocity filter can be applied to separate them which is explained in detail in Chapter 4. Another difference is that body waves travel through the inner Earth to the surface while surface waves only travel at the surface. In order to derive the data model, some of the parameters for propagating waves in space will be introduced in this section [Johnson and Dudgeon (1993)].

In this thesis, a 2D array is used for seismic applications. Since a 2D array is deployed on the Earth's surface, body waves which have a vertical travelling angle, need to be decomposed to the array plane (i.e. the Earth's surface) as shown in Figure 2.1. The $x - y$ plane is the plane where the seismic array is placed. Suppose the signal is travelling in space with velocity \vec{v} , and \vec{u} is defined as the slowness of the signal, they have the same angle. The magnitude of slowness equals the inverse of the magnitude of velocity:

$$\begin{aligned} u &= |\vec{u}| = \frac{1}{|\vec{v}|}, \\ v &= |\vec{v}|. \end{aligned} \tag{2.1}$$

As can be seen from Figure 2.1, the body wave is travelling with vertical incident angle (inclination) ϕ and horizontal angle (azimuth) θ . The slowness \vec{u} has a horizontal component in the $x - y$ plane which is defined as the apparent velocity \vec{u}_a . \vec{u}_a can be decomposed to \vec{u}_x and \vec{u}_y . In practice, the vertical incident angle ϕ remains unknown

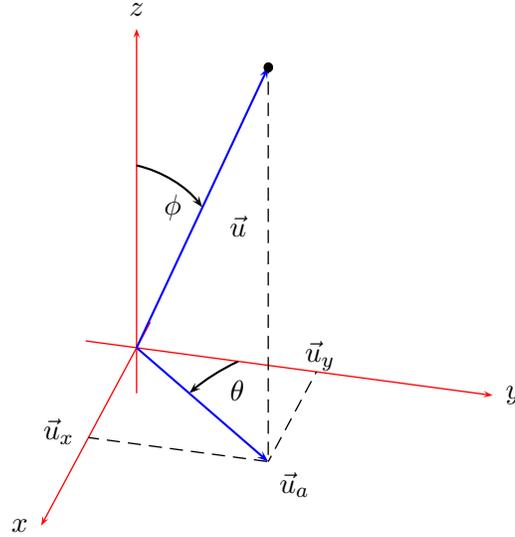


Figure 2.1: Decomposition of the body wave slowness vector \vec{u}

and only the apparent slowness will be estimated [Rost and Thomas (2002)].

$$\begin{aligned}
 u_a &= |\vec{u}_a| = |\vec{u}| \sin \phi = \frac{1}{v} \sin \phi = \frac{1}{(v/\sin \phi)}, \\
 u_x &= |\vec{u}_x| = u_a \sin \theta, \\
 u_y &= |\vec{u}_y| = u_a \cos \theta.
 \end{aligned} \tag{2.2}$$

In the first equation, it can be seen that the apparent velocity value $v_a = \frac{v}{\sin \phi}$ is higher than the true velocity value v for body waves. Since surface waves are travelling along the x-y plane i.e. $\phi = 90^\circ$, its true velocity is the same as the apparent velocity.

Another important parameter for the propagating wave is its wavenumber. Since only the apparent velocities and angles are of interest, from now on, we use \vec{k} to represent the wavenumber in the $x - y$ plane (apparent wavenumber). Moreover, \vec{k} can be decomposed to \vec{k}_x and \vec{k}_y . Assuming ω is the angular frequency of the propagating wave, we have:

$$\begin{aligned}
 k_x &= |\vec{k}_x| = \frac{\omega}{v_a} \sin \theta = \omega u_x, \\
 k_y &= |\vec{k}_y| = \frac{\omega}{v_a} \cos \theta = \omega u_y, \\
 k &= |\vec{k}| = \sqrt{k_x^2 + k_y^2}.
 \end{aligned} \tag{2.3}$$

Assuming τ is the time delay of the seismic signal from the reference point to the sensor location $\vec{x} = (x, y)$, then,

$$\begin{aligned}\tau &= \vec{u} \cdot \vec{x} \\ &= u_x x + u_y y.\end{aligned}\tag{2.4}$$

Thus propagating waves in space can be represented as :

$$\begin{aligned}s(\vec{x}, t) &= s(t - \vec{u} \cdot \vec{x}) \\ &= s(t - \tau).\end{aligned}\tag{2.5}$$

2.2 Frequency Domain Data Model

In seismic applications, the observed signal is normally broadband with relatively low speed, and the distances between the seismic array sensors are very big, which means the delay can not be approximated as a phase shift in the time domain as is the case in radio communications for example [Veen and Leus (2005)]. Therefore in this section, a frequency domain data model will be constructed for array processing applications [Stoica and Moses (1997)].

2.2.1 Fourier Transform theory

Using Fourier Transform theory, and assuming τ is the time delay of the seismic signal between two sensors in different locations, any signal with an arbitrary wave shape can be represented as:

$$\begin{aligned}s(\vec{x}, t) &= s(t - \vec{u} \cdot \vec{x}) \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) \exp \{j\omega(t - \vec{u} \cdot \vec{x})\} d\omega \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) \exp \{j\omega(t - \tau)\} d\omega \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) e^{j\omega t} (e^{-j\omega\tau}) d\omega\end{aligned}\tag{2.6}$$

where $S(\omega)$ is given by the Fourier Transform:

$$S(\omega) = \int_{-\infty}^{\infty} s(t) e^{-j\omega t} dt.\tag{2.7}$$

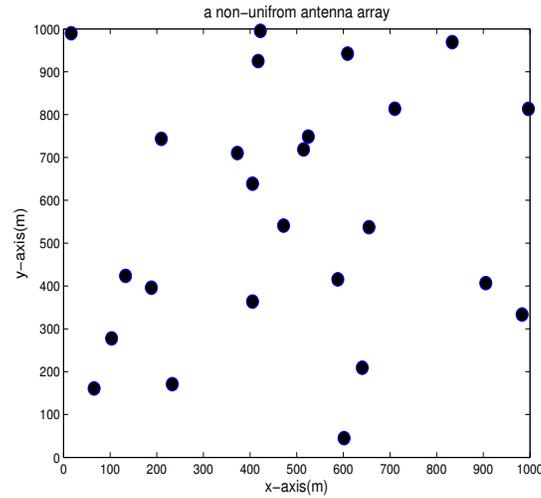


Figure 2.2: An example 2D non-uniform array

Thus we can see that at each angular frequency $\omega = 2\pi f$, a time delay causes a simple phase shift $e^{-j\omega\tau}$. This forms the basis of constructing the data model in frequency domain. Signals received by different sensors have different time delays, which corresponds to phase shifts in the frequency domain, and these phase shifts can be put into the array response vector.

For discrete sampled signals, we can first apply a DFT to short time series to obtain the frequency components, then the frequency of interest can be picked out to form the data model.

2.2.2 Formation of the Data Model

In order to construct a data model for a certain seismic array (an example array with 25 sensors is shown in Figure 2.2), we assume independent sources. The noise received by each array sensor is random, and the signals are uncorrelated with noise.

In this thesis, the data model consists of frequency samples at different times. In general, although the seismic signal received by array sensors are broadband, only the signal at a single frequency f or a small frequency bin centered at f is used. Multiple time samples at frequency f will be obtained in order to construct the data model.

When receiving a long time series at a specific sensor, we will first use short time

windows to chop it into frames. Then to each time frame, the Short Time Fourier Transform (STFT) is applied to convert the time segments into frequency frames, and the frequency component at f from all of the frequency frames will be picked out to form the data vector for that specific sensor. Each component of the data vector is called a snapshot. The details about obtaining time samples for the data model are discussed in the **Practical Issues** section.

Data Model for a single source

Let $\tau_m = u_x x^{(m)} + u_y y^{(m)}$ be the time delay of the source signal at sensor m , where u_x and u_y are the horizontal slownesses of the signal, and $x^{(m)}$ and $y^{(m)}$ determines the sensor location. Suppose I is the number of snapshots for the data model. As explained in the previous section, the phase shift terms can be put into the array response vector $\mathbf{a}(f)$. $\mathbf{X}(f)$ and $\mathbf{N}(f)$ are the $M \times I$ received data matrix and noise matrix respectively, $\mathbf{s}(f)$ is a row vector consisting of I snapshots. Thus the single source data model for an arbitray 2D array with M sensors can be represented as:

$$\mathbf{X}(f) = \mathbf{a}(f)\mathbf{s}(f) + \mathbf{N}(f), \quad \text{where } \mathbf{a}(f) = \begin{bmatrix} \exp(-j2\pi f\tau_0) \\ \exp(-j2\pi f\tau_1) \\ \vdots \\ \exp(-j2\pi f\tau_{M-1}) \end{bmatrix}, \quad (2.8)$$

which equals:

$$\begin{bmatrix} \mathbf{x}_0(f) \\ \mathbf{x}_1(f) \\ \vdots \\ \mathbf{x}_{M-1}(f) \end{bmatrix} = \begin{bmatrix} \exp(-j2\pi f\tau_0) \\ \exp(-j2\pi f\tau_1) \\ \vdots \\ \exp(-j2\pi f\tau_{M-1}) \end{bmatrix} \mathbf{s}(f) + \begin{bmatrix} \mathbf{n}_0(f) \\ \mathbf{n}_1(f) \\ \vdots \\ \mathbf{n}_{M-1}(f) \end{bmatrix}, \quad (2.9)$$

where $\mathbf{x}_m(f)$ is a $1 \times I$ frequency domain data vector at the m -th sensor, and $\mathbf{n}_m(f)$ is a $1 \times I$ noise vector at sensor m .

Data Model for multiple sources

For the more general case, when multiple sources are impinging on the array, the

array vectors for each source can be put into one large matrix. Assuming d sources are present, the data model is written as:

$$\begin{aligned}
\mathbf{X}(f) &= \mathbf{a}_1(f)\mathbf{s}_1(f) + \mathbf{a}_2(f)\mathbf{s}_2(f) + \cdots + \mathbf{a}_d(f)\mathbf{s}_d(f) + \mathbf{N}(f) \\
&= \begin{bmatrix} \mathbf{a}_1(f) & \mathbf{a}_2(f) & \cdots & \mathbf{a}_d(f) \end{bmatrix} \begin{bmatrix} \mathbf{s}_1(f) \\ \mathbf{s}_2(f) \\ \vdots \\ \mathbf{s}_d(f) \end{bmatrix} + \begin{bmatrix} \mathbf{n}_0(f) \\ \mathbf{n}_1(f) \\ \vdots \\ \mathbf{n}_{M-1}(f) \end{bmatrix} \\
&= \mathbf{A}(f)\mathbf{S}(f) + \mathbf{N}(f)
\end{aligned} \tag{2.10}$$

where

$$\mathbf{A}(f) = \begin{bmatrix} \mathbf{a}_1(f) & \mathbf{a}_2(f) & \cdots & \mathbf{a}_d(f) \end{bmatrix}, \quad \mathbf{S}(f) = \begin{bmatrix} \mathbf{s}_1(f) \\ \mathbf{s}_2(f) \\ \vdots \\ \mathbf{s}_d(f) \end{bmatrix}.$$

For the sake of convenience, we write the data model for a specific form as $\mathbf{X} = \mathbf{AS} + \mathbf{N}$ in the future.

2.2.3 Practical Issues

Dealing with real signals

In seismic experiments, data obtained from the sensors are all real numbers. By applying the Fourier Transform to the real data, the spectrum is mirrored at half of the sampling frequency. In other words, the spectrum contains both positive and negative frequency components, which means beamforming should be applied to both frequencies. For instance, we have a single sine wave with a frequency of 10Hz: $s(t) = \sin(\omega t) = \frac{1}{2}e^{j(\omega t - \frac{\pi}{2})} + \frac{1}{2}e^{-j(\omega t - \frac{\pi}{2})}$, and signals received by different array elements are just delayed versions of the reference signal, that is $s(t - \tau) = \frac{1}{2}e^{j(\omega t - \frac{\pi}{2})}e^{-j\omega\tau} + \frac{1}{2}e^{-j(\omega t - \frac{\pi}{2})}e^{j\omega\tau}$. Therefore a delay will cause a phase shift at both +10Hz and -10Hz in the frequency domain, and beamforming needs to be implemented in both frequencies which increases the complexity.

In order to solve the above mentioned problem, we introduce the Hilbert Transform

to transfer real data to complex counterparts. Suppose $z(t)$ is the signal received by the reference sensor, and its Hilbert Transform is $\hat{z}(t)$, then the analytical signal can be formed as $z_a(t) = z(t) + j\hat{z}(t)$. Suppose the Fourier Transform of the real signal $z(t)$ is $Z(f)$, then the function:

$$Z_a(f) = \begin{cases} 2Z(f) & \text{for } f > 0 \\ Z(f) & \text{for } f = 0 \\ 0 & \text{for } f < 0 \end{cases}$$

is the Fourier transform of the analytical signal. The real signal $z(t)$ can be calculated by applying the inverse Fourier transform $Z_a(f)$ and taking the real part. Thus, we can first apply the Hilbert transform to the real received signal before the Fourier transform to obtain the analytical signal. Then we apply beamforming in the frequency domain.

Obtaining Time Samples for the Data Model

From Equations 2.9 and 2.10, the array data $\mathbf{X}(f)$ is a $M \times I$ matrix where I is the number of time samples (snapshots) at frequency f . In order to get sufficient time samples, we chop the time series received at each sensor into I frames, and windows are applied to each frame, which is shown in Figure 2.3. Then, the Short Time Fourier Transform is applied to each time frame to obtain the frequency components, and the frequency samples of interest (at f) will be picked out to form the I temporal samples for each sensor. Alternatively a narrow frequency bin could be selected and the averaged component at the center frequency f will go into the data vector $\mathbf{x}_m(f)$ at the m -th sensor. In this thesis, a hamming window is selected without overlapping,

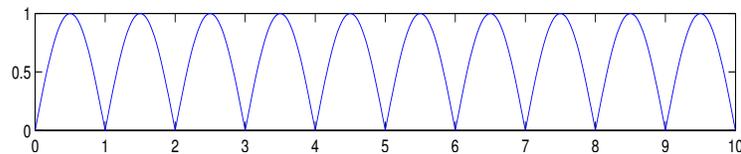


Figure 2.3: Segmentation and windowing of the received time series

and the chosen frequency is 10Hz.

Short-Time Fourier Transform

Suppose $x(t)$ is the time domain signal, then the Short-Time Fourier Transform is defined as:

$$X(t, \omega) = \int_t^{t+T} g(\tau - t)x(\tau)e^{-j\omega\tau} d\tau,$$

where $g(t)$ is a short time window function.

2.3 Conventional Beamforming

2.3.1 Beamforming theory

After the data model has been constructed, conventional beamforming can be applied to estimate the wave-number spectrum for the seismic signal. Recall the frequency domain data model:

$$\begin{aligned} \mathbf{X} &= \mathbf{A}\mathbf{S} + \mathbf{N} \\ &= \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_d \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \vdots \\ \mathbf{s}_d \end{bmatrix} + \begin{bmatrix} \mathbf{n}_0 \\ \mathbf{n}_1 \\ \vdots \\ \mathbf{n}_{M-1} \end{bmatrix} \\ &= \mathbf{a}_1\mathbf{s}_1 + \mathbf{a}_2\mathbf{s}_2 + \dots + \mathbf{a}_d\mathbf{s}_d + \mathbf{N}. \end{aligned} \quad (2.11)$$

Note that \mathbf{a}_l only consists of frequency phase shift terms, where $l \in [1, d]$ indicates the l -th source. In order to recover the source signal \mathbf{s}_l , conventional beamforming multiplies a weight vector $\mathbf{w}^* = \frac{1}{M}\mathbf{a}_l^*$ to the data matrix \mathbf{X} , where M is the number of sensors. Thus,

$$\begin{aligned} \mathbf{w}^*\mathbf{X} &= \frac{1}{M}(\mathbf{a}_1^*\mathbf{a}_1\mathbf{s}_1 + \dots + \mathbf{a}_l^*\mathbf{a}_l\mathbf{s}_l + \dots + \mathbf{a}_d^*\mathbf{a}_d\mathbf{s}_d + \mathbf{a}_l^*\mathbf{N}) \\ &= \mathbf{s}_l + \frac{1}{M}\mathbf{Q}, \end{aligned} \quad (2.12)$$

where $*$ represents complex conjugate transpose, and \mathbf{Q} is the interference plus noise term,

$$\mathbf{Q} = \mathbf{a}_l^*\mathbf{a}_1\mathbf{s}_1 + \dots + \mathbf{a}_l^*\mathbf{a}_{l-1}\mathbf{s}_{l-1} + \dots + \mathbf{a}_l^*\mathbf{a}_{l+1}\mathbf{s}_{l+1} + \dots + \mathbf{a}_l^*\mathbf{a}_d\mathbf{s}_d + \mathbf{a}_l^*\mathbf{N}. \quad (2.13)$$

In order to estimate the velocity and incident angle for the seismic signals, i.e., to estimate the f-k spectrum (the same as the wavenumber-frequency spectrum), the array response needs to be calculated and plotted. Since the array vector \mathbf{a}_l is determined both by the incident angle and velocity of the signal, a 2D polar graph will be used to plot the array response. The plotted values of the polar graph are the averaged power output from the array, which is computed by the following equation:

$$\text{power output} = \frac{1}{I}(\mathbf{w}^* \mathbf{X})(\mathbf{w}^* \mathbf{X})^* = \mathbf{w}^* \hat{\mathbf{R}}_{xx} \mathbf{w}, \quad (2.14)$$

I is the number of snapshots in each sensor, and $\hat{\mathbf{R}}_{xx} = \frac{1}{I} \mathbf{X} \mathbf{X}^*$ is the estimated covariance matrix of the received signal.

2.3.2 Implementation

The basic procedure for implementing conventional beamforming is listed below:

1. Determine the frequency bin of interest and the length of the time frame. Chop the time series into frames at each sensor, and apply a window to each frame.
2. Apply DFT to the windowed frames to obtain the frequency components of interest, construct the data matrix \mathbf{X} .
3. Apply the beamforming algorithm to calculate the average output power from Equation 2.14.
4. Plot the array response with respect to output power in a polar graph.

In oil and gas exploration experiments, the received array signal needs to be filtered for further processing, as explained in Chapter 4. Reconstructing the irregular array signal into a regular array is the first step of the filtering procedure. When the array is uniform, conventional beamforming can be implemented by the IFFT. Assuming B and D are the length (number of sensors) of the uniform array along the x -axis and y -axis, and the sensors are equi-spaced with interval Δx and Δy respectively. Suppose $p(b, d)$ is the spatial sample at the sensor location $(b\Delta x, d\Delta y)$, while \mathbf{p} is a $D \times B$ matrix. $\tilde{p}(n_x, n_y)$ is the (n_x, n_y) -th spectral sample at the f-k domain. Since the spatial Fourier

Transform is defined by:

$$\begin{aligned}\tilde{p}(k_x, k_y) &= \sum_{b=0}^{B-1} \sum_{d=0}^{D-1} p(b, d) e^{j(k_x x_b + k_y y_d)} \\ &= \sum_{b=0}^{B-1} \sum_{d=0}^{D-1} p(b, d) e^{j(k_x b \Delta x + k_y d \Delta y)}.\end{aligned}$$

By sampling the wave-number spectrum, $k_x = \frac{2\pi n_x}{B\Delta x}$, $k_y = \frac{2\pi n_y}{D\Delta y}$, we have:

$$\begin{aligned}\tilde{p}(n_x, n_y) &= \sum_{b=0}^{B-1} \sum_{d=0}^{D-1} p(b, d) e^{j(\frac{2\pi n_x}{B\Delta x} b \Delta x + \frac{2\pi n_y}{D\Delta y} d \Delta y)} \\ &= \sum_{b=0}^{B-1} \sum_{d=0}^{D-1} p(b, d) e^{j(\frac{2\pi n_x b}{B} + \frac{2\pi n_y d}{D})}.\end{aligned}\tag{2.15}$$

The spectral sample intervals are $\Delta k_x = \frac{2\pi}{B\Delta x}$ along the x -axis and $\Delta k_y = \frac{2\pi}{D\Delta y}$ along the y -axis. Thus 2D IFFT can be adopted to calculate the f-k spectrum $\tilde{\mathbf{p}}$. In order to suppress the sidelobes in the f-k domain, a 2D window can be pre-multiplied with the spatial signal \mathbf{p} before IFFT. Zero padding may be used to increase the resolution of the image. After the spectral samples $\tilde{p}(n_x, n_y)$ are computed, the power spectrum for the received signals can be obtained.

2.3.3 Simulations

When the number of sensors in seismic experiments increases, or the distance between the sensors becomes larger, the resolution of conventional beamforming will be increased. However, increasing the number of sensors will increase the cost at the same time. Also, the distance between the sensors cannot be too large. To avoid spatial aliasing, the shortest distance between the sensors should be smaller than half of the signal wavelength. To meet both objectives, sometimes, irregular arrays are designed. Note from now on, the angles in the thesis are representing the incident azimuth angle of the signal, and velocity (or slowness) represents the apparent velocity (or slowness) for the sake of convenience.

We assume two random signals are received by an irregular array with 25 sensors, and the noise received by each sensor is random. The incident angle and velocity for the first random source are 30 and 2.5km/s, while the second source has an incident

angle of 120 degree and $5km/s$ velocity. In order to plot the array response, the conventional beamforming method is used. All the pixels in Figure 2.4 represent the array response vectors for the possible sources. For each pixel location in the polar graph, the pertaining angle and slowness value determine the corresponding array response vector, then conventional beamforming is applied to compute the output power for the possible source. Afterwards the obtained beamformer vector is applied to the received signal matrix to compute the output power (see Equation 2.14). If there is no source signal present at the pixel location, the output power will be very small. When the array response vector for a certain pixel matches one of the source signal's array vectors, we will see large output power. All the possible sources are scanned and the output powers are plotted.

The simulation results are shown in Figure 2.4. Plot (a) and (b) show the first and second source respectively, (c) is the irregular array, and in Figure 2.4(d), a polar graph is used, where the incident angle is plotted along the circumference and the radius represents slownesses. The color corresponds to the output power in dB. From Figure 2.4(d), we can see large output power at the source signal locations. The side-lobes in the polar graph are due to the convolution between the wavenumber-frequency spectrum of the spatial signal and the aperture smoothing function [(Johnson and Dudgeon, 1993, pp. 38-39)]. We conclude that the interference cancellation and noise suppression capabilities of the conventional beamforming method are insufficient.

2.4 Conclusion

In this chapter, a frequency data model was constructed, and the conventional beamforming method was explained. A simulation result was plotted to illustrate its use in estimating signal characteristics. However, the resolution and interference cancellation capabilities of the conventional beamforming method can be improved. Therefore in the next chapter, Capon and Robust Capon Beamforming will be introduced in order to improve the beamforming performance.

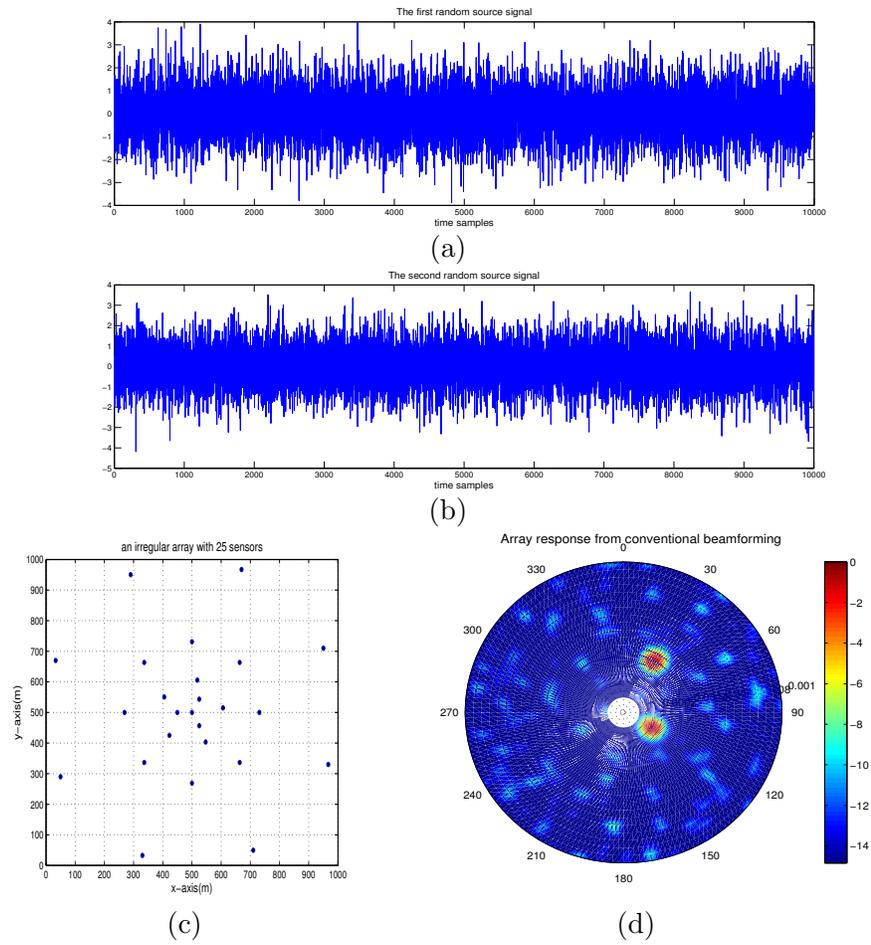


Figure 2.4: Conventional beamforming for an irregular array (a) The 1st source in the time domain (b) The 2nd source in the time domain (c) An irregular array with 25 sensors (d) Conventional beamforming array response

Robust Capon Beamforming

3.1 Introduction

As discussed in Chapter 2, the interference cancellation and noise suppression capabilities of the conventional beamforming method can be improved. Therefore, in order to achieve better estimation performance, Capon and Robust Capon beamforming will be introduced in this section. Some numerical results will also be presented.

Capon beamforming [Capon (1969)] is also called the Linear Constraint with Minimum Variance (LCMV) method. By applying a linear constraint, the signal of interest will not be distorted. At the same time, Capon's method will minimize the output power (or variance), thus the interference and noise terms are suppressed.

Although Capon's method has much better performance than the conventional beamforming method, it has certain limitations. For the cases where the array response vector is inaccurate, or the covariance matrix is singular, the Robust Capon Beamforming method can be adopted to solve the problems [Li et al. (2003)] [Li and Stoica (2005)].

3.2 Standard Capon Beamforming

3.2.1 Problem Statement

Recall the data model in Chapter 2,

$$\begin{aligned}\mathbf{X} &= \mathbf{A}\mathbf{S} + \mathbf{N} \\ &= \mathbf{a}_1\mathbf{s}_1 + \cdots + \mathbf{a}_d\mathbf{s}_d + \mathbf{N}.\end{aligned}$$

There are d sources presented at the array. Suppose the l -th source \mathbf{s}_l is the source of interest, and the Capon beamformer \mathbf{w} is trying to recover it from the signal matrix \mathbf{X} which is obtained from the received array signal. The other $l - 1$ sources in \mathbf{X} are interferences and \mathbf{N} is the noise term. The average output power from the array is $\mathbf{w}^* \mathbf{R}_{xx} \mathbf{w}$, and the Capon beamformer can be interpreted as the solution of the following problem:

$$\min_{\mathbf{w}} \mathbf{w}^* \mathbf{R}_{xx} \mathbf{w}, \quad \text{with constraint } \mathbf{w}^* \mathbf{a}_l = 1. \quad (3.1)$$

where \mathbf{a}_l is the array response vector for the signal of interest (l -th source).

Beamforming can be interpreted as a multiplication of the weight vector \mathbf{w}^* and the data matrix \mathbf{X} , i.e. $\mathbf{w}^* \mathbf{X}$ is the beamforming output. In the SCB (Standard Capon Beamforming) method, suppose the l -th source signal is the signal of interest. By applying a linear constraint to the array response vector \mathbf{a}_l : $\mathbf{w}^* \mathbf{a}_l = 1$, we keep the signal of interest \mathbf{s}_l undistorted, since $\mathbf{w}^* \mathbf{a}_l \mathbf{s}_l = \mathbf{s}_l$. The minimization term in Equation 3.1 is to suppress the interference and noise terms. The limitation of Capon's method is that, when the array response vector \mathbf{a}_l for the signal of interest is not accurately known, we may have applied a linear constraint to an interference, and the true signal of interest is interpreted as an interference which will be suppressed. Thus in this case Capon's method will fail to provide a better estimation result. Also, when the covariance matrix \mathbf{R}_{xx} is poorly conditioned or singular, Capon's method cannot be applied to the data.

3.2.2 Solution

The problem in Equation 3.1 can be solved by the Lagrange multiplier method. The cost function J [Van Trees (2002)] which needs to be minimized is defined as:

$$\begin{aligned} J &= \mathbf{w}^* \mathbf{R}_{xx} \mathbf{w} + \lambda(\mathbf{w}^* \mathbf{a}_l - 1) + \lambda^*(\mathbf{a}_l^* \mathbf{w} - 1) \\ &= \mathbf{w}^* \mathbf{R}_{xx} \mathbf{w} + \lambda \mathbf{w}^* \mathbf{a}_l + \lambda^* \mathbf{a}_l^* \mathbf{w} - \lambda^* - \lambda, \end{aligned}$$

Taking derivative with respect to \mathbf{w} is equivalent to taking the derivative to $\bar{\mathbf{w}}$ (complex conjugate of \mathbf{w}) for stationary points, but the latter is more simple. Therefore we take

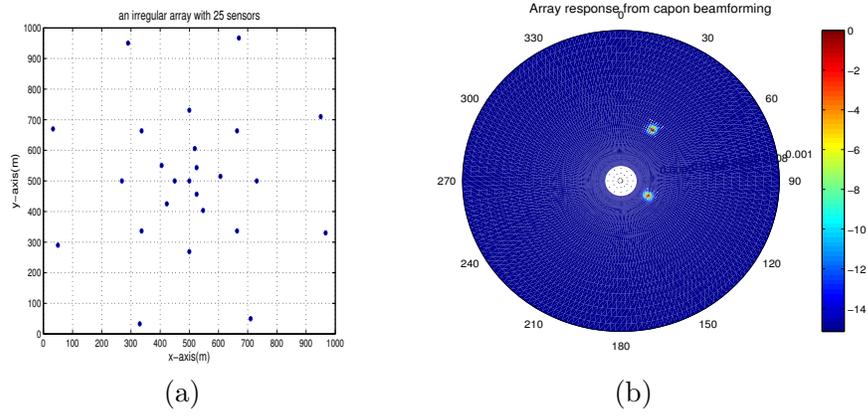


Figure 3.1: Capon beamforming for an irregular array with 25 sensors (a) An irregular array (b) Array response computed by capon beamforming

the derivative of J with respect to $\bar{\mathbf{w}}$ which yields:

$$\begin{aligned}\nabla_{\bar{\mathbf{w}}} J &= \nabla_{\bar{\mathbf{w}}} \mathbf{w}^* \mathbf{R}_{xx} \mathbf{w} + \nabla_{\bar{\mathbf{w}}} \lambda \mathbf{w}^* \mathbf{a}_l + \nabla_{\bar{\mathbf{w}}} \lambda^* \mathbf{a}_l^* \mathbf{w} \\ &= \mathbf{R}_{xx} \mathbf{w} + \lambda \mathbf{a}_l.\end{aligned}$$

Setting $\nabla_{\bar{\mathbf{w}}} J = 0$ yields $\mathbf{w} = -\lambda \mathbf{R}_{xx}^{-1} \mathbf{a}_l$. By using the linear constraint, it can be found that $\lambda = -(\mathbf{a}_l^* \mathbf{R}_{xx}^{-1} \mathbf{a}_l)^{-1}$, Substituting λ , we arrive at the solution:

$$\mathbf{w} = \frac{\mathbf{R}_{xx}^{-1} \mathbf{a}_l}{\mathbf{a}_l^* \mathbf{R}_{xx}^{-1} \mathbf{a}_l}. \quad (3.2)$$

After computing the Capon beamformer \mathbf{w} , the output power from the array can be expressed as:

$$\begin{aligned}\text{output power} &= \mathbf{w}^* \mathbf{R}_{xx} \mathbf{w}, \\ &= \frac{1}{\mathbf{a}_l^* \mathbf{R}_{xx}^{-1} \mathbf{a}_l}.\end{aligned} \quad (3.3)$$

3.2.3 Numerical Results

In this section, the same array as shown in Figure 2.4(c) is used for simulating the Capon beamforming performance. The array response is plotted in Figure 3.1(b). Assuming two random sources are present at the array, coming in azimuths of 30 and 120 degree, slowness $4 \times 10^{-4} s/m$ and $2 \times 10^{-4} s/m$ respectively. The noise received by each sensor

is random as well. As can be seen from the plot, Capon beamforming provides better estimation performance than the conventional beamforming method (in Figure 2.4).

As in Figure 2.4(b), all the possible array response vectors will be scanned, and the output powers in Equation 3.3 are plotted to find the angle and slowness location for the true source signals. From Figure 3.1, we can see that the resolution of Capon's method is much better than the conventional one, and interferences are also well suppressed.

3.3 Robust Capon Beamforming

3.3.1 Problems of Capon Beamforming

Standard Capon Beamforming (SCB) provides a better performance compared to the conventional beamforming algorithm, with better resolution and interference rejection capabilities. However, the Standard Capon Beamforming algorithm requires a rather accurate array response vector: a small variation of the array response vector will degrade the performance of Capon beamforming to become even worse to that of the conventional beamforming method. Also, Capon's method requires the computation of the inverse of the covariance matrix \mathbf{R}_{xx} . If the covariance matrix is poorly conditioned or singular, Capon's method will fail.

In practical cases, the array response vector may have small variations, e.g., the assumed arrival angle of the source signal may be not accurate, or the covariance matrix may be poorly conditioned. Therefore the Robust Capon beamforming algorithm which can tackle these problems is introduced in this section [Li et al. (2003)].

3.3.2 Derivation of the Robust Capon Beamforming(RCB) algorithm

In this section, we summarize the work of Jian Li [Li et al. (2003)].

3.3.2.1 Reformulation of the SCB method

Suppose \mathbf{a}_l is the array response vector corresponding to the signal of interest \mathbf{s}_l , and $\tilde{\sigma}_l^2$ is the estimated Capon beamformer output power:

$$\tilde{\sigma}_l^2 = \frac{1}{\mathbf{a}_l^* \mathbf{R}_{xx}^{-1} \mathbf{a}_l}. \quad (3.4)$$

It will be shown that $\tilde{\sigma}_l^2$ is the solution of the following covariance fitting problem [Li and Stoica (2005)]:

$$\max_{\sigma^2} \sigma^2 \quad \text{subject to } \mathbf{R}_{xx} - \sigma^2 \mathbf{a}_l \mathbf{a}_l^* \geq 0. \quad (3.5)$$

where the notation $\mathbf{G} \geq 0$ means \mathbf{G} is a positive semidefinite matrix. The problem indicated in Equation 3.5 leads to the same solution as Standard Capon Beamforming as will be shown below. \mathbf{R}_{xx} is a Hermitian matrix i.e., $\mathbf{R}_{xx}^* = \mathbf{R}_{xx}$, where $*$ represents complex conjugate transpose. For the sake of convenience, we use \mathbf{R} to represent \mathbf{R}_{xx} from now on, thus we have:

$$\begin{aligned} \mathbf{R} - \sigma^2 \mathbf{a}_l \mathbf{a}_l^* &\geq 0 \\ \Leftrightarrow \mathbf{I} - \sigma^2 \mathbf{R}^{-\frac{1}{2}} \mathbf{a}_l \mathbf{a}_l^* \mathbf{R}^{-\frac{1}{2}} &\geq 0 \\ \Rightarrow (\mathbf{R}^{-1/2} \mathbf{a}_l)^* (\mathbf{R}^{-1/2} \mathbf{a}_l) - \sigma^2 (\mathbf{R}^{-1/2} \mathbf{a}_l)^* \mathbf{R}^{-1/2} \mathbf{a}_l \mathbf{a}_l^* \mathbf{R}^{-1/2} (\mathbf{R}^{-1/2} \mathbf{a}_l) &\geq 0 \\ \Leftrightarrow \mathbf{a}_l^* \mathbf{R}^{-1} \mathbf{a}_l - \sigma^2 (\mathbf{a}_l^* \mathbf{R}^{-1} \mathbf{a}_l)^2 &\geq 0 \\ \Leftrightarrow 1 - \sigma^2 \mathbf{a}_l^* \mathbf{R}^{-1} \mathbf{a}_l &\geq 0 \\ \Leftrightarrow \sigma^2 \leq \frac{1}{\mathbf{a}_l^* \mathbf{R}^{-1} \mathbf{a}_l} = \tilde{\sigma}_l^2 & \end{aligned} \quad (3.6)$$

Hence $\sigma^2 = \tilde{\sigma}_l^2$ is the largest value of σ^2 for which the constraint in Equation 3.5 is satisfied.

3.3.2.2 RCB with Nondegenerate Spherical Uncertainty Set

The array response vector \mathbf{a}_l for the signal of interest is not accurately known, and the assumed array response vector for \mathbf{a}_l is $\bar{\mathbf{a}}$. In order to find the correct array response vector \mathbf{a}_l , we append a spherical uncertainty set to $\bar{\mathbf{a}}$. Suppose \mathbf{a} is an array response vector belonging to the uncertainty set, then we seek the maximum output power σ^2

for different array response vectors \mathbf{a} , the correspondent array vector is the one which approximates the actual array response vector \mathbf{a}_l best. Assuming ϵ is a positive real number which defines the size of the uncertainty set, then we can represent the RCB problem as:¹

$$\begin{aligned} \max_{\sigma^2, \mathbf{a}} \sigma^2 \quad & \text{subject to } \mathbf{R} - \sigma^2 \mathbf{a} \mathbf{a}^* \geq 0, \\ & \text{for any } \mathbf{a} \text{ satisfying } (\mathbf{a} - \bar{\mathbf{a}})^* (\mathbf{a} - \bar{\mathbf{a}}) \leq \epsilon. \end{aligned} \quad (3.7)$$

For any given array response vector \mathbf{a} inside the uncertainty set, the estimated output power $\tilde{\sigma}^2$ is equal to:

$$\tilde{\sigma}^2 = \frac{1}{\mathbf{a}^* \mathbf{R}^{-1} \mathbf{a}}. \quad (3.8)$$

Therefore the RCB problem can be reformulated as:

$$\min_{\mathbf{a}} \mathbf{a}^* \mathbf{R}^{-1} \mathbf{a} \quad \text{subject to } \|\mathbf{a} - \bar{\mathbf{a}}\|^2 \leq \epsilon. \quad (3.9)$$

Let $\bar{\mathcal{S}}$ represent the spherical uncertainty set. In order to exclude the solution $\mathbf{a} = 0$, the assumed array response vector should meet the requirement:

$$\|\bar{\mathbf{a}}\|^2 > \epsilon. \quad (3.10)$$

Since we know the solution of Equation 3.9 lies at the boundary of the uncertainty set, the problem can be reformuated as:

$$\min_{\mathbf{a}} \mathbf{a}^* \mathbf{R}^{-1} \mathbf{a} \quad \text{subject to } \|\mathbf{a} - \bar{\mathbf{a}}\|^2 = \epsilon. \quad (3.11)$$

The above indicated problem can be solved by the *Lagrange multiplier method* [Quarteroni et al. (2007)]. Define the cost function as:

$$f(\mathbf{a}, \lambda) = \mathbf{a}^* \mathbf{R}^{-1} \mathbf{a} + \lambda (\|\mathbf{a} - \bar{\mathbf{a}}\|^2 - \epsilon), \quad (3.12)$$

where λ is the non-negative real-valued lagrange multiplier. To find the optimal estimation of the SOI (Signal Of Interest) array vector $\hat{\mathbf{a}}_l$, we have to differentiate $f(\mathbf{a}, \lambda)$ with respect to \mathbf{a} . Moreover, for any stationary point, differentiation to \mathbf{a} is equivalent

¹More in general, a nondegenerate ellipsoidal uncertainty set can be used, that is: $(\mathbf{a} - \bar{\mathbf{a}})^* \mathbf{C}^{-1} (\mathbf{a} - \bar{\mathbf{a}}) \leq 1$, where \mathbf{C} is the constraint matrix.

to differentiation to the complex conjugate of \mathbf{a} . The gradient of $f(\mathbf{a}, \lambda)$ to the complex conjugate of \mathbf{a} , which gives the optimal estimation $\hat{\mathbf{a}}_l$, is:

$$\mathbf{R}^{-1}\hat{\mathbf{a}}_l + \lambda(\hat{\mathbf{a}}_l - \bar{\mathbf{a}}) = \mathbf{0}. \quad (3.13)$$

Then:

$$\hat{\mathbf{a}}_l = \left(\frac{\mathbf{R}^{-1}}{\lambda} + \mathbf{I} \right)^{-1} \bar{\mathbf{a}} \quad (3.14)$$

$$= \bar{\mathbf{a}} - (\mathbf{I} + \lambda\mathbf{R})^{-1}\bar{\mathbf{a}} \quad (3.15)$$

where the second equality is obtained by the matrix inversion lemma. By substituting the array response vector $\hat{\mathbf{a}}_l$ into the constraint equation 3.11 to find λ , we get:

$$g(\lambda) = \|(\mathbf{I} + \lambda\mathbf{R})^{-1}\bar{\mathbf{a}}\|^2 = \epsilon. \quad (3.16)$$

Applying an eigenvalue decomposition to \mathbf{R} , that is, $\mathbf{R} = \mathbf{U}\mathbf{\Gamma}\mathbf{U}^*$, we get:

$$\begin{aligned} g(\lambda) &= \|(\mathbf{I} + \lambda\mathbf{R})^{-1}\bar{\mathbf{a}}\|^2, \\ &= \|(\mathbf{U}\mathbf{U}^* + \mathbf{U}(\lambda\mathbf{\Gamma})\mathbf{U}^*)^{-1}\bar{\mathbf{a}}\|^2, \\ &= \|(\mathbf{U}(\mathbf{I} + \lambda\mathbf{\Gamma})\mathbf{U}^*)^{-1}\bar{\mathbf{a}}\|^2, \\ &= \|(\mathbf{U}(\mathbf{I} + \lambda\mathbf{\Gamma})^{-1}\mathbf{U}^*)\bar{\mathbf{a}}\|^2, \\ &= [\mathbf{U}(\mathbf{I} + \lambda\mathbf{\Gamma})^{-1}\mathbf{U}^*\bar{\mathbf{a}}]^* [\mathbf{U}(\mathbf{I} + \lambda\mathbf{\Gamma})^{-1}\mathbf{U}^*\bar{\mathbf{a}}], \\ &= (\mathbf{U}^*\bar{\mathbf{a}})^*(\mathbf{I} + \lambda\mathbf{\Gamma})^{-2}(\mathbf{U}^*\bar{\mathbf{a}}) \quad \text{now let } \mathbf{z} = \mathbf{U}^*\bar{\mathbf{a}}, \quad \text{then,} \\ &= \mathbf{z}^*(\mathbf{I} + \lambda\mathbf{\Gamma})^{-2}\mathbf{z}, \\ &= \sum_{m=1}^M \frac{|z_m|^2}{(1 + \lambda\gamma_m)^2} = \epsilon, \end{aligned} \quad (3.17)$$

where z_m is the element in vector \mathbf{z} , and $\gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_M$ are the eigenvalues of the covariance matrix \mathbf{R} .

As can be seen from Equation 3.17, $g(\lambda)$ is a monotonically decreasing function when $\lambda \geq 0$. From Equations 3.10 and 3.16, $g(0) > \epsilon$, therefore $\lambda \neq 0$. From Equation 3.17, $\lim_{\lambda \rightarrow \infty} g(\lambda) = 0 < \epsilon$. Hence we can find a unique λ from Equation 3.17. The interval for λ is given by:

$$\frac{\|\bar{\mathbf{a}}\| - \sqrt{\epsilon}}{\gamma_1\sqrt{\epsilon}} \leq \lambda \leq \min \left\{ \left(\frac{1}{\epsilon} \sum_{m=1}^M \frac{|z_m|^2}{\gamma_m^2} \right)^{1/2}, \frac{\|\bar{\mathbf{a}}\| - \sqrt{\epsilon}}{\gamma_M\sqrt{\epsilon}} \right\}. \quad (3.18)$$

Note that the upper boundary of λ is not available when \mathbf{R} is rank deficient, since then $\gamma_M = 0$. Newton's method can be used to find the Lagrange multiplier.

Equation 3.17 can be solved either by the Newton's method, or by setting the interval of λ and trying to find the point for which $h(\lambda) = g(\lambda) - \epsilon$ crosses zero. When the Lagrange multiplier λ is found, $\hat{\mathbf{a}}_l$ can be calculated as:

$$\hat{\mathbf{a}}_l = \bar{\mathbf{a}} - \mathbf{U}(\mathbf{I} + \lambda\mathbf{\Gamma})^{-1}\mathbf{U}^*\bar{\mathbf{a}}. \quad (3.19)$$

When the estimated array vector $\hat{\mathbf{a}}_l$ for the signal of interest (SOI) has been calculated, the Robust Capon beamformer $\hat{\mathbf{w}}_l$ has the same form as the SCB vector (Equation 3.2), and can be derived as follows:

$$\begin{aligned} \hat{\mathbf{w}}_l &= \frac{\mathbf{R}^{-1}\hat{\mathbf{a}}_l}{\hat{\mathbf{a}}_l^*\mathbf{R}^{-1}\hat{\mathbf{a}}_l} \\ &= \frac{\mathbf{R}^{-1}\left(\frac{\mathbf{R}^{-1}}{\lambda} + \mathbf{I}\right)^{-1}\bar{\mathbf{a}}}{\left[\left(\frac{\mathbf{R}^{-1}}{\lambda} + \mathbf{I}\right)^{-1}\bar{\mathbf{a}}\right]^*\mathbf{R}^{-1}\left(\frac{\mathbf{R}^{-1}}{\lambda} + \mathbf{I}\right)^{-1}\bar{\mathbf{a}}} \\ &= \frac{(\mathbf{R} + \frac{1}{\lambda}\mathbf{I})^{-1}\bar{\mathbf{a}}}{\bar{\mathbf{a}}^*\left(\frac{\mathbf{R}^{-1}}{\lambda} + \mathbf{I}\right)^{-1}\mathbf{R}^{-1}\left(\frac{\mathbf{R}^{-1}}{\lambda} + \mathbf{I}\right)^{-1}\bar{\mathbf{a}}} \\ &= \frac{(\mathbf{R} + \frac{1}{\lambda}\mathbf{I})^{-1}\bar{\mathbf{a}}}{\bar{\mathbf{a}}^*\left(\frac{\mathbf{R}^{-1}}{\lambda} + \mathbf{I}\right)^{-1}\mathbf{R}^{-1}\mathbf{R}\left(\mathbf{R} + \frac{1}{\lambda}\mathbf{I}\right)^{-1}\bar{\mathbf{a}}} \\ &= \frac{(\mathbf{R} + \frac{1}{\lambda}\mathbf{I})^{-1}\bar{\mathbf{a}}}{\bar{\mathbf{a}}^*\left(\mathbf{R} + \frac{1}{\lambda}\mathbf{I}\right)^{-1}\mathbf{R}\left(\mathbf{R} + \frac{1}{\lambda}\mathbf{I}\right)^{-1}\bar{\mathbf{a}}}. \end{aligned} \quad (3.20)$$

In practice, the covariance matrix \mathbf{R} is unknown and needs to be estimated as well. Suppose we have I snapshots for the data matrix \mathbf{X} , the estimated covariance matrix $\hat{\mathbf{R}} = \frac{1}{I}\mathbf{X}\mathbf{X}^*$ is substituted to calculate $\hat{\mathbf{w}}_l$.

At the same time, the power for the signal of interest can also be calculated. As-

suming $\hat{\sigma}_l^2$ is the estimated SOI power, we have:

$$\begin{aligned}
\hat{\sigma}_l^2 &= \hat{\mathbf{w}}_l^* \mathbf{R} \hat{\mathbf{w}}_l \\
&= \frac{1}{\hat{\mathbf{a}}_l^* \mathbf{R}^{-1} \hat{\mathbf{a}}_l} \\
&= \frac{1}{\left[\left(\frac{\mathbf{R}^{-1}}{\lambda} + \mathbf{I} \right)^{-1} \bar{\mathbf{a}} \right]^* \mathbf{R}^{-1} \left[\left(\frac{\mathbf{R}^{-1}}{\lambda} + \mathbf{I} \right)^{-1} \bar{\mathbf{a}} \right]} \\
&= \frac{1}{\bar{\mathbf{a}}^* \left(\frac{\mathbf{R}^{-1}}{\lambda} + \mathbf{I} \right)^{-1} \mathbf{R}^{-1} \left(\frac{\mathbf{R}^{-1}}{\lambda} + \mathbf{I} \right)^{-1} \bar{\mathbf{a}}} \\
&= \frac{1}{\bar{\mathbf{a}}^* \left(\mathbf{R} + \frac{1}{\lambda} \mathbf{I} \right)^{-1} \mathbf{R} \left(\mathbf{R} + \frac{1}{\lambda} \mathbf{I} \right)^{-1} \bar{\mathbf{a}}} \\
&= \frac{1}{\bar{\mathbf{a}}^* \left(\mathbf{U} \mathbf{\Gamma} \mathbf{U}^* + \frac{1}{\lambda} \mathbf{I} \right)^{-1} \mathbf{U} \mathbf{\Gamma} \mathbf{U}^* \left(\mathbf{U} \mathbf{\Gamma} \mathbf{U}^* + \frac{1}{\lambda} \mathbf{I} \right)^{-1} \bar{\mathbf{a}}} \\
&= \frac{1}{\bar{\mathbf{a}}^* \mathbf{U} \left(\mathbf{\Gamma} + \frac{1}{\lambda} \mathbf{I} \right)^{-1} \mathbf{U}^* \mathbf{U} \mathbf{\Gamma} \mathbf{U}^* \mathbf{U} \left(\mathbf{\Gamma} + \frac{1}{\lambda} \mathbf{I} \right)^{-1} \mathbf{U}^* \bar{\mathbf{a}}} \\
&= \frac{1}{\bar{\mathbf{a}}^* \mathbf{U} \mathbf{\Gamma} \left(\mathbf{\Gamma} + \frac{1}{\lambda} \mathbf{I} \right)^{-1} \left(\mathbf{\Gamma} + \frac{1}{\lambda} \mathbf{I} \right)^{-1} \mathbf{U}^* \bar{\mathbf{a}}} \\
&= \frac{1}{\bar{\mathbf{a}}^* \mathbf{U} \mathbf{\Gamma} \left(\lambda^{-2} \mathbf{I} + 2\lambda^{-1} \mathbf{\Gamma} + \mathbf{\Gamma}^2 \right)^{-1} \mathbf{U}^* \bar{\mathbf{a}}}. \tag{3.21}
\end{aligned}$$

As before, the estimated covariance matrix $\hat{\mathbf{R}} = \hat{\mathbf{U}} \hat{\mathbf{\Gamma}} \hat{\mathbf{U}}^*$ is substituted for \mathbf{R} . For the power estimation, there is a *scaling ambiguity* in the covariance term $\mathbf{R} - \sigma^2 \mathbf{a} \mathbf{a}^*$, in the sense that (σ^2, \mathbf{a}) and $(\sigma^2/\alpha, \alpha^{1/2} \mathbf{a})$ give the same term $\sigma^2 \mathbf{a} \mathbf{a}^*$. To eliminate this ambiguity, we estimate the SOI power $\hat{\sigma}_l^2$ as:

$$\hat{\sigma}_l^2 = \hat{\sigma}_l^2 \|\hat{\mathbf{a}}_l\|^2 / M. \tag{3.22}$$

where M is the number of sensors.

This concludes the summary of Jian Li's work.

Newton's Method

In order to solve Equation 3.17, Newton's method [Quarteroni et al. (2007)] can be used to find the root of the function $h(\lambda) = g(\lambda) - \epsilon = \sum_{m=1}^M \frac{|z_m|^2}{(1+\lambda\gamma_m)^2} - \epsilon$. At the same time, Newton's method is capable of finding the root of a function with only a small amount of iterations when the starting search point is near to the true value of the root.

Based on Newton's method, we begin with the lower bound of λ in Equation 3.18. At each iteration, the value of λ is updated,

$$\lambda_{n+1} = \lambda_n - \frac{h(\lambda_n)}{h'(\lambda_n)} \quad (3.23)$$

where $h'(\lambda)$ is the derivative of function $h(\lambda)$:

$$\begin{aligned} h'(\lambda_n) &= \frac{d}{d\lambda} \left[\sum_{m=1}^M \frac{|z_m|^2}{(1 + \lambda\gamma_m)^2} - \epsilon \right] \Big|_{\lambda=\lambda_n} \\ &= \sum_{m=1}^M |z_m|^2 (-2)(1 + \lambda_n\gamma_m)^{-3} \gamma_m \\ &= \sum_{m=1}^M \frac{-2|z_m|^2 \gamma_m}{(1 + \lambda_n\gamma_m)^3} \end{aligned} \quad (3.24)$$

3.3.3 Simulation Results

In this section, some of the simulation results for Robust Capon beamforming are shown. Both the beampattern and estimated power computed by the RCB and SCB method are plotted. By comparing the beampattern and power estimation performance of the SCB and RCB method, we will see that RCB outperforms SCB.

Beampattern

When the Robust Capon beamformer $\hat{\mathbf{w}}_l$ for the signal of interest (l -th source) has been calculated, the array beampattern can be plotted. Suppose \mathbf{a} is the array vector for a source signal with a certain angle and slowness, the beampattern is calculated by $|\hat{\mathbf{w}}_l^* \mathbf{a}|^2$, while varying \mathbf{a} through all the angles or slownesses.

Power Estimation

Power estimation performance of the RCB method is also simulated. The estimated power is calculated by Equation 3.22, thus the true power of the source signal has been computed.

The simulation section is divided into three subsections, where three different scenarios will be presented. **Case 1** is the scenario when the incoming sources have the same slowness but different angles, **Case 2** is for the sources with same arrival angle

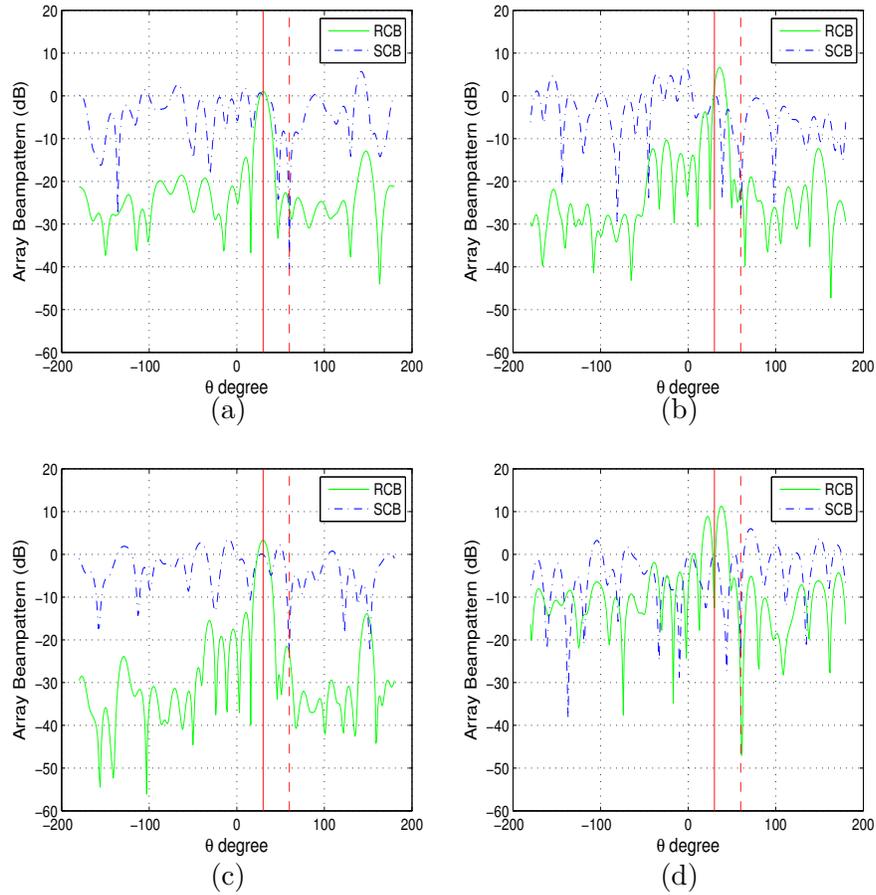


Figure 3.2: Beampattern for a 10by10 square array using the RCB and SCB method in Case 1: (a) Beampattern without mismatch, $\epsilon = 1$ (b) Beampattern with 2 degree of mismatch, $\epsilon = 10$ (c) Beampattern without mismatch, selection error for ϵ , $\epsilon = 10 > \epsilon_0 = 0$ (d) Beampattern with 2 degree of mismatch, selection error for ϵ , $\epsilon = 1 < \epsilon_0 = 10$

but different slownesses, while **Case 3** represents the different angles and different slownesses scenario. In **Case 1** and **Case 2**, both the beampattern and estimated power will be plotted, in **Case 3**, only the power estimation performance is simulated.

3.3.3.1 Case 1: Sources with different arrival angles but same slowness

In this case, we assume two sources with different arrival angles but same slowness imping on a 10by10 square array, with distance 150 meters between neighboring sensors. The arrival angles of the sources are 30 and 60 degree respectively, while both travel

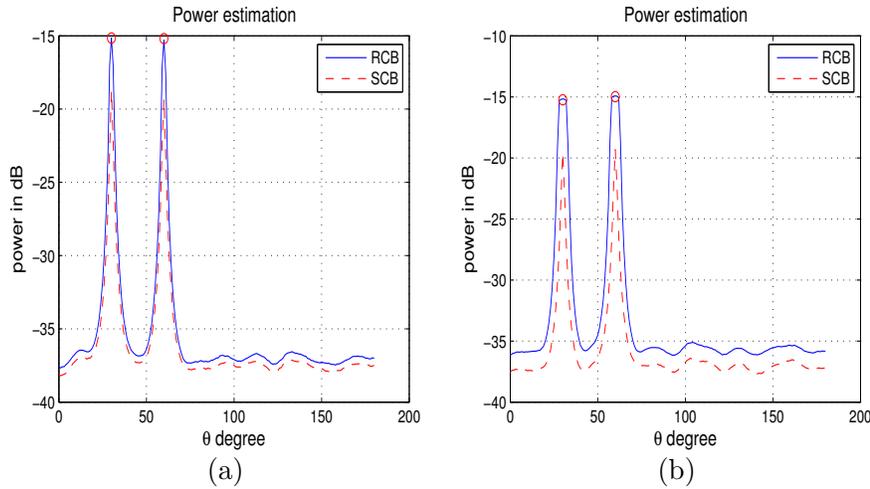


Figure 3.3: Power estimation for a 10by10 square array using the RCB and SCB mehtod in Case 1: (a) Power estimation $\epsilon = 1$ (b) Power estimation $\epsilon = 10$

at the speed of $3400m/s$. We choose the first source with the arrival angle of 30 degree as the signal of interest. The beampattern simulation results are plotted in Figure 3.2.

The left plots in Figure 3.2 (a) and (c) are the beampatterns for no mismatch of the assumed array vector $\bar{\mathbf{a}}$ case, while the right plots in Figure 3.2 (b) and (d) have a 2 degree of angle mismatch for the assumed array vector. The vertical solid red line indicates the location of the signal of interest, while the dotted red line is the interference (arrival angle of 60 degree). We consider beampatterns computed from both the SCB method and RCB method. As can be seen from the plots, Robust Capon Beamforming outperforms Standard Capon Beamforming.

In order to simulate power estimation of the RCB method, we add a small complex random number to each element of the assumed array vector $\bar{\mathbf{a}}$. The real power for the source signals are also plotted (red circles) in our simulation results. The same 10by10 square array is used, and the two sources are also the same as before. As can be seen from Figure 3.3, the RCB method has a better power estimation accuracy than the SCB method.

Selecting values for ϵ

For the no mismatch plots (a) and (c) in Figure 3.2, the true value for ϵ is $\epsilon_0 = 0$.

For (b) and (d) in Figure 3.2, $\epsilon_0 = \|\mathbf{a}_l - \bar{\mathbf{a}}\|^2 = 9.7429 \approx 10$, where a relatively big array is deployed with 100 sensors, thus the array response vector is a 100×1 vector. As can be seen from the plots, in the no mismatch case, the selection value for ϵ has no effects on the beampattern. However, when a mismatch is present in the assumed array vector, then if ϵ is chosen too small, the RCB method cannot find the correct array vector, which means RCB has almost the same performance as SCB.

3.3.3.2 Case 2: Sources with different slownesses but same arrival angle

In this section, the source signals have different slownesses but same arrival angle. Both of them are from the direction of 30 degree, and the travelling velocities are $3400m/s$ and $5000m/s$ respectively, thus slownesses $2.94 \times 10^{-4}s/m$ and $2 \times 10^{-4}s/m$ for the sources. A same 10by10 square array as in **Case 1** is adopted. The simulation results are plotted in Figure 3.4.

The beampattern for the first source (travelling velocity $3400m/s$) is plotted in Figure 3.4(a) and (b). (a) is for the nomismatch case, while (b) is the beampattern with $10m/s$ velocity error (or slowness error) for the assumed array vector.

Figure 3.4 (c)-(f) shows the simulation results for power estimation. (c) is the power estimation result when the assumed array vector $\bar{\mathbf{a}}$ has no mismatch, while (d) is the mismatch case where each element in the assumed array vector is added with a small complex random number. Figure 3.4 (e) and (f) are the simulation results when ϵ is chosen too large or too small respectively.

In power estimation, the selected value for ϵ is important. The true uncertainty set size $\epsilon_0 \approx 1$. As can be seen from plots Figure 3.4 (e) to (f), when ϵ is chosen too large, the resolution of the RCB method is worse than that of the SCB method. If ϵ is selected much smaller than the its true value, then the uncertainty set doesnot include the correct array vector \mathbf{a}_l , which results in almost the same performance between RCB and SCB.

3.3.3.3 Case 3: Sources with different slownesses and arrival angles

In this section, only the estimated power for the sources with different slownesses and angles are plotted. Assuming the same 10by10 square array as before, the simulation results are plotted in Figure 3.5. The first source has an angle of 30 degree, and velocity 3400m/s. The angle and velocity for the second source are 60 degree and 5000m/s. Figure 3.5 (a) and (b) represent the simulation results of the no array response vector (ARV or array vector) mismatch case, for SCB and RCB respectively. Figure 3.5 (c) and (d) are SCB and RCB power estimation with ARV mismatch. Figure 3.5(e) is the case with mismatch when ϵ is too large and (f) is the ϵ too small case. The mismatch term consists of complex random numbers as before.

3.4 Conclusion

In this chapter, the standard Capon beamforming method is analyzed and simulated. In order to improve the performance of the SCB method, robust Capon beamforming is introduced and analyzed. From the simulation results, we conclude that RCB outperforms SCB in beamforming applications.

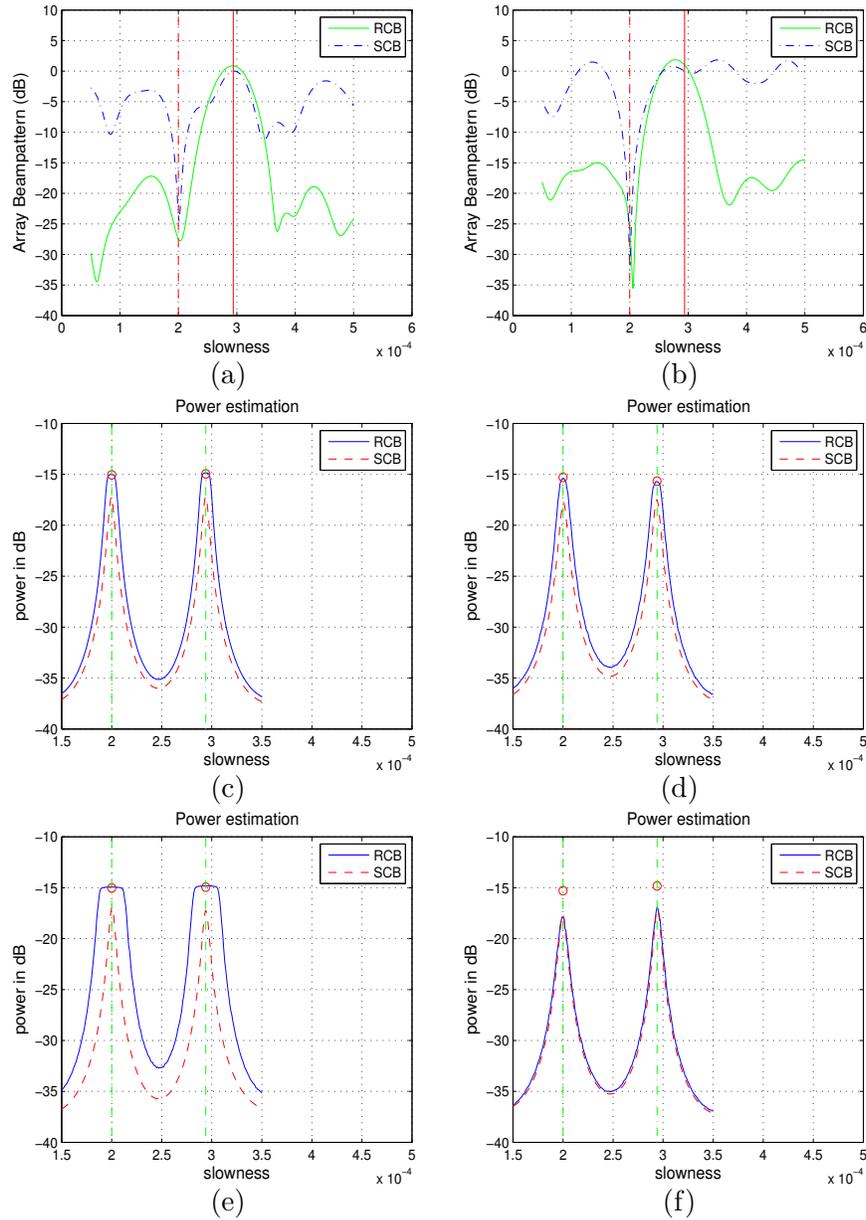


Figure 3.4: Beampattern and Power estimation for a 10by10 array using RCB and SCB in Case 2: (a) Beampattern without mismatch, $\epsilon = 0.5$ (b) Beampattern with 10m/s of velocity mismatch, $\epsilon = 0.5$ (c) Power estimation without mismatch, $\epsilon = 1$ (d) Power estimation with mismatch, good selection of ϵ , $\epsilon = 1$ (e) Power estimation with mismatch, ϵ is chosen bigger than the true value, $\epsilon = 10$ (f) Power estimation with mismatch and ϵ is chosen smaller than the true value, $\epsilon = 0.1$

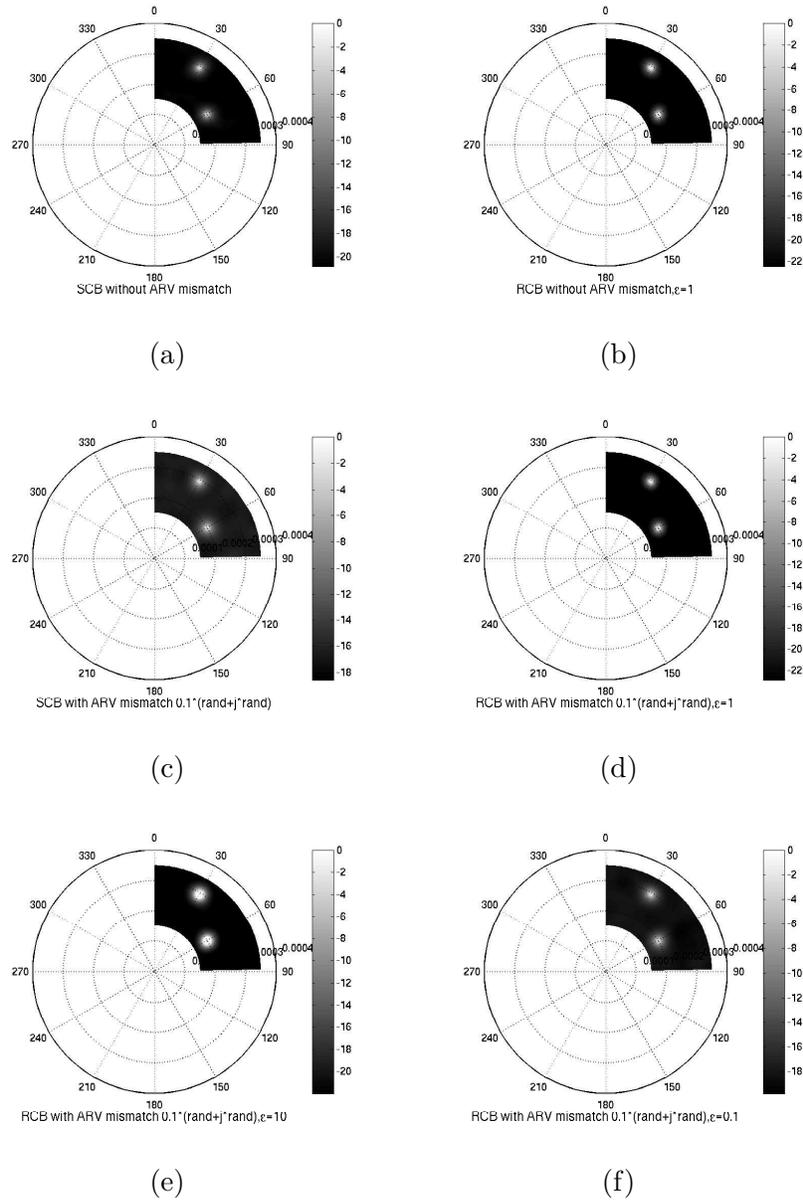


Figure 3.5: Power estimation for a 10by10 square array using RCB and SCB in Case 3: (a) Power estimation of the SCB method without ARV mismatch (b) Power estimation of the RCB method without ARV mismatch, $\epsilon = 1$ (c) Power estimation of the SCB method with ARV mismatch (d) Power estimation of the RCB method with ARV mismatch, $\epsilon = 1$ (e) RCB power estimation with ARV mismatch, and ϵ is chosen bigger than the true value, $\epsilon = 10$ (f) RCB power estimation with ARV mismatch and ϵ is chosen smaller than the true value, $\epsilon = 0.1$

4.1 Problem Statement

In geophysical applications, sensor arrays for monitoring seismic events are normally non-uniformly positioned, in order to obtain as many spatial samples as possible at different locations. By focusing on only one wave type in seismic processing, the spectral estimation performance can be improved. However, filtering is required to filter out surface waves or body waves. Since the design of a non-uniformly sampled digital filter is very complex, we propose to first reconstruct the irregularly sampled spatial signal in a uniform grid, and then apply a velocity filter to remove unwanted signals. This reduces the complexity since a uniformly sampled velocity filter can be constructed in a straight forward way.

In the previous chapter, we used Capon and Robust Capon beamforming to estimate the wavenumber-frequency spectrum of the seismic data. Both methods apply a linear constraint to the array response vector. In principle, we could also implement linear constraints to cancel the interference signals. The problem, however, is that the number of interferences we have in seismic data is unknown (maybe big). Also, if the interferences are close to one another, they are correlated. This means cancelling one interference influences the cancellation of another. In order to cancel multiple interference signals, the linear constraint matrix for beamforming has to be constructed. Since the constraint vectors are correlated to each other, the constraint matrix is rank deficient [Van Trees (2002)], which will deteriorate the estimation performance. Therefore, we will concentrate on spatial signal reconstruction [Zwartjes (2005)] and filtering in this section. First the basic methodology will be introduced, then each section will be analyzed in detail, the simulation results of the reconstruction and filtering algorithm

1. Split the time series data at each sensor into frequency bins, $\mathbf{p}^{(i)}$ is the $M \times 1$ vector of frequency components of interest at time i collected from each sensor, $i \in [1, I]$
2. Estimate the wavenumber-frequency spectrum $\tilde{\mathbf{p}}^{(i)}$ for the i -th snapshot $\mathbf{p}^{(i)}$, which is obtained in step 1.
3. Reconstruct the signal in a uniform grid using the Fourier Reconstruction with Sparse Inversion(FRSI) method, thus a new data vector $\hat{\mathbf{p}}^{(i)}$ for the reconstructed array is obtained.
4. Design and apply the 2D velocity filter \mathbf{h} to filter out surface or body waves in the data vector $\hat{\mathbf{p}}^{(i)}$, the filtered data is $\hat{\mathbf{p}}_f^{(i)} = (\mathbf{h} \otimes \hat{\mathbf{p}}^{(i)})$.
5. Repeat step 2-4 for each snapshot, and all of the filtered data vectors $[\hat{\mathbf{p}}_f^{(1)} \cdots \hat{\mathbf{p}}_f^{(I)}]$ form a data matrix with I snapshots.
6. Use beamforming algorithms to estimate the high-resolution wavenumber-frequency spectrum based on the data matrix in step 5.

Figure 4.1: The general steps for filtering of seismic signals

are plotted as well.

4.2 Solution

The indicated problem can be solved by several consecutive steps, as explained in the above box. The basic scheme for reconstruction is to re-sample and interpolate the seismic data spatially in a uniform array by using the Fourier Reconstruction method, as shown in Figure 4.2, where the non-uniform array with 36 sensors is reconstructed on a 10×10 rectangular array. The signal is still considered random in time.

4.2.1 Data Model

For reconstruction of seismic data in a uniform grid, the first step is to estimate the uniformly sampled wavenumber-frequency spectrum. Reconstruction and filtering are applied to each snapshot independently. The estimation of the f-k spectrum $\tilde{\mathbf{p}}^{(i)}$ for the i -th snapshot will be taken as an example.

Suppose $\mathbf{p}^{(i)}$ is the i -th snapshot for the array which is obtained from the received

signal at each sensor. In this thesis, we assume the received signals are random. \mathbf{F} is the inverse spatial Fourier transform matrix which is determined by the sampling structure in the spatial and wavenumber-frequency domains. $\tilde{\mathbf{p}}^{(i)}$ is the wavenumber-frequency spectrum which needs to be estimated based on $\mathbf{p}^{(i)}$. When sampling the wavenumber-frequency spectrum, we can choose the appropriate bandwidth (with respect to wavenumber-frequency spectrum) and sampling interval. In general, the chosen bandwidth is smaller than the full bandwidth of the spatial signal, thus $\mathbf{n}^{(i)}$ is used to represent the unknown noise outside the chosen bandwidth.

The data model for this application is defined as:

$$\mathbf{p}^{(i)} = \mathbf{F}\tilde{\mathbf{p}}^{(i)} + \mathbf{n}^{(i)}, \quad (4.1)$$

where \mathbf{F} is a $M \times N$ matrix, where M is the number of sensors, and N represents the number of wavenumber-frequency spectral samples. In general, $N \gg M$, thus \mathbf{F} is a wide matrix. Δk_x and Δk_y are the wavenumber-frequency spectral sampling intervals along the k_x and k_y axis respectively. $k_{x,n} = n\Delta k_x$ and $k_{y,n} = n\Delta k_y$ are the n -th ($n \in [1, N]$) f-k spectral sample indexes. (x_m, y_m) is the location of the m -th ($m \in [1, M]$) sensor.

F_{mn} is the component in matrix \mathbf{F} with row index m and column index n . $p_m^{(i)}$ is the m -th spatial sample in $\mathbf{p}^{(i)}$, and $\tilde{p}_n^{(i)}$ is the n -th wavenumber-frequency spectral sample in $\tilde{\mathbf{p}}^{(i)}$, they can be written as:

$$\begin{aligned} F_{mn} &= \frac{\Delta k_x \Delta k_y}{4\pi^2} e^{-j(k_{x,n}x_m + k_{y,n}y_m)} \\ p_m^{(i)} &= p^{(i)}[x_m, y_m] \\ \tilde{p}_n^{(i)} &= \tilde{p}^{(i)}[k_{x,n}, k_{y,n}] \end{aligned} \quad (4.2)$$

$\mathbf{p}^{(i)}$ and $\tilde{\mathbf{p}}^{(i)}$ stack all the spatial samples and spectrum samples into vector form, and can be written as:

$$\mathbf{p}^{(i)} = \begin{bmatrix} p_1^{(i)} \\ \vdots \\ p_M^{(i)} \end{bmatrix}, \quad \tilde{\mathbf{p}}^{(i)} = \begin{bmatrix} \tilde{p}_1^{(i)} \\ \vdots \\ \tilde{p}_N^{(i)} \end{bmatrix}. \quad (4.3)$$

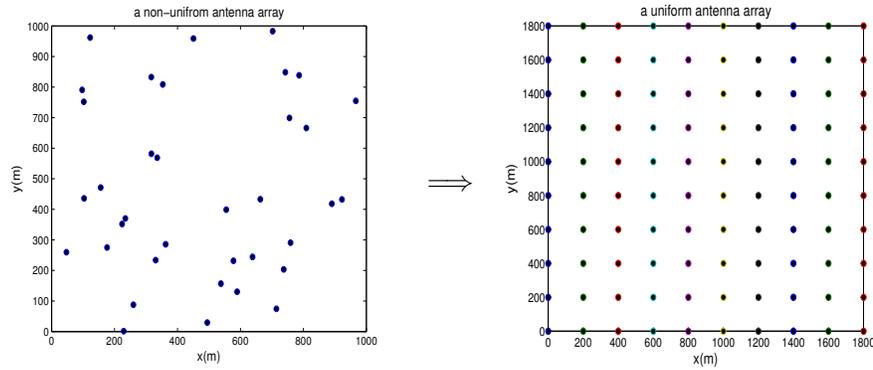


Figure 4.2: The basic scheme of reconstruction

4.2.2 Estimate the wavenumber-frequency spectrum

In order to filter the interferences, we will first reconstruct the non-uniformly sampled seismic data in a uniform grid. Three methods for estimating the wavenumber-frequency spectrum $\tilde{\mathbf{p}}^{(i)}$ for the i -th snapshot will be explained in the following sections. The FRSI method explained in the third section proves to be the best one for reconstructing purposes. Afterwards, inverse Fourier Transform can be used to obtain uniformly sampled spatial signals for further processing (filtering).

1. Objective function without regularization

As can be seen from the Data Model, $\tilde{\mathbf{p}}^{(i)}$ is the unknown parameter which needs to be estimated. The main strategy is to recast the spectrum estimation as an inverse problem. By constructing an objective function J , the inverse problem can be solved efficiently. In least square sense, the objective function is constructed as follows:

$$\begin{aligned} J &= \|(\mathbf{C}_n^{(i)})^{-\frac{1}{2}}(\mathbf{p}^{(i)} - \mathbf{F}\tilde{\mathbf{p}}^{(i)})\|_2^2 \\ &= \frac{1}{c^2} \|\mathbf{W}^{\frac{1}{2}}(\mathbf{p}^{(i)} - \mathbf{F}\tilde{\mathbf{p}}^{(i)})\|_2^2, \end{aligned} \quad (4.4)$$

where $\mathbf{C}_n^{(i)}$ is the noise covariance matrix for the i -th snapshot $\mathbf{n}^{(i)}$ and is given by: $\mathbf{C}_n^{(i)} = c^2 \mathbf{W}^{-1}$, where c is a constant and \mathbf{W} is a data weighting matrix. The weights in \mathbf{W} are defined as $W_{nn} + \Delta x_n$, and are normalized such that $\sum \Delta x_n = 2\pi/\Delta k$. This equation is minimized by the least square estimator, the estimated spectrum $\hat{\mathbf{p}}^{(i)}$ is

written as:

$$\hat{\mathbf{p}}^{(i)} = (\mathbf{F}^H \mathbf{W} \mathbf{F})^{-1} \mathbf{F}^H \mathbf{W} \mathbf{p}^{(i)}. \quad (4.5)$$

However, \mathbf{F} is wide matrix i.e. $N \gg M$, which causes $\mathbf{F}^H \mathbf{W} \mathbf{F}$ to be rank deficient. Because of this, the inverse in Equation 4.5 requires large computation time. Also, a small error in $\mathbf{p}^{(i)}$ will cause a big error in $\hat{\mathbf{p}}^{(i)}$, resulting in poor estimation performance. Therefore regularization is required.

2. Objective function with quadratic regularization term

In this section, we add a quadratic regularization term to the objective function. This method is called Fourier Reconstruction with Minimum Norm (FRMN) [Zwartjes (2005)] [Schonewille (2000)]. The objective function is thus defined as:

$$J = \frac{1}{c^2} \|\mathbf{W}^{\frac{1}{2}}(\mathbf{p}^{(i)} - \mathbf{F}\tilde{\mathbf{p}}^{(i)})\|_2^2 + \frac{1}{(\sigma_{\tilde{\mathbf{p}}}^{(i)})^2} \|\tilde{\mathbf{p}}^{(i)} - \tilde{\mathbf{p}}_0^{(i)}\|_2^2, \quad (4.6)$$

where $(\sigma_{\tilde{\mathbf{p}}}^{(i)})^2$ is the a-priori model variance for the i -th snapshot and $\tilde{\mathbf{p}}_0^{(i)}$ is an a-priori model estimate, which is 0 in our case. The solution to the above equation is given by:

$$\begin{aligned} \hat{\mathbf{p}}^{(i)} &= (\mathbf{F}^H \mathbf{W} \mathbf{F} + \lambda_i \mathbf{I})^{-1} (\mathbf{F}^H \mathbf{W} \mathbf{p}^{(i)} - \lambda_i \tilde{\mathbf{p}}_0^{(i)}) \\ &= [(\mathbf{F}^H \mathbf{W} \mathbf{F} + \lambda_i \mathbf{I})^{-1} \mathbf{F}^H \mathbf{W}] \mathbf{p}^{(i)}, \end{aligned} \quad (4.7)$$

where $\lambda_i = \frac{c^2}{(\sigma_{\tilde{\mathbf{p}}}^{(i)})^2}$ is the damping term for the i -th snapshot, and $\tilde{\mathbf{p}}_0^{(i)} = 0$ is used.

Problems of Fourier Reconstruction with Minimum Norm:

- Poor reconstruction in large spatial gaps, the reason is the damping term λ_i [(Zwartjes, 2005, pp. 18-25)].
- Limited bandwidth for reconstruction, in other words, the number of Fourier coefficients that can be estimated is limited [(Zwartjes, 2005, pp. 18-25)].

3. Fourier Reconstruction with Sparse Inversion

In this section, a non-quadratic regularization term is used instead of a quadratic one [Zwartjes (2005)] in order to yield a sparse solution. The objective function is then:

$$J = \frac{1}{c^2} \|\mathbf{W}^{\frac{1}{2}}(\mathbf{p}^{(i)} - \mathbf{F}\tilde{\mathbf{p}}^{(i)})\|_2^2 + \rho(\tilde{\mathbf{p}}^{(i)}), \quad (4.8)$$

where $\rho(\tilde{\mathbf{p}}^{(i)})$ is the model penalty term (Regularization term), possible model penalty terms are *Cauchy* or *German* functions etc [(Zwartjes, 2005, pp. 32-35)]. For these model penalty terms, the general solution of minimizing J with respect to $\tilde{\mathbf{p}}^{(i)}$ is:

$$\hat{\mathbf{p}}^{(i)} = (\mathbf{F}^H \mathbf{W} \mathbf{F} + c^2 (\mathbf{C}_{\tilde{\mathbf{p}}}^{(i)})^{-1})^{-1} \mathbf{F}^H \mathbf{W} \mathbf{p}^{(i)}, \quad (4.9)$$

where $\mathbf{C}_{\tilde{\mathbf{p}}}^{(i)}$ is the spectral covariance matrix for the i -th snapshot which depends on the chosen model penalty term $\rho(\tilde{\mathbf{p}}^{(i)})$. We assume different spectral coefficients are uncorrelated, thus $\mathbf{C}_{\tilde{\mathbf{p}}}^{(i)}$ is diagonal. The diagonal elements can be written as:

$$(\mathbf{C}_{\tilde{\mathbf{p}},jj}^{(i)})^{-1} = (\tilde{p}_j^{(i)} \tilde{p}_j^{(i)*} + (\sigma_{\tilde{\mathbf{p}}}^{(i)})^2)^{-a}, \quad (4.10)$$

where $a = 1$ for *Cauchy* and $a = 2$ for *German* model penalty terms respectively. In this thesis, the l_1 function is selected as the model penalty term where $a = \frac{1}{2}$.

Strength of the FRSI method:

- Better reconstruction in gaps, the algorithm can cope with large spatial gaps very well [(Zwartjes, 2005, pp. 38-39)].
- Full bandwidth reconstruction, providing a wider estimate of the wavenumber-frequency spectrum [(Zwartjes, 2005, pp. 38-39)].

MAP estimator

A Maximum A Posteriori (MAP) estimator [van der Tol and van der Veen (2007)] can also be used to estimate the wavenumber-frequency spectrum. By assuming the data is *Gaussian* random, the objective function can be recast as:

$$J = \|(\mathbf{C}_n^{(i)})^{-\frac{1}{2}} (\mathbf{p}^{(i)} - \mathbf{F} \tilde{\mathbf{p}}^{(i)})\|_2^2 + \|(\mathbf{C}_{\tilde{\mathbf{p}}}^{(i)})^{-\frac{1}{2}} \tilde{\mathbf{p}}^{(i)}\|_2^2 \quad (4.11)$$

and the solution is given by:

$$\begin{aligned} \hat{\mathbf{p}}^{(i)} &= (\mathbf{F}^H (\mathbf{C}_n^{(i)})^{-1} \mathbf{F} + (\mathbf{C}_{\tilde{\mathbf{p}}}^{(i)})^{-1})^{-1} \mathbf{F}^H (\mathbf{C}_n^{(i)})^{-1} \mathbf{p}^{(i)} \\ &= (\mathbf{F}^H \mathbf{W} \mathbf{F} + c^2 (\mathbf{C}_{\tilde{\mathbf{p}}}^{(i)})^{-1})^{-1} \mathbf{F}^H \mathbf{W} \mathbf{p}^{(i)}. \end{aligned} \quad (4.12)$$

When $\mathbf{C}_{\tilde{\mathbf{p}}}^{(i)} = (\sigma_{\tilde{\mathbf{p}}}^{(i)})^2 \mathbf{I}$, the MAP estimator reduces to the FRMN estimator, and if $\mathbf{C}_{\tilde{\mathbf{p}},jj}^{(i)} = (\tilde{p}_j^{(i)} \tilde{p}_j^{(i)*} + (\sigma_{\tilde{\mathbf{p}}}^{(i)})^2)^{\frac{1}{2}}$, the MAP estimator is equivalent to the FRSI estimator with l_1 model penalty term. Therefore both FRMN and FRSI are MAP estimators, and the objective function J in FRSI can be recast as a quadratic model penalty term, however, with the specified spectral covariance matrix $\mathbf{C}_{\tilde{\mathbf{p}}}^{(i)}$.

4.2.3 Reconstruct the spatial signal in a uniform grid

When the uniform sampled spectrum $\tilde{\mathbf{p}}^{(i)}$ for the i -th snapshot is estimated, we can reconstruct the spatial seismic data in a uniform array by using inverse spatial Fourier Transform. Suppose \mathbf{F}_u is the inverse Fourier Transform matrix for the reconstructed array where the 2D spatial domain is uniformly sampled. The i -th reconstructed data $\hat{\mathbf{p}}^{(i)}$ in a uniform array is given by:

$$\hat{\mathbf{p}}^{(i)} = \mathbf{F}_u \hat{\tilde{\mathbf{p}}}^{(i)}. \quad (4.13)$$

Therefore the original spatial signal has been reconstructed on a uniform array. After the reconstruction algorithm is applied to all of the snapshots, the data matrix $\left[\hat{\mathbf{p}}^{(1)} \dots \hat{\mathbf{p}}^{(I)} \right]$ is constructed, and each column in the matrix will be convolved with the velocity filter to remove the interferences.

4.2.4 Filter the seismic signal

After reconstructing the non-uniformly sampled seismic data in a uniform grid, a velocity filter can be applied to the reconstructed data set $\hat{\mathbf{p}}^{(i)}$, which will filter the data so that the wave of interest will remain. For instance, we can design a low-pass f-k filter to remove surface waves, and the remaining signal can be used for further beamforming applications to find the angle and velocity of the source signal (body waves in this case).

To design the 2D velocity filter, we first select the lowest velocity v_{min} we want to keep, and seismic signals which have higher velocities will remain and others will be filtered out. With respect to wavenumber k , the maximum value k_{max} is used when designing the filter.

$$\begin{aligned} k &= \sqrt{k_x^2 + k_y^2} \\ &= \sqrt{\left(\frac{2\pi f}{v} \sin \theta\right)^2 + \left(\frac{2\pi f}{v} \cos \theta\right)^2} \\ &= \frac{2\pi f}{v} \leq k_{max} = \frac{2\pi f}{v_{min}} \end{aligned} \quad (4.14)$$

From the above equations, $\sqrt{k_x^2 + k_y^2} \leq k_{max}$ occupies a circle field in the (k_x, k_y) plane as shown in Figure 4.3. The wavenumber k corresponds to the frequency when we

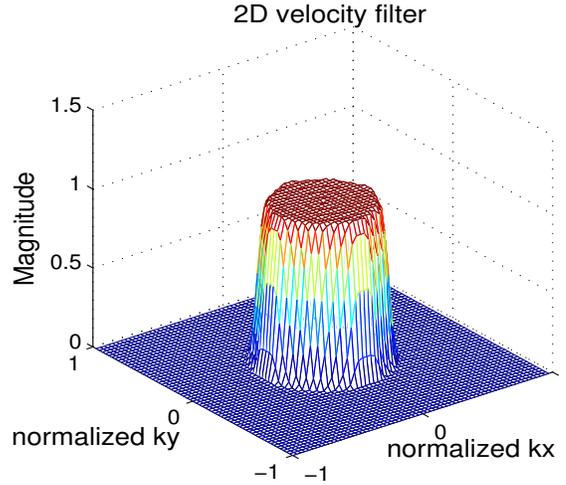


Figure 4.3: An example 2D velocity filter

design a temporal filter, the amplitude for the wavenumber-frequency samples inside the circle is 1 and outside is 0, thus this velocity filter is interpreted as a low-pass f-k filter. The filter phase is chosen as 0 in this thesis. The filter frequency response $H(k_x, k_y)$ can be written as:

$$H(k_x, k_y) = \begin{cases} 1, & \sqrt{k_x^2 + k_y^2} \leq k_{max} \\ 0, & \text{otherwise} \end{cases}$$

Then the filter impulse response can be obtained by inverse Fourier Transform to $H(k_x, k_y)$. In order to suppress the ripples in the wavenumber-frequency domain, a 2D Hamming window is multiplied with the filter impulse response. The general steps for the design of a 2D velocity filter is listed below.

1. Select the lowest velocity v_{min} for the filter, which further defines the maximum wavenumber $k_{max} = \frac{2\pi f}{v_{min}}$.
2. Design the filter in the frequency domain, set the amplitude inside the circle field $\sqrt{k_x^2 + k_y^2} \leq k_{max}$ to be 1, outside as 0, and select the filter phase as 0.
3. Use Inverse Fourier Transform to obtain the impulse response $h(x, y)$ for the filter, and multiply it with a 2D window in order to suppress ripples in the wavenumber-frequency domain.

Then the designed filter will be convolved with the reconstructed spatial signals, thus the data matrix obtained after filtering is represented as:

$$\mathbf{Y} = \left[\hat{\mathbf{p}}_f^{(1)} \cdots \hat{\mathbf{p}}_f^{(I)} \right], \quad (4.15)$$

where $\hat{\mathbf{p}}_f^{(i)}$ is the filtered signal, consisting of spatial samples after applying reconstruction and filtering steps to the snapshot $\mathbf{p}^{(i)}$.

4.2.5 Practical issue: Implementation of the FRSI method

At time i , the cost function which needs to be minimized in the sparse reconstruction algorithm is:

$$J = \frac{1}{c^2} \|\mathbf{W}^{\frac{1}{2}}(\mathbf{p}^{(i)} - \mathbf{F}\tilde{\mathbf{p}}^{(i)})\|_2^2 + \|(\mathbf{C}_{\tilde{p}}^{(i)})^{-\frac{1}{2}}\tilde{\mathbf{p}}^{(i)}\|_2^2, \quad (4.16)$$

where

$$\mathbf{C}_{\tilde{p},jj}^{(i)} = (\tilde{p}_j^{(i)} \tilde{p}_j^{(i)*} + (\sigma_{\tilde{p}}^{(i)})^2)^{\frac{1}{2}}. \quad (4.17)$$

Suppose $\mathbf{z} = (\mathbf{C}_{\tilde{p}}^{(i)})^{-\frac{1}{2}}\tilde{\mathbf{p}}^{(i)}$, and $\tilde{\mathbf{F}} = \mathbf{F}(\mathbf{C}_{\tilde{p}}^{(i)})^{\frac{1}{2}}$, then the cost function J becomes:

$$J = \frac{1}{c^2} \|\mathbf{W}^{\frac{1}{2}}(\mathbf{p}^{(i)} - \tilde{\mathbf{F}}\mathbf{z})\|_2^2 + \|\mathbf{z}\|_2^2. \quad (4.18)$$

Minimizing the above cost function with respect to \mathbf{z} yields the solution:

$$\mathbf{z} = (\tilde{\mathbf{F}}^* \mathbf{W} \tilde{\mathbf{F}} + c^2 \mathbf{I})^{-1} \tilde{\mathbf{F}}^* \mathbf{W} \mathbf{p}^{(i)}. \quad (4.19)$$

After estimating \mathbf{z} , the wavenumber-frequency spectrum of the seismic signal is:

$$\tilde{\mathbf{p}}^{(i)} = (\mathbf{C}_{\tilde{p}}^{(i)})^{\frac{1}{2}} \mathbf{z}. \quad (4.20)$$

In order to calculate \mathbf{z} , the inversion of a big matrix needs to be computed which will consume much time. To solve this issue, the Conjugate Gradient (CG) method is used to find the solution of \mathbf{z} . Assuming $\mathbf{B} = (\tilde{\mathbf{F}}^* \mathbf{W} \tilde{\mathbf{F}} + c^2 \mathbf{I})$ and $\mathbf{g} = \tilde{\mathbf{F}}^* \mathbf{W} \mathbf{p}^{(i)}$, we have:

$$\mathbf{B}\mathbf{z} = \mathbf{g}, \quad (4.21)$$

thus CG can be adopted to solve the above equation efficiently without matrix inversion. In our simulations, we choose \mathbf{W} is an identity matrix, thus \mathbf{B} is a positive definite

matrix. For any non-zero N by 1 vector y , we have:

$$\begin{aligned} y^* \mathbf{B} y &= y^* \tilde{\mathbf{F}}^* \tilde{\mathbf{F}} y + c^2 y^* y \\ &= (\tilde{\mathbf{F}} y)^* (\tilde{\mathbf{F}} y) + c^2 y^* y > 0. \end{aligned}$$

Beyond CG, the Iterative Reweighted Least Squares (IRLS) algorithm is also adopted to update the wavenumber-frequency spectrum covariance matrix $\mathbf{C}_{\tilde{p}}^{(i)}$. After CG, the estimated wavenumber-frequency spectrum $\tilde{\mathbf{p}}^{(i)}$ is used to compute $\mathbf{C}_{\tilde{p}}^{(i)}$ for the next IRLS loop. The whole process for estimating $\tilde{\mathbf{p}}^{(i)}$ including CG and IRLS is explained below.

Implementation of the FRSI algorithm

Initialization: $\tilde{\mathbf{p}}^{(i)} = 0$, $\mathbf{z} = 0$.

Calculate $\mathbf{C}_{\tilde{p}}^{(i)}$ from Equation 4.17.

IRLS: loop size (2-5 iterations).

$k = 0$.

$$\mathbf{B} = (\mathbf{C}_{\tilde{p}}^{(i)})^{\frac{1}{2}} \mathbf{F}^* \mathbf{W} \mathbf{F} (\mathbf{C}_{\tilde{p}}^{(i)})^{\frac{1}{2}} + c^2 \mathbf{I}.$$

$$\mathbf{r}_k = (\mathbf{C}_{\tilde{p}}^{(i)})^{\frac{1}{2}} \mathbf{F}^* \mathbf{W} \mathbf{p}^{(i)} - \mathbf{B} \mathbf{z}.$$

$$\mathbf{d}_k = \mathbf{r}_k.$$

$$\mathbf{z}_k = \mathbf{z}.$$

CG: while $\|\mathbf{r}_k\|^2 > \text{threshold}$

$$\alpha_k = \frac{\|\mathbf{r}_k\|^2}{\mathbf{d}_k^* \mathbf{B} \mathbf{d}_k}.$$

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \alpha_k \mathbf{d}_k.$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{B} \mathbf{d}_k.$$

$$\beta_k = \frac{\|\mathbf{r}_{k+1}\|^2}{\|\mathbf{r}_k\|^2}.$$

$$\mathbf{d}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{d}_k.$$

$$k = k + 1.$$

$$\tilde{\mathbf{p}}^{(i)} = (\mathbf{C}_{\tilde{p}}^{(i)})^{\frac{1}{2}} \mathbf{z}_k.$$

Update $\mathbf{C}_{\tilde{p}}^{(i)}$ using the new derived $\tilde{\mathbf{p}}^{(i)}$, and $\mathbf{z} = \mathbf{z}_k$, go to **IRLS**.

4.2.6 Simulations

To solve the FRSI problem, the Iterative Reweighted Least Squares (IRLS) method plus the Conjugate Gradient for Normal Equations algorithm (CGNE) will be used. As can be seen from Equation 4.10, the diagonal elements of the covariance matrix need to be updated at each iteration of the IRLS method. CGNE is used for computing the matrix inverse term in Equation 4.9. For FRMN, direct inversion can be applied to find the optimal estimation.

4.2.6.1 Reconstruction

The original array is an irregular array with 25 sensors. Suppose two random source signals are impinging on the array, with arrival angle 30 and 60 degree, and velocity $2000m/s$ and $5000m/s$ respectively. The noise received by the array is also random. The non-uniformly sampled spatial signal is reconstructed on a 20×20 uniform array by the FRSI method, the wavenumber-frequency spectrum for both the original received signal and the reconstructed signal are estimated by the conventional beamforming method. The simulation results are shown in Figure 4.5.

As can be seen from Figure 4.5(c) and (d), the wavenumber-frequency spectrum for the reconstructed signal has less sidelobes than that of the original received signal, and the estimation resolution is also increased. This is because of the reason that the FRSI method will yield a sparse solution of the estimation problem where the noise (sidelobes) will be suppressed, i.e. FRSI estimated a very sparse wavenumber-frequency spectrum for the data.

The first step of reconstruction is to estimate the wavenumber-frequency spectrum for the original received signal. The wavenumber-frequency spectrum estimated by the FRSI method is plotted in Figure 4.4. The FRSI method will use this spectrum to reconstruct the spatial seismic data in a uniform 2D array. As can be seen from the plot, FRSI yields a sparse estimation of the seismic data's f-k spectrum, since the sampling interval dk_x and dk_y in the wavenumber-frequency domain are being set to very small values.

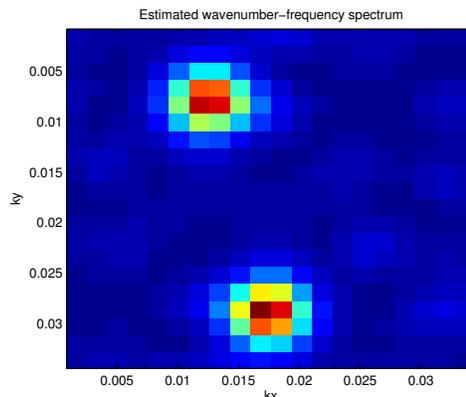


Figure 4.4: The estimated wavenumber-frequency spectrum using FRSI

4.2.6.2 Filtering

The velocities of the two sources are $2000m/s$ and $5000m/s$ respectively. By designing a 2D velocity filter with minimum velocity $v_{min} = 3500m/s$, the low speed source is filtered. In order to obtain better resolution, here the Robust Capon Beamforming method is used to estimate the wavenumber-frequency spectrum of the signals. The simulation results are plotted in Figure 4.6, where (a) and (b) are the wavenumber-frequency spectrum for the original received signal and the reconstructed signal. (c) is the 2D velocity filter with length 21 at each of the axis, the wavenumber frequency k_x and k_y are being normalized into the range $[-1, 1]$. As shown in Figure 4.6(d), the estimated wavenumber-frequency spectrum for the filtered signal excludes the surface wave which has a lower propagation speed.

4.2.6.3 Apply the reconstruction algorithm to real data

In this section, we will apply the reconstruction algorithm to the real data recorded from the potential oil field. The original array is again a 25 sensor array, and Fourier Reconstruction with Sparse Inversion is selected as the reconstruction method, to reconstruct the data into a 20by20 square array. Filtering was not applied to the data

since body waves are very weak in this case. The reconstructed data can be used for later interpolation purpose. The simulation results are shown in Figure 4.7.

4.2.6.4 Suggestions on selecting different array structures

When we use the array plotted in Figure 4.5(b) to receive the same signal present at the original irregular array with 25 sensors (shown in Figure 4.5(a)), the output power computed by the conventional beamforming method and RCB method are shown in Figure 4.8 and Figure 4.9 respectively. Compare Figure 4.5(d) and Figure 4.6(b), to Figure 4.8 and Figure 4.9, we can see that the wavenumber-frequency spectrum estimated by the reconstructed array and a 20by20 array has no big differences. The reconstruction is based on an irregular array with only 25 sensors, which is much cheaper than adopting a 20by20 square array. In the future, we can first use small size array to sample the wave fields, and then reconstruct the sampled spatial data in a bigger size array.

4.3 Conclusion

In this section, several spatial signal reconstruction algorithms are presented and compared, and a 2D cylinder velocity filter is designed to remove surface waves in seismic application. Fourier Reconstruction with Sparse Inversion (FRSI) was adopted for reconstruction purposes.

In order to achieve better resolution and interference cancellation for beamforming after the reconstruction and filtering steps, more advanced beamforming (e.g. RCB) algorithms can be used for this purpose. In the future, the possibility for merging the reconstruction and filtering steps into one single step, will be considered, since this will reduce the complexity of our algorithm.

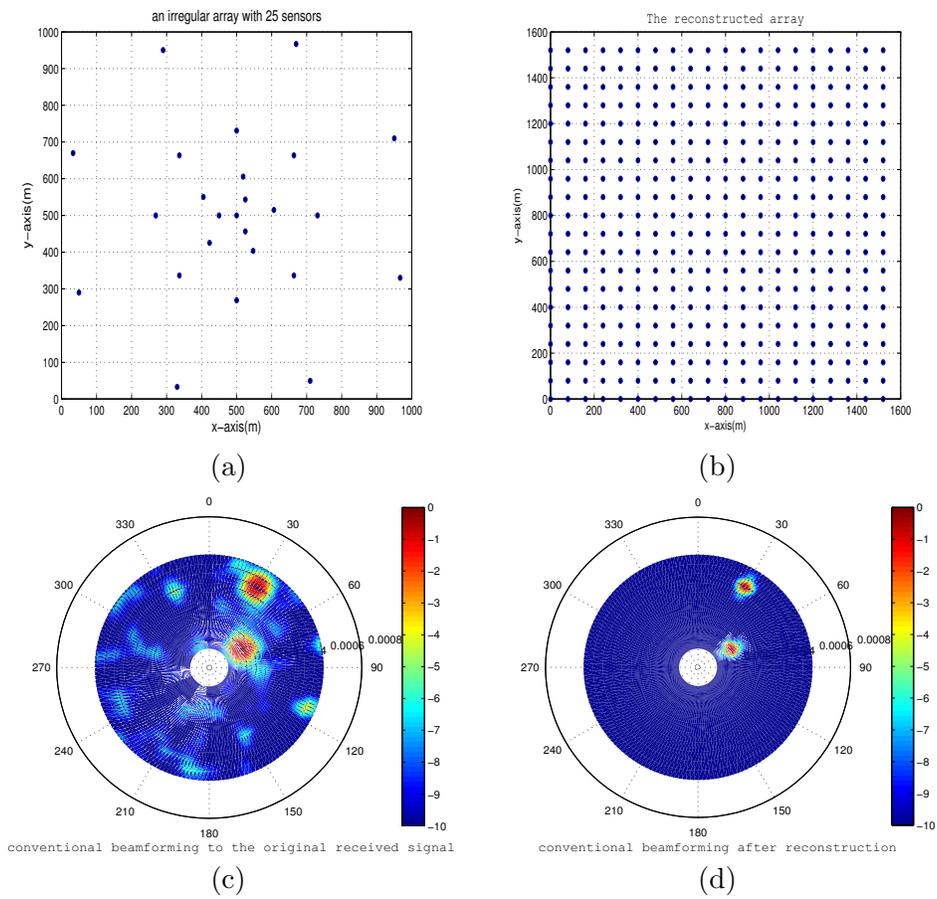


Figure 4.5: Reconstruction of an irregular array to a uniform array: (a) A non-uniform array with 25 sensors (b) The uniform 20by20 array after reconstruction (c) conventional beamforming to the received signal (d) Conventional beamforming to the reconstructed signal.

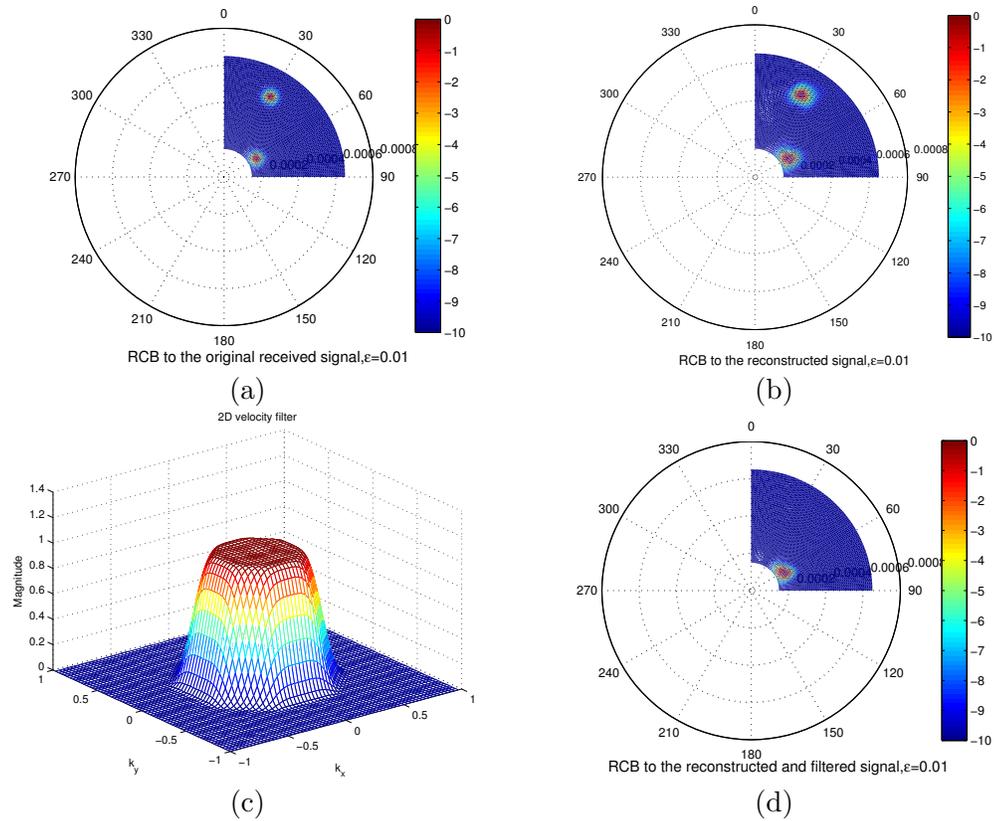


Figure 4.6: Reconstruction and filtering of seismic signal: (a) Apply RCB to estimate the received signal's f-k spectrum (b) Apply RCB to estimate the reconstructed signal's f-k spectrum (c) The 2D velocity filter for removing surface waves (d) Estimated f-k spectrum for the filtered signal via RCB

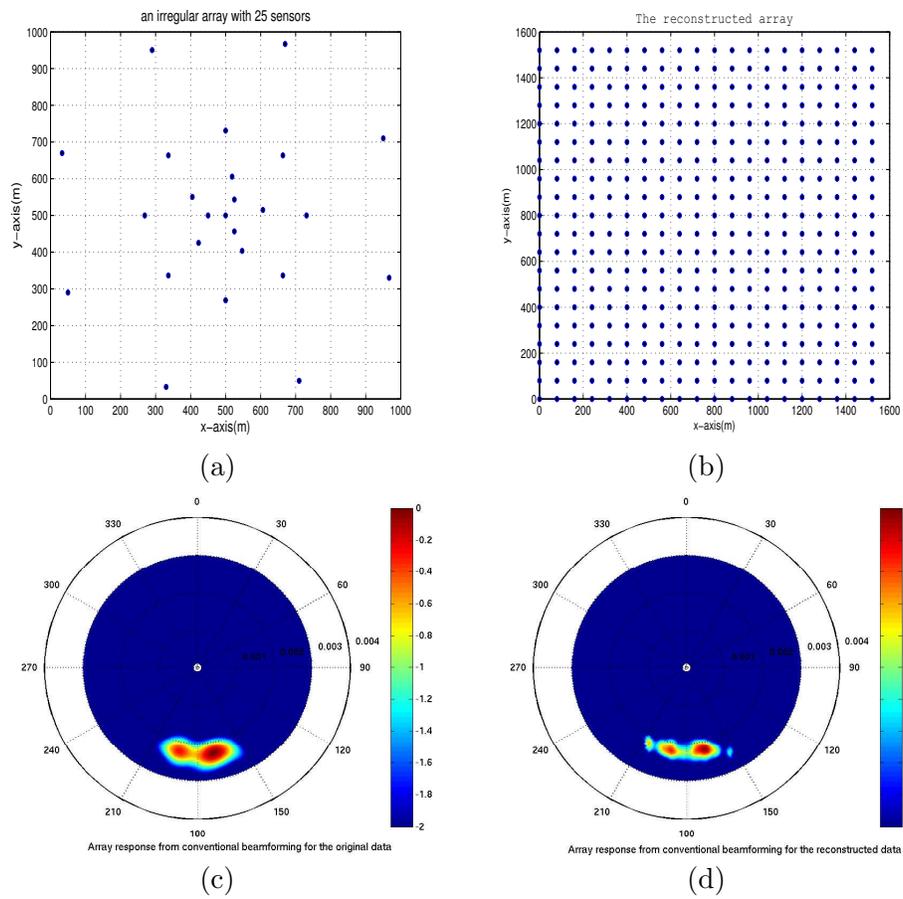


Figure 4.7: Reconstruction algorithm (FRSI) applied to real data: (a) A non-uniform array with 25 sensors (b) The uniform 20by20 array after reconstruction (c) conventional beamforming to the received signal (d) Conventional beamforming to the reconstructed signal.

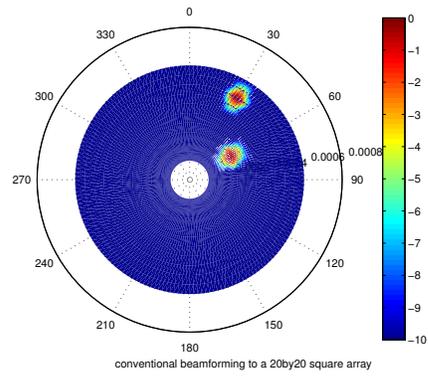


Figure 4.8: Conventional beamforming to a 20by20 square array

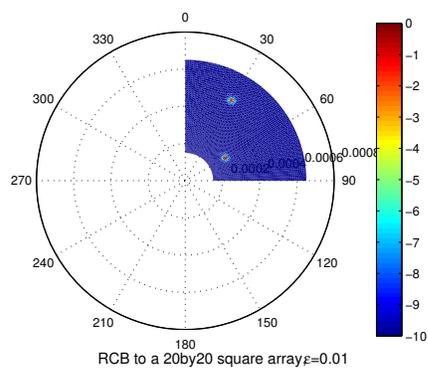


Figure 4.9: Robust Capon Beamforming to a 20by20 square array

Conclusion and Future Work

In this thesis, first a narrow-band data model was introduced for beamforming applications in characterizing seismic noise. Beamforming algorithms for estimating velocities and angles are discussed. Data were simulated based on the model. Conventional beamforming is one of the methods, however, its estimation resolution and interference cancellation are not sufficient, therefore the performance of the conventional method needs to be improved. We introduced the standard Capon beamforming method, which has a much better estimation resolution and interference rejection capability than conventional beamforming. Beside its advantages, the limitations of the standard Capon method are: it requires a rather accurate array response vector, and the covariance matrix must be non-singular. If one of the conditions is not met, the performance of the Capon's method is degraded and usually worse than the conventional beamforming method.

In real applications, either an accurate array response vector is very difficult to achieve, or a full rank covariance matrix is not always the case. To overcome these problems, Robust Capon Beamforming is introduced to tackle these problems. For solving the singular covariance matrix problem, an appropriate diagonal element is added to the covariance matrix. By searching within a spherical uncertainty set, the accurate array response vector is found.

For separating body waves and surface waves, a new idea was proposed in this thesis. That is, first the non-uniformly sampled spatial signal is reconstructed on a uniform array, and then a digital velocity filter is adopted for separating different types of waves. Several reconstruction algorithms were presented and compared, and the FRSI method was selected for reconstruction purposes. A 2D Hamming window was applied in order to suppress spectral ripples in the frequency domain when designing

the filter. After the data have been reconstructed and filtered, beamforming algorithms were applied to find the angle and velocity information for remaining seismic signal. The reconstructed and filtered data can also be used for interpolation purpose which is an important application in seismic processing.

5.1 Recommendations for Future Work

- This thesis mainly focused on developing theory for seismic array processing applications, only a small piece of real data has been simulated. In the future, the introduced algorithms will be applied to more real data.
- In the reconstruction section, FRSI was proven to be the best algorithm. To implement it, IRLS and CGNE algorithm have been applied. Further work for optimizing and analyzing these algorithms should be carried out.
- After reconstructing and filtering the data in several frequencies of interest, the processed data can be used for interpolation in time domain, which is also an important application in seismic processing.
- To reduce and simplify the computations, the possibility to merge the reconstruction and filtering steps should be considered in the future.

Bibliography

- Capon, J. (1969) *High-resolution frequency-wavenumber spectrum analysis*, Proceedings of the IEEE, **57**, pp. 1408–1418.
- Johnson, D. H. and Dudgeon, D. E. (1993) *Array Signal Processing: Concepts and Techniques*, Prentice Hall.
- Li, J. and Stoica, P. (2005) *Robust Adaptive Beamforming*, Wiley.
- Li, J., Stoica, P., and Wang, Z. (2003) *On robust capon beamforming and diagonal loading*, IEEE Transaction on Signal Processing, **51**, pp. 1702–1715.
- Quarteroni, A., Sacco, R., and Saleri, F. (2007) *Numerical Mathematics*, Springer.
- Rost, S. and Thomas, C. (2002) *Array seismology: Methods and applications*, Review of Geophysics, **40**, pp. 1–27.
- Schonewille, M. (2000) *Fourier reconstruction of irregular sampled seismic data*, Ph.D. thesis, Delft University of Technology.
- Stoica, P. and Moses, R. (1997) *Introduction to Spectral Analysis*, Prentice Hall.
- van der Tol, S. and van der Veen, A. (2007) *Ionospheric calibration for the lofar radio telescope*, in: *Proceedings of the IEEE*.
- Van Trees, H. L. (2002) *Optimum Array Processing, Part IV: Detection, Estimation and Modulation theory*, Wiley, chap. 6.
- Veen, A.-J. V. D. and Leus, G. (2005) *Signal processing for communications*.
- Yao, H., Campman, X., De Hoop, M., and Van Der Hilst, R. (2009) *Estimation of surface-wave green's function from correlations of direct waves, coda waves, and ambient noise in se tibet*, Phys. Earth Planet. Inter, in press.
- Zwartjes, P. (2005) *Fourier reconstruction with sparse inversion*, Ph.D. thesis, Delft University of Technology.