秋田県立大学大学院博士学位論文

Compressive Sensing Based Low-Complexity and High-Resolution Radar Signal Processing Algorithms

（圧縮センシングに基づく低演算量かつ高解像度のレーダー信号処理アルゴリズム）

Liu Beiyi

劉　貝易

2020 年 3 月

Compressive Sensing Based Low-Complexity and High-Resolution Radar Signal Processing Algorithms

MARCH, 2020

DOCTOR OF ENGINEERING

Liu Beiyi

AKITA PREFECTURAL UNIVERSITY

Contents

Compressive Sensing Based Low-Complexity and High-Resolution Radar Signal Processing Algorithms

(Abstract)

Radar is an object detection system utilizing radio waves to determine the range, angle, or velocity of objects. A class of novel methods based on compressive sensing (CS) have been proposed for digital radar signal processing to achieve a high resolution on both the range and velocity joint estimation (i.e. the time sampling case) and the angle estimation (i.e. the spatial sampling case). However, as will be discussed in detail below, the existing CS-based methods suffer from a high computational complexity for reconstructing signals and a low accuracy for non-Gaussian impulsive noise environment.

In view of this background, the purpose of this research is to develop a class of new CS-based methods having higher resolution, less computational complexity and stronger robustness against non-Gaussian impulsive noise than the existing CS-based methods. Specifically, both of the time sampling case and the spatial sampling case will be intensively investigated, and the main results and contributions are concentrated on the following aspects:

Firstly, for the time sampling case, the range and velocity joint detection problem is in fact a two-dimensional (2D) problem characterized by a matrix representation, while the existing CS-based methods treated this problem by first vectorizing the 2D matrix representation into a 1D vector representation and utilizing the well-known CS algorithms which is originally developed for 1D signal reconstruction. It is the dimension transformation (vectorization) operation that results in a huge measurement matrix for the resultant CS problem and thus causes a high computational complexity. To overcome this difficulty, 2D-CS algorithms are first proposed that can directly solve a 2D signal reconstruction problem, and then 2D-CS-based methods for the range and velocity joint detection problem are established so that the dimension transformation operation is no longer required and the computational complexity can be reduced largely.

Moreover, since the existing CS-based methods for time sampling have been developed based on the assumption of Gaussian noise environment, the existing methods usually suffer from a performance deterioration in a non-Gaussian impulsive noise environment. Therefore, robust-2D-CS-based methods are proposed by introducing new robust cost functions in the 2D-CS-based algorithms so that the performance of CS-based methods for non-Gaussian impulsive noise environment can be significantly improved.

For the spatial sampling case, the existing CS-based methods achieve the high accuracy

of direction-of-arrival (DOA) estimation by gridding densely and searching the entire range of interest, which results in a huge measurement matrix. In addition, the inversion operation for this measurement matrix is required in each iteration of the algorithms. These two problems lead to the high complexity of the existing CS-based methods for DOA estimation. In order to avoid the first problem caused by the dense griding, a two-step method is proposed: first, the range of interest is divided into a relatively low-resolution grid and the conventional beam former is used to quickly identify the candidate or potential areas where true targets may exist; then the candidate areas obtained in the first step are divided into a denser sampling grid and the $\ell_{2,1}$-norm minimization algorithm is utilized to locate the targets with a high-resolution. For the second problem of inversion operation, a new adaptive algorithm is proposed without involving any matrix inversion operation.

Simulation results show that the proposed methods have much lower complexity and higher accuracy than the existing CS-based methods.

Acknowledgement

It would be impossible to complete this thesis work without the kind assistance and encouragement from my supervisors, fellow doctoral students, friends and of course my family.

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Li Xu in Department of Intelligent Mechatronics, Faculty of Systems Science and Technology, Akita Prefectural University (APU). Professor Xu has walked me through all the stages of this thesis. Without his illuminating, this thesis could not have reached its present form. During the study period, his elaborated guidance, considerable encouragement and invaluable discussion make my research of great achievement and my study life unforgettable. He was always ready for discussions or to help me when I was having a difficult time either in life and research. What I have learned from Professor Xu will spur me to go on with the research and to be a good researcher throughout the rest of my life.

I would like to thank Professor Guan Gui for the valuable advice on this research. He bought me into the field of compressive sensing, and carefully guide me on how to start a research. I am grateful to him for his help, patience and kindness to me all the time.

Next, I would like to thank the committee members Professor Xi Zhang (The University of Electro-Communications), Professor Yoji Isota (APU), and Associate Professor Shin-ya Matsushita (APU), for their valuable comments and suggestions to my research. I also sincerely appreciate the help and suggestions of Associate Professor Hiroshi Okamoto (APU).

I would like to thank the past and present members of the System Control Engineering Laboratory for making my study and stay in Japan pleasant. I am also deeply indebted to Dr. Dongdong Zhao and Dr. Qiaoqiao Li for sharing their precious experiences with me and selfless assistance in my work and life.

Finally, I thank my parents and my friends, who always have believed in me, encouraged me, and given me all their support.

Abbreviations

EBA	Emergency Brake Assist
CS	Compressive Sensing or Compressed Sensing
OMP	Orthogonal Matching Pursuit
CoSaMP	Compressive Sampling Matching Pursuit
IHT	Iterative Hard Thresholding
ZAP	Zero-Attractive Projection
CRT	Chinese Remainder Theorem
ADC	Analog-to-Digital Converter
RF	Random Filter
RC	Random Convolution
RD	Random Demodulation
MWC	Modulated Wideband Converter
NUS	NonUniform Sampling
SMV	Single Measurement Vector
MMV	Multiple Measurement Vectors
SVD	Singular Value Decomposition
LMS	Least Mean Square
SNR	Signal-to-Noise Ratio
ULS	Underspread Linear System
RCS	Radar Cross Section

ISTA	Iterative Shrinkage-Thresholding Algorithm
FISTA	Fast Iterative Shrinkage-Thresholding Algorithm
ULA	Uniform Linear Array
UPA	Uniform Plane Array
UCA	Uniform Circular Array
CBF	Conventional Beam Former

List of Algorithms

Chapter 1

Introduction

1.1 Background and motivation

Radio detection and ranging (Radar) is a target detection system utilizing radio waves to determine the range, angle, or velocity of targets. It has been applied in many fields, such as ground penetrating radar, ballistic missile defense, air-traffic control, law-enforcement, highway safety and automatic driving [1–4]. The demand for modern radar systems with high resolution is increasing. The higher bandwidth radar signals and more antenna are required, which brings the high sampling rates and large sensor array, and according by the large amount of sampled data require a vast memory capacity [5, 6].

In contrast, only few targets, or sources are concerned in practice, such as some airplanes in the wide sky, thus the final output is often sparse. An example of sparse scene is shown in Fig.1.1. It is not efficient to detect the sparse target utilizing a large amount of sampled data. Therefore, the topic of developing new radar signal processing algorithms to accomplish the high resolution with a few sampled data has attracted considerable attention.

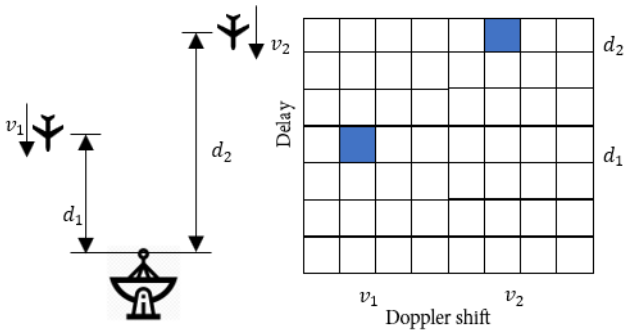


Figure 1.1: The delay-Doppler map.

The research in this thesis is focused on two missions: (1) delay-Doppler joint estimation by ultra wide band (UWB) radar (range and velocity detection) and (2) direction-of-arrival (DOA)

estimation by multiple-input-multiple-output (MIMO) radar (angle detection). They do not only represent the hot research topics nowadays but also are of crucial importance for human societies, as argued in this chapter.

The sampled data of echoes corresponding to different pulses is used for the delay-Doppler joint estimation. This sampling process is called "time sampling". It is widely believed that the range resolution is significantly proportional to the bandwidth of the transmitted signal [7]. With the development of wireless communication, X-band or Ku-band signal cannot satisfy the demand for resolution for range detection. The use of the UWB can greatly improve the range resolution to meet modern industrial demand. Additionally, the UWB signal can improve the robustness against the multipath interference [8]. However, according to the Shannon-Nyquist theorem, the bandwidth of the signal is wider, the faster sampling rate is required. Therefore, the sampling rate for UWB radar systems is a large cost, even difficult to achieve for the analog-to-digital converter (ADC), and a large memory capacity for large amounts of sampled data is required [5, 6]. There, the traditional sampling method based on the Shannon-Nyquist theorem is difficult to satisfy the requirement of UWB signals.

The data sampled by multiple antenna is used for DOA estimation. This sampling process is called "spatial sampling". Multiple antennas are used in MIMO radar systems, and each transmitting antenna radiates the electromagnetic wave independently with other transmitting antennas, while each receiving antenna receives these signals. Unlike the standard phased-array radar system which also uses multiple antennas, the wave transmitted by MIMO radar systems may be chosen quite freely. This waveform diversity makes the MIMO radar system has better performance for, such as, spatial resolution, robustness to interference and/or probability of detection, than the standard phased-array radar system. Due to its excellent performance, MIMO radar systems have been receiving increasing attention in last few decades. Many high-resolution DOA algorithms for the MIMO radar systems have been documented in the literature (see, e.g., [9, 10], and the references therein), and most of these methods depend on the statical properties of the collected data. Thus, a sufficiently large number of snapshots are required for accurate estimation [9–13]. However, in many practical applications, due to various physical constraints, only a few snapshots or, in the worst case, a single snapshot can be used for DOA estimation [14–16]. For example, in automotive radar systems, adaptive cruise control (ACC) and/or emergency brake assist (EBA) require the real-time estimation of the distances and speeds with respect to other targets as well as the DOAs. Due to the rapid change of the relative positions, the number of available snapshots is heavily restricted. Another example can be found in sonar processing where the physical constraints, such as sound speed, limits the

number of available snapshots. Thus, the traditional high-resolution DOA algorithms cannot satisfy the requirement for the limited snapshots. Therefore, it is of great interest to detect the sparse targets with high resolution more effectively and efficiently by establishing new methods to reduce the large amounts of sampled data.

In the last decades, an innovative signal processing method called "*compressive sensing*" or "*compressed sensing*" (CS) is proposed in [17–19]. In a system of underdetermined linear equations — there are fewer equations than unknowns, if there are a lot of zeros in the solution, CS is a powerful mathematical tool to solve it. Such a solution is called "*sparse*" solution. Hence, CS is able to reduce the sampling data which can be represented as sparse vector in a specific domain [17–20]. The CS-based methods have been first applied to radar signal processing for time sampling as well as for spatial sampling [6].

1.1.1 Time sampling

For the time sampling case, a class of novel sampling methods which are evolved from CS theory have attracted considerable research attention, such as the random convolution (RC) [21], the random filter (RF) [22], the random demodulation (RD) [23–26], and the modulated wideband converter (MWC) [27–29]. The RD and MWC can both be thought of as being based on the underlying concept of the RF and RC. These sampling methods enable the successful reconstruction of signals by sampling at the sub-Nyquist rate, which is slower than the Nyquist rate. The RD shares the similar structure to the MWC, while the RD is a single channel sub-Nyquist sampling strategy and the MWC is a multi-channel sub-Nyquist sampling strategy [30]. Another sampling method based on CS is nonuniform sampling (NUS) method [31–35]. The sampling interval of the NUS method is controlled by an unequal interval clock. But the random change of the sampling rate is quite difficult to implement on hardware. Among them, the RD is a very attractive technique in microwave spectral analysis [36], because it is very simple and its prototype hardware has already been developed in [37, 38].

A lot of CS-based algorithms have been proposed to reconstruct the signal sampled by sub-Nyquist sampling rate [39–43]. Orthogonal matching pursuit (OMP) and compressive sampling matching pursuit (CoSaMP) algorithms [39] are the greedy method that builds up the support set of the reconstructed sparse vector iteratively by adding indices of the elements that are non-zero to the current support set at each iteration. Iterative hard thresholding (IHT) algorithm [40] is a thresholding method that keeps the indices of the non-zero elements of the sparse vector by a hard thresholding at each iteration. Zero-attractive projection (ZAP) algorithm [41] utilizes the zero attraction term to attract the vector from a least-square solution to a sparse solution. The

feasibility of CS-based algorithms for delay-Doppler joint estimation has been shown in [44–50]. Although these methods estimate both the delay and Doppler frequency from few sampled data, there are still the following problems:

- The traditional CS-based algorithms only consider the one-dimensional (1D) problem, while the delay-Doppler joint estimation is a two-dimensional (2D) problem. In order to use the 1D CS-based algorithms to solve this the 2D problem, one has to stack the 2D signal into a huge column vector based on the vector space, and then recover the huge vector in the 1D domain. However, such an operation increases the complexity and memory usage exponentially [51, 52].

- The traditional CS-based algorithms only consider the Gaussian noise environment. Gaussian noise model is an ideal model, which does not consider the impulsive noise caused by atmospherics (lighting), meteor train echoes and so on [53]. In UWB radar systems, the accuracy of the traditional CS-based algorithms is easily deteriorated by these impulse noises.

1.1.2 Spatial sampling

For the spatial sampling case, a novel class of DOA estimation methods based on CS theory has been proposed for the limited snapshots. In the worst case, i.e., under the assumption that only a single snapshot is available, the single snapshot DOA estimation problem can be formulated as a problem of finding a sparse representation of Single Measurement Vector (SMV) (i.e., a vector having small number of nonzero entries) [54, 55], and ℓ_1-singular value decomposition (ℓ_1-SVD) algorithm is recognized as a powerful sparse recovery algorithm for DOA estimation with single snapshot [56]. It incorporates the SVD step of the subspace algorithms into a sparse recovery method with ℓ_1-norm penalty so that closely spaced correlated sources can be dealt with effectively.

On the other hand, for the case having a few snapshots (more than one single snapshot, but much fewer than that required by the traditional methods) available, in order to improve the accuracy, the DOA estimation problem can be formulated as a Multiple Measurement Vectors (MMV) problem in an over-complete dictionary, i.e., the problem of finding joint-sparse representations (matrices having small number of rows that contain nonzero entries) of MMV (see, e.g., [10, 55]). Based on MMV framework, some CS-based algorithms have recently been proposed for DOA estimation with a few snapshots by essentially generalizing the methods for the SMV case to the MMV case [54–56]. The joint $\ell_{2,0}$ approximation DOA (JLZA-DOA) al-

gorithm solves the joint-sparse solution by minimizing a mixed $\ell_{2,0}$-norm approximation. It has been demonstrated by numerical simulations that the JLZA-DOA algorithm can achieve high-resolution by using only a small number of snapshots [10]. Along the similar line, other kinds of extensions of the ℓ_1-SVD algorithm have also been reported [57, 58]. The Noise Subspace Weighted ℓ_1-ℓ_2 (NSW-ℓ_1-ℓ_2) algorithm [57] is established based on weighted ℓ_1 minimization, with the weights determined by utilizing the orthogonality between the noise subspace and the measurement matrix. In [58], the Capon spectrum is employed to design a weighted ℓ_1-norm penalty to get a better ℓ_1-norm approximation and further enforce the sparsity so that better DOA estimation performance can be achieved.

Although these methods can estimate DOA from limited snapshots, there are still the following problems:

- All these algorithms have to search the potential DOAs or areas actually containing sources over the entire range of interest by a convex optimization algorithm. To obtain a high resolution result, the entire range of interest should be divided into a dense sampling grid, which requires operation of a measurement matrix with very high dimension in every iteration and thus results in a rather heavy computational cost.

- Since the number of antennas becomes very large in MIMO systems, the amount of data that need to be processed also becomes very huge. It results in a huge measurement matrix, whose matrix inversion operation is required in each iteration for these CS-based algorithms. Thus, the computational cost of the CS-based algorithms is still rather heavy for MIMO systems, although only a small number of snapshots are used [59].

1.2 Main contributions

In this thesis, we focus on reducing the complexity and improving accuracy of the CS-based algorithms for high-resolution radar signal processing. As with previous studies, we divide radar signal processing into time sampling case for delay Doppler joint estimation and spatial sampling case for DOA estimation, and study them separately.

For the time sampling case, in order to reduce the complexity caused by dimension transformation, a class of the 2D-CS-based algorithms is established for delay Doppler joint estimation whose data is sampled by the RD with a low rate ADC. Since the 2D-CS-based algorithms solve the 2D data model without stacking the matrix of 2D signals into a huge column vector, the memory requirement and complexity can be reduced greatly.

In addition, in order to improve the accuracy in non-Gaussian impulsive noise environment, the robust 2D-CS-based algorithms will be developed based on the proposed 2D-CS-based algorithms. These extended algorithms use the robust cost functions which provide efficiency when the data is contaminated by a small noise and reliable albeit not optimal behavior when the data is contaminated by impulsive noise.

For the spatial sampling case, in order to overcome the difficulty of high computational complexity caused by the dense sampling grid, we develop a two-step method. In the first stage, the range of interest is divided into a relatively low-resolution grid, and the conventional beam former is used to quickly identify the candidate or potential areas where true sources may exist. In the second stage, the candidate areas obtained in the first stage are divided into a denser sampling grid, and the CS-based method is utilized to locate the targets with a high-resolution.

Another method to reduce the complexity is using the adaptive filtering algorithms, e.g., the ℓ_0-least mean square (ℓ_0-LMS) algorithm [41, 60]. It uses a row in measurement matrix in each iteration that leads a low complexity, and has higher robustness against noise than other CS-based algorithms. However, it is proposed for a single snapshot vector (SMV) problem, which corresponds to DOA estimation with a single snapshot [61, 62]. Motivated by the ℓ_0-LMS algorithm, a new adaptive filtering algorithm, e.g., $\ell_{2,0}$-LMS algorithm is proposed for the DOA estimation with small snapshots. The adaptive filter frame makes the proposed algorithm inherit the low complexity, and the mixed norm (approximate $\ell_{2,0}$-norm) is used to improve the accuracy in low signal-to-noise ratio (SNR) environment.

1.3 Thesis outline

The thesis is organized in the following chapters. In Chapter 2, the mathematical model of delay Doppler joint estimation and the CS based RD sampling method are introduced. In Chapter 3, the 2D-CS-based algorithms which use a matrix norm instead of the vector norm to reduce the complexity are proposed. In Chapter 4, the non-Gaussian impulsive noise model are introduced, and the robust 2D-CS-based algorithms which is extended from 2D-CS-based algorithms are proposed. In Chapter 5, the mathematical model of DOA estimation with small snapshots is presented. In Chapter 6, in order to reduce the complexity and improve the accuracy, the Dimension-Reduced CS-based algorithm is proposed for DOA estimation with small snapshots. In Chapter 7, the $\ell_{2,0}$-LMS algorithm based on adaptive filter is proposed to avoid matrix inversion operation. In Chapter 8, the main results and contributions are briefly summarized. In addition, some possible future topics are also discussed.

Chapter 2

Signal model for delay Doppler joint estimation with sub-Nyquist rate

The modern pulse Doppler radar is built by a UWB signal generator, transmitter, receiver, antenna, and signal processing, which is shown in Fig. 2.1. In this chapter, the delay Doppler joint estimation for pulse Doppler radar is introduced.

2.1 Mathematical model for delay Doppler joint estimation

2.1.1 Range and velocity estimation

In general, radar system transmits a pulse signal to achieve the delay-Doppler map which shows the radial distance and radial velocity information of targets. The time which takes one pulse to travel the two-way path between antenna and the kth target is the delay τ_k, where $k = 1, ..., K$, K is the number of the targets. The delay τ_k is measured to compute the radial distance R between the kth target and radar. Since the velocity of the electromagnetic waves is a constant

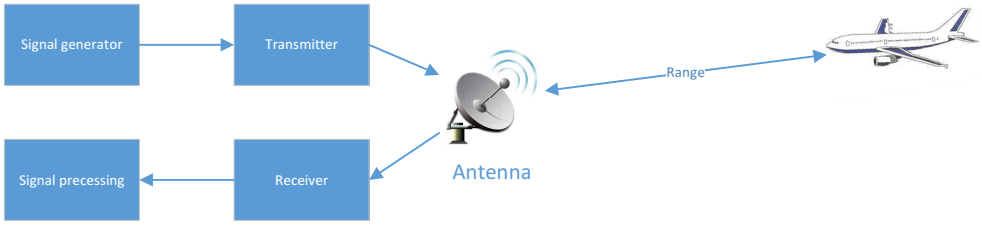


Figure 2.1: A simplified radar block diagram

value, i.e., $c \approx 3 \times 10^8$ m/s, the distance R can be computed by

$$R = \frac{c}{2}\tau_k. \tag{2.1}$$

where τ_k is in seconds, and R is in meters.

The velocity information of the kth target is achieved by Doppler effect, which is the change in frequency of an electromagnetic waves for the target moving relative to antenna. It can be written as follows:

$$\nu_k = \left(\frac{c + v_k}{c + v_s}\right)\nu_0 \tag{2.2}$$

where ν_0 is the frequency of transmitted signal, ν_k is the changed frequency termed "Doppler frequency", v_k denotes the velocity of the kth target, and v_s denotes the velocity of radar. Since in this paper we only consider the monostatic radar whose antenna is stationary in general, v_s equals to 0.

2.1.2 Ambiguity function

In pulse radar system, the ambiguity function is used to show the distortion of a returned pulse. Assume that the transmitted pulse signal is $x(t)$, and it is reflected by one target, then the detected signal which is decided by delay (τ) and Doppler frequency (ν) which can be written as follows:

$$x_{\tau,\nu} = x(t - \tau)e^{j2\pi\nu t}. \tag{2.3}$$

In the noise-free environment, the received signal is a copy of the original pulse but delayed by a certain time τ (delay) related to the target's range and shifted by a certain frequency ν (Doppler frequency) related to the target's velocity. However, it is difficult to solve τ or ν from $x_{\tau,\nu}$ directly, because noise, such as Thermal noise, clutters and interference, exists in the radar system. In general, the correlation between the detected signal and transmitted signal is used to determine whether there is a target. Unfortunately, this method may lead to a wrong position, which are highly correlated with the detected signal. In this sense, the detected signal may be ambiguous. In other words, the ambiguous may occurs when there is a high correlation between $x(\tau, \nu)$ and $x(\tau', \nu')$ while $((\tau, \nu) \neq (\tau', \nu'))$. The ambiguity function is used to define the ambuguous quantitatively. It can be written as follows:

$$|\chi(\tau, \nu)|^2 = \left|\int x(t)x^*(t - \tau)e^{j2\pi\nu t}\right|^2. \tag{2.4}$$

It is similar to a match filter which calculates the correlation between $x(t,0)$ and $x(t-\tau,\nu)$. Only when $(\tau,\nu)=(0,0)$, $|\chi(\tau,\nu)|^2$ is highest. With the same (τ,ν), the peak width of $|\chi(\tau,\nu)|^2$ is narrower, the resolution of $x(t)$ is better.

2.1.3 Linear frequency modulation signal

Different transmitted pulse shapes result in different ambiguity functions. The linear frequency modulation (LFM) is often used in radar transmitted pulse signal due to its better ambiguity function. The LFM signal increases or decreases its frequency with the change of time, linearly. A LFM radar signal can be written as follows:

$$x_T(t)=\frac{1}{\sqrt{T_p}}\text{rect}\left[\frac{t}{T_p}\right]e^{j\frac{B}{T_p}\pi t^2}, \tag{2.5}$$

where B is the bandwidth of transmitted signal, T_p is the pulse width, t is the time, and rect $\left[\frac{t}{T_p}\right]$ is a step signal which is shown as

$$\text{rect}\left[\frac{t}{T_p}\right]=\begin{cases}1, & \left|\frac{t}{T_p}\right|\leq 1;\\ 0, & \text{elsewise.}\end{cases} \tag{2.6}$$

The shape of $N(N=4)$ pulses with the pulse repetition frequency (PRI) is shown in Fig. 2.2. Its bandwidth, pulse width and PRI are set as 100KHz, 50μs and 0.1ms, respectively. In detail, one pulse shape in time domain and in time-frequency domain are shown in Fig. 2.3 and Fig. 2.4, respectively. In Fig. 2.3, the up figure is the real part of the signal and the bottom figure is the imaginary part. The x-axis denotes the time, and the y-axis denotes the amplitude. It can be found that the frequency increases in pulse width T_p while the signal is zero when t is not in T_p. In Fig. 2.4, x-axis denotes the time, y-axis denotes the frequency, and color denotes the amplitude. Red denotes the strong amplitude while blue denotes the weak amplitude. It is obvious to see that the frequency whose color is red increases with the time in T_p, linearly. Such a change in frequency can be used to characterize signals at different points in time during one T_p. Therefore, the time resolution can be improved.

The improved resolution is shown in the ambiguity function of the LFM signals. Substituting Eq. (2.5) into Eq. (2.4), the ambiguity function of the LFM signals can be written as

$$|\chi(\tau,\nu)|^2=\frac{1}{T_p}\left|\int\text{rect}\left[\frac{t}{T_p}\right]\text{rect}\left[\frac{t-\tau}{T_p}\right]e^{-j\frac{B}{T_p}\pi t^2}e^{-j\frac{B}{T_p}\pi(t-\tau)^2}e^{j2\pi\nu t}\right|^2. \tag{2.7}$$

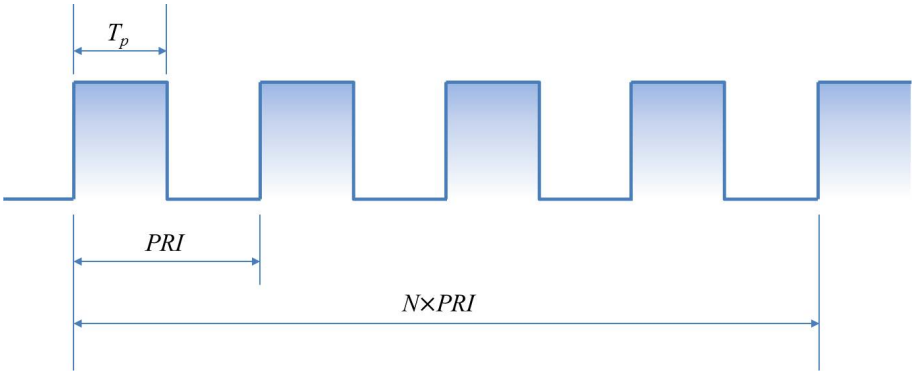


Figure 2.2: Coherent pulse (N=4)

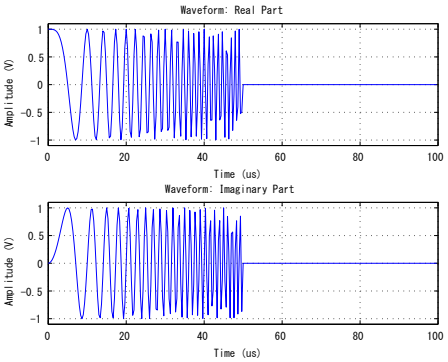


Figure 2.3: LFM signal in time domain

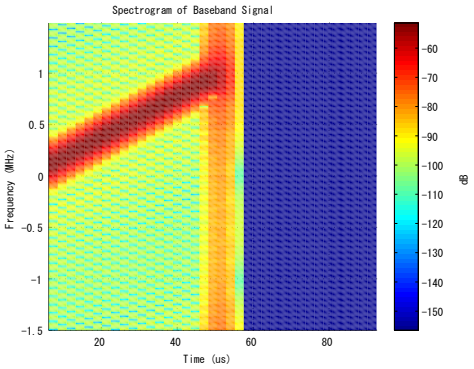


Figure 2.4: LFM signal in time-frequency domain

Finishing the integration process yields:

$$|\chi(\tau,\nu)|^2 = \left|\left(1-\frac{\tau}{T_p}\right)\frac{\sin\left((\pi B\tau+\pi T_p\nu)\left(1-\frac{\tau}{T_p}\right)\right)}{(\pi B\tau+\pi T_p\nu)\left(1-\frac{\tau}{T_p}\right)}\right|^2. \tag{2.8}$$

If only consider the delay τ, Eq. (2.8) can be simplified as

$$|\chi(\tau,0)|^2 = \left|\left(1-\frac{\tau}{T_p}\right)\frac{\sin\left((\pi B\tau)\left(1-\frac{\tau}{T_p}\right)\right)}{(\pi B\tau)\left(1-\frac{\tau}{T_p}\right)}\right|^2. \tag{2.9}$$

The first null occurs at

$$\tau \approx \frac{1}{B}. \tag{2.10}$$

In other words, the resolution of range (delay) equals to $\frac{1}{B}$, approximately. The ambiguity function of one pulse signal which is not modulated is shown as

$$|\chi(\tau,0)|^2 = \left(1-\frac{\tau}{T_p}\right)^2. \tag{2.11}$$

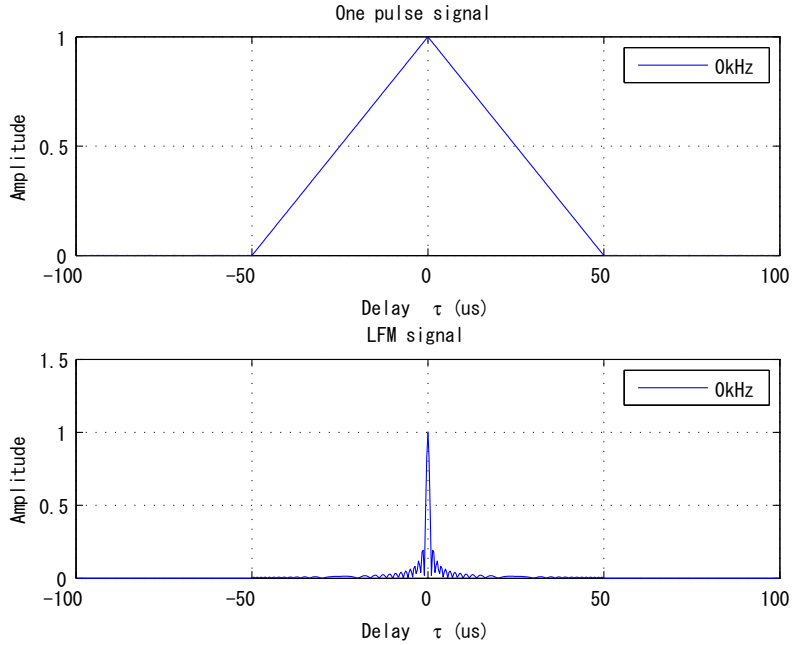


Figure 2.5: One pulse ambiguity plot and LFM ambiguity plot (delay)

It can be found that the value of the ambiguity function of the unmodulated signal is depend on the pulse width T_p, while the one of LFM is depend on the bandwidth B. In real applications, the pulse width cannot be shortened too much because a too short pulse cannot keep its energy

in the air. Thanks for a wide-band signal even UWB signal generator having been developed, increasing the bandwidth is an effect method to improve the resolution of range (delay).

In Fig. 2.5, the up figure and bottom figure show one pulse ambiguity function and LFM ambiguity function, respectively, whose pulse width is the same. It shows that LFM has a more narrow anbiguity function than one pulse.

The resolution of Doppler frequency is introduced in the following. Since the frequency of the LFM signal is changed with the time, the Doppler frequency is difficult to be achieved by only one pulse. Hence, multiple pulses are transmitted to detect the target's velocity. The different received signals have different phases due to the Doppler effect caused by the moving of targets. It should be noted that the moving distance is too small to cause the change of the delay usually, because the sample time is very short. But a large change of the phase of the echo signals corresponding to different pulses will occur when the frequency is high.

Hence pulse accumulation technology which accumulates the change of phases is used to detect the target's velocity. It is shown in Fig. 2.6. Three echo signals are reflected by one target. The phases are treated as the amplitude of the correlation. In other words, the amplitude of the correlation is largest when $f_d = 0$. Then this three pulses build up a sine function which is the dotted line. The frequency of the dotted line can be calculated by Fourier transform. The frequency of the sine function and the velocity of target is one to one correspondence.

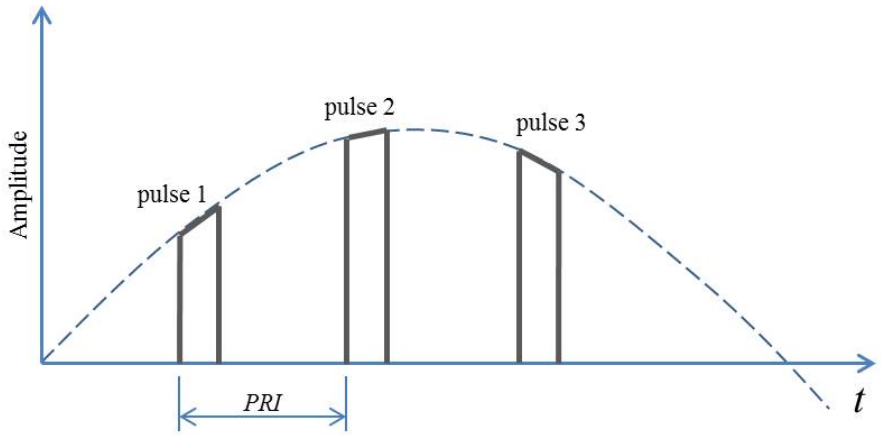


Figure 2.6: Pulse accumulation with 4 pulses

2.1.4 Echo signal model

Consider the radar transmits N pulses to detect K targets in the search space. This procedure is similar to the problem of identification of underspread linear systems (ULSs) [63]. The received

signal has three important parameters to be solved: the amplitude α_k, which is related to the radar cross section (RCS) of kth target; the delay τ_k, which is related to the range of kth target; and the Doppler frequency ν_k, which is related to the velocity of kth target. It can be shown as:

$$u(m,n) = \sum_{n=1}^{N}\sum_{k=1}^{K} \alpha_k x_T\left(t_m - \tau_k\right) e^{-j2\pi\nu_k n T_{\mathrm{PRI}}}, \tag{2.12}$$

where T_{PRI} is the PRI, $m = 1, ...M$, M is the number of the sampled data of the echo corresponding to one pulse. Then the estimated result of the three important parameters is shown in the delay-Doppler map with the $M \times N$ grid. The grid is decided by the resolution of delay and Doppler frequency. The delay-Doppler map is shown in Fig. 2.7. The x-axis denotes the Doppler frequency domain and the y-axis denotes the delay domain. The value in the grid denotes the $\alpha_{m,n}$. Ideally, if there is a target with τ_m and ν_n, $\alpha_{m,n} \neq 0$, otherwise $\alpha_{m,n} = 0$. To achieve the delay-Doppler map, the received data is arrayed as the radar cube which is shown in Fig. 2.8. The received data from one pulse is arrayed as one column, and each of these row vectors contains received data from different pulses corresponding to the same delay bin.

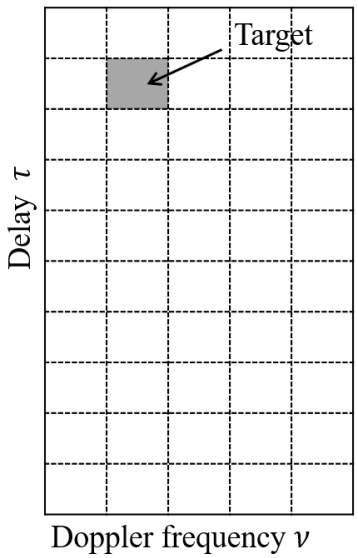


Figure 2.7: Delay-Doppler map

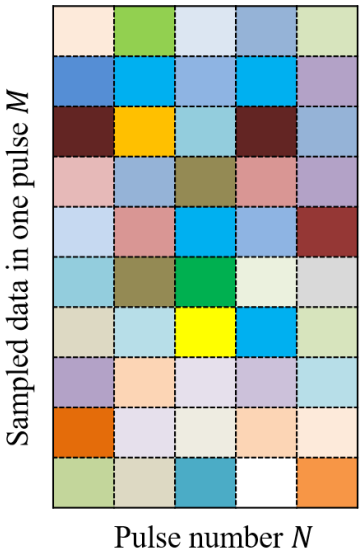


Figure 2.8: Two-dimensional radar data cube

Based on the radar data cube, i.e., the matrix representation of received data, the sampled data model can be simply written in the matrix form as

$$\mathbf{U} = \mathbf{\Psi S E}^H, \tag{2.13}$$

where

$$\mathbf{U} = \begin{bmatrix} u(1,1) & \dots & u(1,N) \\ \vdots & \ddots & \vdots \\ u(M,1) & \cdots & u(M,N) \end{bmatrix}, \tag{2.14}$$

$$\mathbf{S} = \begin{bmatrix} \alpha_{1,1} & \dots & \alpha_{1,N} \\ \vdots & \ddots & \vdots \\ \alpha_{M,1} & \cdots & \alpha_{M,N} \end{bmatrix}, \tag{2.15}$$

denotes the delay-Doppler map, $\alpha_{m,n}$ corresponds the value of the grid in the mth row and the nth column,

$$\mathbf{\Psi} = \begin{bmatrix} x_T(t_1,\tau_1) & \dots & x_T(t_1,\tau_M) \\ \vdots & \ddots & \vdots \\ x_T(t_M,\tau_1) & \cdots & x_T(t_M,\tau_M) \end{bmatrix}, \tag{2.16}$$

is the dictionary matrix including all possible delay, and

$$\mathbf{E}^H = \begin{bmatrix} e^{-j2\pi\nu_1 T_{\mathrm{RIP}}} & \dots & e^{-j2\pi\nu_1 NT_{\mathrm{RIP}}} \\ \vdots & \ddots & \vdots \\ e^{-j2\pi\nu_N T_{\mathrm{RIP}}} & \dots & e^{-j2\pi\nu_N NT_{\mathrm{RIP}}} \end{bmatrix} \tag{2.17}$$

is the twiddle factor, which including all possible Doppler frequency. Once the matrix $\mathbf{S}$ is solved from $\mathbf{U}$, the delay τ and the Doppler frequency ν of targets can be known, since the nonzero $\alpha_{m,n}$ means that there is a target with τ_m and ν_n.

2.2 Random demodulation

The Nyquist sampling theory indicates the lowest sampling rate that a discrete sampled data to reconstruct all the information from a continuous-time signal of finite bandwidth. According to the Nyquist sampling theory, if an analog signal $x(t)$ has the max frequency BHz, it is completely determined by giving its ordinates at a series of points spaced 1/(2B) seconds apart. In other words, the signal should be sampled at $f_s > 2B$ at least for perfect reconstruction. Otherwise, aliasing occurs as Fig. 2.9. The red curve is the real analog signal. The black dot is the data sampled by a lower rate than Nyquist sampling rate. Thus, the reconstructed signal is shown as the black dotted line. Obviously, it does not coincide with the red curve.

In this section, the RD method without Nyquist sampling rate is introduced. It is consisted of a mixer, an integrator, a pseudo-random Bernoulli signal generator and a low-rate ADC. The sampled data $y(l)$ is created by the RD, Then the sparse information s is extracted from $y(l)$ by the digital signal processing (DSP), where $l = 1, ..., L$. The block diagram of the RD is shown in Fig. 2.10.

Before sampling by a low-rate ADC, the front-end of the RD has two main steps: demodulation and integration (compression). First, an unknown continuous signal $x(t)$ is demodulated by a mixer with a white noise-like pseudo-random Bernoulli signal $p_n(t) \in \pm 1$, which must be faster than Nyquist rate of $x(t)$. It avoids destroying the important information of $x(t)$ by the integrator in the next step. Thus the demodulated signal can be written as:

$$x_p(t) = x(t) \cdot p_n(t). \tag{2.18}$$

Then $x_p(t)$ is compressed by an integrator in integration part. The integrator, which can be performed in the continuous-time in the discrete-time domain, makes one output pulse value to be the integral of its input signal in $\mathcal{L}$ time. At last, the output signal is sampled by a low-rate ADC at the sub-Nyquist rate $\mathcal{L}$. The sampled data can be written as follows:

$$\begin{aligned} y(l) &= \int_{-\infty}^{\infty} x_p(t_\tau) h(t - t_\tau) dt_\tau + w(l) \\ &= \int_{-\infty}^{\infty} x(t_\tau) p_n(t_\tau) h(t - t_\tau) dt_\tau + w(l), \end{aligned} \tag{2.19}$$

where $t = l\mathcal{L}$, $w[l]$ denotes the white Gaussian noise, and $h(t_\tau)$ is a step signal which is shown as follows:

$$h(t_\tau) = \begin{cases} 1, & (l-1)\mathcal{L} < t_\tau \leq l\mathcal{L}; \\ 0, & \text{elseothers.} \end{cases} \tag{2.20}$$

According to Nyquist-Shannon sampling theorem, the continuous signal $x(t)$ can be represented by discrete finite elements $\boldsymbol{x}(m) = [x(1), x(2), ..., x(M)]^T$, where N is the number of Nyquist samples which relates to the bandwidth. Similarly, $\boldsymbol{p}_l(m) = [p_l(1), p_l(2), ..., p_l(M)]^T$. Then we achieve the matrix representation of the demodulated signal which is shown as follows:

$$\boldsymbol{x}_p = \mathbf{P}\boldsymbol{x}, \tag{2.21}$$

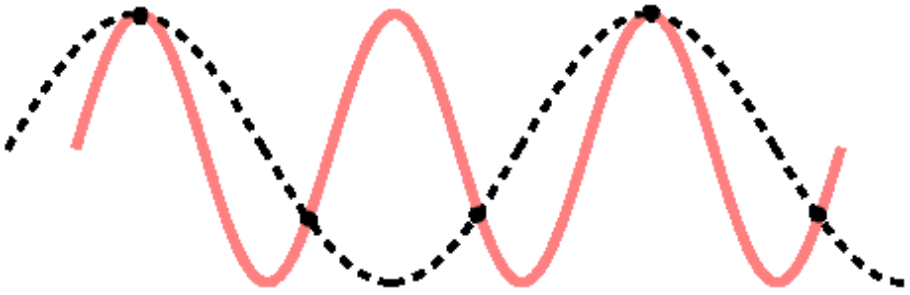


Figure 2.9: Aliasing occurs when sampling rate is too slow

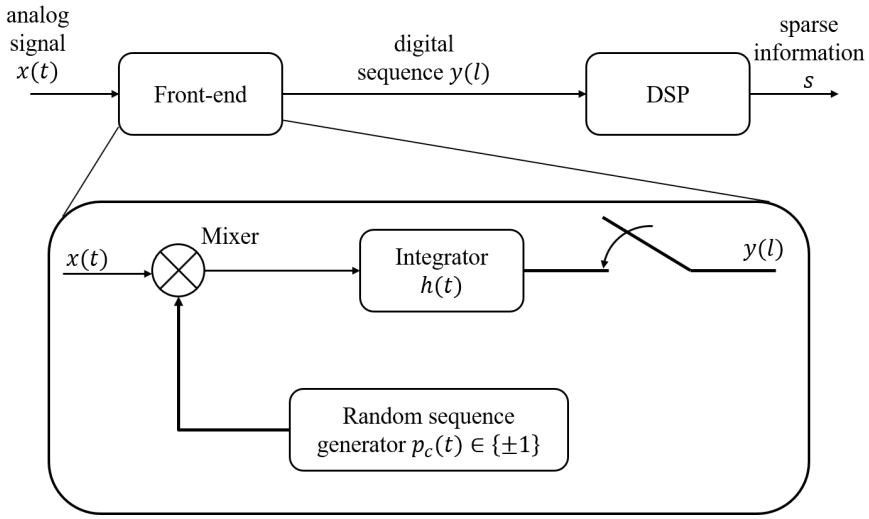


Figure 2.10: The block diagram of RD

where $\mathbf{P} = \text{diag}\,(\boldsymbol{p}_l(m))$. Then the sampled data $y(l)$ can be written as follows

$$y(l) = \sum_{t=(l-1)\mathcal{L}+1}^{m\mathcal{L}} p_l(t)x(t) + w(l). \tag{2.22}$$

The above equation can be written as the matrix representation which is written as follows:

$$\boldsymbol{y}_{sa} = \mathbf{HP}\boldsymbol{x} + \boldsymbol{w}, \tag{2.23}$$

where $\boldsymbol{y}_{sa} = [y(1), y(2), ..., y(L)]$, $\boldsymbol{w}$ denotes the noise vector, and the integrator matrix $\mathbf{H}$ can be shown as follows:

$$\mathbf{H} = \begin{bmatrix} \overbrace{1...1}^{\mathcal{L}} & & & \\ & \overbrace{1...1}^{\mathcal{L}} & & \\ & & \ddots & \\ & & & \overbrace{1...1,}^{\mathcal{L}} \end{bmatrix} \tag{2.24}$$

where $\mathcal{L} = M/L$. The discrete sampled data $\boldsymbol{y}_{sa}$ is characterized as a linear transformation of the vector $\boldsymbol{x}$, thus, $\boldsymbol{y}_{sa}$ keeps all information of $\boldsymbol{x}$.

In the UWB radar systems, the received signal is sampled by the RD whose demodulation signal $p_l(m)$ is the same in each pulse, thus it can be written as follows:

$$\mathbf{X} = \mathbf{HPU} + \mathbf{W} = \mathbf{HP\Psi SE}^H + \mathbf{W}. \tag{2.25}$$

The goal is estimating $\mathbf{S} \in \mathbb{C}^{M\times N}$ from $\mathbf{X} \in \mathbb{C}^{L\times N}$ which is buried in white Gaussian noise

$\mathbf{W} \in \mathbb{C}^{L \times N}$.

2.3 Compressive sensing

In this section, a brief of compressive sensing is given. In many practical problems of signal processing, reconstructing a signal from measured data is an important task. Assuming that the measurement method is linear, which is very common, the problem results in solving a linear system of equations. It can be shown as:

$$\mathbf{A}\mathbf{x} = \mathbf{y}, \tag{2.26}$$

where matrix $\mathbf{A} \in \mathbb{C}^{M \times N}$ models the linear measurement process, vector $\mathbf{y} \in \mathbb{C}^{M \times 1}$ denotes the measured data, ,vector $\mathbf{x} \in \mathbb{C}^{N \times 1}$ denotes the unknown signal, and $M < N$. In this equation, there are infinitely many solutions since the number of equation (M) is smaller than the number of the unknowns (N). The task of CS is to find the sparse solution from the infinitely many solutions.

2.3.1 Sparse signal

The sparse signal means that most entries are zeros in the signal. In most practical applications, the signal can be repressed as a sparse signal by some particular basis, such as Fourier transform basis, wavelet transform basis, Hartley transformation basis, etc. It explains the effectiveness of the compression techniques for JPEG, MP3, etc.

For example, the time domain and frequency representations of an audio signal "prince" spoken by a lady is shown as Fig. 2.11 and Fig. 2.12, respectively. The signal in frequency domain becomes pulse in some taps while others are close to zero. Further, Fig. 2.13 shows the same signal in time-frequency domain. The white part means that there is zero coefficient in this blog, while the dark part means that there is non-zero coefficient in this blog. It can be found that most positions are white, which means that only a few non-zero coefficient exists in the time-frequency domain. Thereby, a non-sparse signal has a sparse representation by a specific transformation, such as $\mathbf{x} = \mathbf{\Psi}\mathbf{s}$, where $\mathbf{\Psi}$ denotes the specific transformation basis, and $\mathbf{s}$ denotes the sparse signal. Hence, Eq.(2.26) can be written as follows:

$$\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{\Psi}\mathbf{s} = \mathbf{y}, \tag{2.27}$$

Then $\mathbf{A}\mathbf{\Psi}$ in Eq.(2.27) equivalents to the measurement matrix $\mathbf{A}$ in Eq.(2.26), and $\mathbf{s}$ in Eq.(2.27) equivalents to the sparse signal $\mathbf{x}$ in Eq.(2.26). Therefore, Eq.(2.27) equivalents to Eq.(2.26).

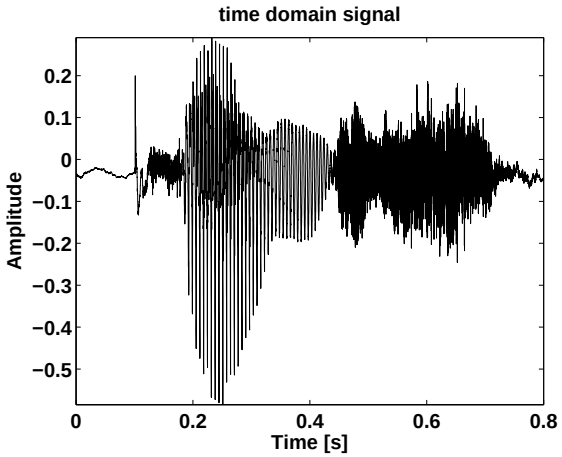


Figure 2.11: the signal of "prince" in time domain

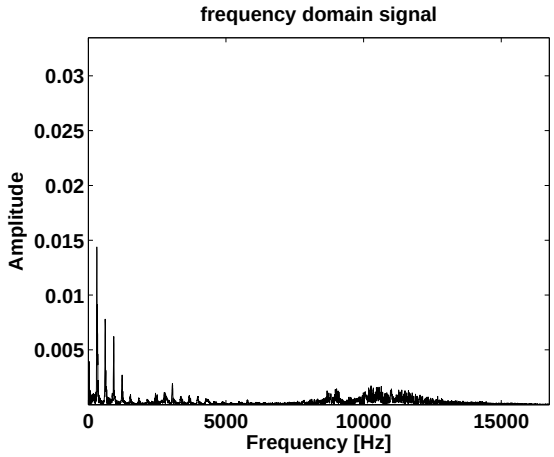


Figure 2.12: The signal of "prince" in frequency domain

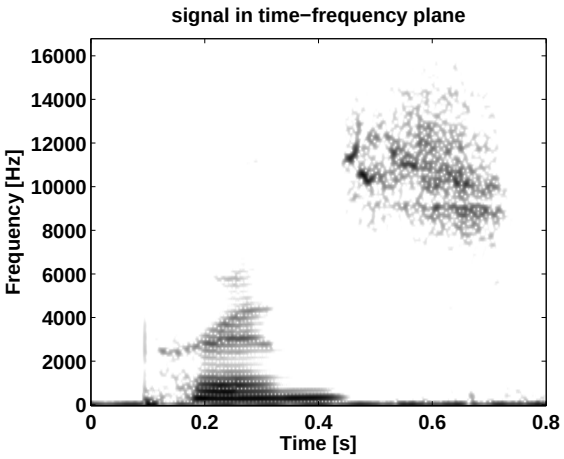


Figure 2.13: The signal of "prince" in time-frequency plane

In the sparse signal, only a few tap whose value is not zero is our concern. In radar system, the number of target is much fewer than the search space, so that it also can be considered as a spares signal. Since most zero taps can be set to zero simply, recovering all entries is not necessary. Therefore, it is easy to know that recovering a sparse signal from a large number of measured data is not effective. CS a efficient method to search the sparse solution from the reduced sampled data by solving the underdetermined linear question Eq. (2.26).

2.3.2 Sparse constraint

In CS theory, the sparse solution is searched by a sparse constraint, which relates to norm. In related areas of mathematics, a norm is used to express a strictly positive "length" or "size" to each matrix. For a vector $\boldsymbol{x} = [x_1, ..., x_L]$, the most common norms are listed as follows:

ℓ_1-norm: $\|\boldsymbol{x}\|_1 = \sum_{l=1}^{L} |x_l|$;

ℓ_2-norm: $\|\boldsymbol{x}\|_2 = \sqrt{\sum_{l=1}^{L} |x_l|^2}$;

ℓ_p-norm: $\|\boldsymbol{x}\|_p = \left[\sum_{l=1}^{L} |x_l|^p\right]^{1/p}, (0 < p < 1)$;

and ℓ_0-norm which is not actually a norm. Here, ℓ_0-norm is defined as the number of non-zero elements in a vector $\boldsymbol{x}$.

The matrix norm is a natural extension of the vector norms. For a matrix $\mathbf{x} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N]$, where $\boldsymbol{x}_n = [x_{1,n}, ..., x_{M,n}]^T$, the most common norms are often defined as follows:

ℓ_1-norm: $\|\mathbf{X}\|_1 = \sum_{m=1}^{M} \sum_{n=1}^{N} |x_{mn}|$;

ℓ_F-norm: $\|\mathbf{X}\|_F = \sqrt{\sum_{m=1}^{M} \sum_{n=1}^{N} |x_{mn}|^2}$;

ℓ_p-norm: $\|\mathbf{X}\|_p = \left[\sum_{m=1}^{M} \sum_{n=1}^{N} |x_{mn}|^p\right]^{1/p}, (0 < p < 1)$;

and ℓ_0-norm: the number of non-zero elements in a matrix $\mathbf{X}$, where ℓ_F-norm denotes Frobenius norm which is corresponding to ℓ_2-norm of vector.

The first algorithmic approach to recovery $\mathbf{x}$ from $\mathbf{y}$ is ℓ_0-minimization, which can be shown as follows:

$$\text{minnimize} \|\tilde{\mathbf{x}}\|_0 \ \text{ subject to } \mathbf{A}\tilde{\mathbf{x}} = \mathbf{y}. \tag{2.28}$$

In words, the sparse solution is searched by ℓ_0-minimizing, while satisfying $\mathbf{A}\tilde{\mathbf{x}} = \mathbf{y}$. However, ℓ_0-minimization is an NP-hard problem in general. It is difficult to be solved.

CS points that instead of ℓ_0-minimization, ℓ_1-minimization is also an effective alternative. The geometrical interpretation of different norms is shown as Fig. 2.14. In this figure, assume that $N = 2$ and $M = 1$. The red line denotes the solution set.

· In the left figure, the blue line denotes ℓ_p-norm, where $0 < p < 1$. The intersection of the solution set and ℓ_p-norm is $\tilde{x}_{\ell p}$. It is sparse.

· In the middle figure, the blue line denotes ℓ_1-norm. The intersection of the solution set and ℓ_1-norm is $\tilde{x}_{\ell 1}$. It is also sparse.

· In the right figure, the blue line denotes ℓ_2-norm. The intersection of the solution set and ℓ_2-norm is $\tilde{x}_{\ell 2}$. It is not sparse.

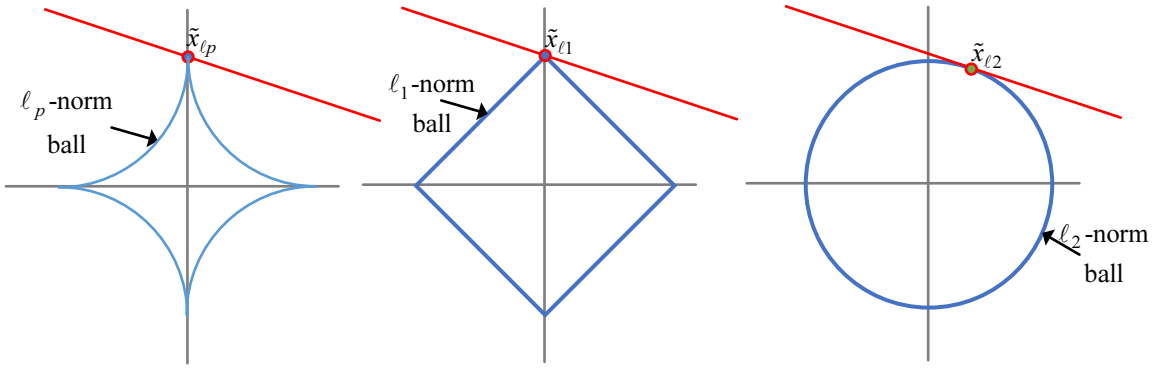


Figure 2.14: Geometrical interpretation of sparse constraint

It explains simply why ℓ_p-minimization and ℓ_1-minimization can achieve the sparse solution as the same as ℓ_0-minimization, while ℓ_2-minimization can't. Futher, ℓ_1-norm is an convex function. The following problem can be solved by convex optimization effectively:

$$\text{minnimize} \|\tilde{\mathbf{x}}\|_1 \quad \text{subject to } \mathbf{A}\tilde{\mathbf{x}} = \mathbf{y}. \tag{2.29}$$

However, in practical applications, the measured data is not slightly inaccurate due to the noise. Hence, the following problem which is the quadratically constrained ℓ_1-minimization is often solved to find the sparse solution:

$$\text{minnimize} \|\tilde{\mathbf{x}}\|_1 \quad \text{subject to } \|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{y}\|_2 \leq \eta. \tag{2.30}$$

It is easy to know that Eq.(2.26) cannot be solved when $\mathbf{A} = \mathbf{0}$ ($\mathbf{0}$ means all elements in the matrix are zero). Restricted Isometry Property (RIP) is an important property of $\mathbf{A}$ to ensure the effectiveness of CS.

Define For a matrix $\mathbf{A} \in \mathbb{C}^{M \times N}$ $(M < N)$ with ℓ_2-normalized columns. and for an integer

scalar $K \leq M$. Assume that the sub-matrices $\mathbf{A}_K$ contains K columns from $\mathbf{A}$. Define δ_K make

$$(1-\delta_K)\left\|\mathbf{x}\right\|_2^2 \leq \left\|\mathbf{A}\mathbf{x}\right\|_2^2 \leq (1+\delta_K)\left\|\mathbf{x}\right\|_2^2 \tag{2.31}$$

to hold true for any choice of s columns, then $\mathbf{A}$ is said to have an s-RIP with constant δ_K. If $0 < \delta_K < 1$, Eq.2.26 can be solved by CS.

Proof Assume that there are $\tilde{\mathbf{x}}_1$ and $\tilde{\mathbf{x}}_2$ satisfying $\mathbf{y} = \mathbf{A}\tilde{\mathbf{x}}$ at the same time. We define $\mathbf{d} = \tilde{\mathbf{x}}_1 - \tilde{\mathbf{x}}_2$. Then

$$\left\|\mathbf{A}\mathbf{d}\right\|_2^2 = \left\|\mathbf{A}\tilde{\mathbf{x}}_\mathbf{1} - \mathbf{A}\tilde{\mathbf{x}}_2\right\|_2^2 = \mathbf{0}. \tag{2.32}$$

Consinder Eq.(2.31), we can know that $\delta_K = 1$, which conflicts with $0 < \delta_K < 1$. Thereby, when $0 < \delta_K < 1$, $\tilde{\mathbf{x}}_1$ is equal to $\tilde{\mathbf{x}}_2$ certainly, namely there is only one solution for Eq.(2.26).

2.3.3 Domain transformation

Recall the sampled data model Eq.(2.25) is a 2D problem. There are delay and Doppler frequency information in matrix $\mathbf{S}$, while the basic CS model Eq.(2.26) is only consider the 1D problem. In order to solve Eq.(2.25) by CS, Eq.(2.25) should be written as the vector operation which is shown as follows:

$$\begin{aligned}
\boldsymbol{x} &= \mathrm{vec}\left(\mathbf{X}\right) \\
&= \mathrm{vec}\left(\mathbf{H}\mathbf{P}\boldsymbol{\Psi}\mathbf{S}\mathbf{E}^H + \mathbf{W}\right) \\
&= \mathrm{vec}\left(\mathbf{H}\mathbf{P}\boldsymbol{\Psi}\mathbf{S}\mathbf{E}^H\right) + \mathrm{vec}\left(\mathbf{W}\right) \\
&= \boldsymbol{\Phi}\boldsymbol{s} + \boldsymbol{w},
\end{aligned} \tag{2.33}$$

where $\boldsymbol{x} = \mathrm{vec}\left(\mathbf{X}\right)$, $\boldsymbol{s} = \mathrm{vec}\left(\mathbf{S}\right)$, and $\boldsymbol{w} = \mathrm{vec}(\mathbf{W})$, $\mathrm{vec}(\cdot)$ denotes the vectorization of a matrix by stacking the columns of the matrix into a single column vector:

$$\mathrm{vec}\left(A\right) = \left[\ a_{1,1},\ \ \cdots,\ \ a_{m,1},\ \ a_{1,2},\ \ \cdots,\ \ a_{m,2},\ \ \cdots,\ \ a_{1,n},\ \ \cdots,\ \ a_{m,n},\ \right]^T,$$

where $a_{i,j}$ represents $A(i,j)$ and the superscript $[\cdot]^T$ denotes the transpose. Therefore, some properties of 1D CS are also suitable for Eq.(2.33).

The measurement matrix $\boldsymbol{\Phi} = \mathbf{E} \otimes \mathbf{H}\mathbf{P}\boldsymbol{\Psi}$, where $\otimes$ denotes the Kronecker product. In mathematics, the Kronecker product, denoted by $\otimes$, is an operation on two matrices of arbitrary size resulting in a block matrix. It is a generalization of the outer product (which is denoted by the same symbol) from vectors to matrices and gives the matrix of the tensor product with respect to a standard choice of basis. The Kronecker product is named after Leopold Kronecker.

It has been used widely for multidimensional signal processing [64–66]. Let

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,N} \\ \vdots & \ddots & \vdots \\ a_{M,1} & \cdots & a_{M,N} \end{bmatrix}, \tag{2.34}$$

and

$$\mathbf{B} = \begin{bmatrix} b_{1,1} & \cdots & b_{1,Q} \\ \vdots & \ddots & \vdots \\ b_{P,1} & \cdots & b_{P,Q} \end{bmatrix}, \tag{2.35}$$

Thus, $\mathbf{A} \otimes \mathbf{B}$ is defined by

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{1,1}\mathbf{B} & \cdots & a_{1,N}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{M,1}\mathbf{B} & \cdots & a_{M,N}\mathbf{B} \end{bmatrix}, \tag{2.36}$$

more explicitly:

$$\mathbf{A}\otimes\mathbf{B} = \begin{bmatrix} a_{1,1}b_{1,1} & a_{1,1}b_{1,2} & \cdots & a_{1,1}b_{1,Q} & \cdots & \cdots & a_{1,N}b_{1,1} & a_{1,N}b_{1,2} & \cdots & a_{1,N}b_{1,Q} \\ a_{1,1}b_{2,1} & a_{2,1}b_{2,2} & \cdots & a_{1,1}b_{2,Q} & \cdots & \cdots & a_{1,N}b_{2,1} & a_{1,N}b_{2,2} & \cdots & a_{1,N}b_{2,Q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{1,1}b_{P,1} & a_{1,1}b_{P,2} & \cdots & a_{1,1}b_{P,Q} & \cdots & \cdots & a_{1,N}b_{P,1} & a_{1,N}b_{P,2} & \cdots & a_{1,N}b_{P,Q} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ a_{M,1}b_{1,1} & a_{M,1}b_{1,2} & \cdots & a_{M,1}b_{1,Q} & \cdots & \cdots & a_{M,N}b_{1,1} & a_{M,N}b_{1,2} & \cdots & a_{M,N}b_{1,Q} \\ a_{M,1}b_{2,1} & a_{M,1}b_{2,2} & \cdots & a_{M,1}b_{2,Q} & \cdots & \cdots & a_{M,N}b_{2,1} & a_{M,N}b_{2,2} & \cdots & a_{M,N}b_{2,Q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{M,1}b_{P,1} & a_{M,1}b_{P,2} & \cdots & a_{M,1}b_{P,Q} & \cdots & \cdots & a_{M,N}b_{P,1} & a_{M,N}b_{P,2} & \cdots & a_{M,N}b_{P,Q} \end{bmatrix}. \tag{2.37}$$

The Kronecker product will increase the memory requirement to save $\mathbf{\Phi}$.

Chapter 3

2D-CS-based algorithms for delay Doppler joint estimation

3.1 Matrix operation for CS

Although there are a lot of CS-based algorithms for 1D problem, the 2D problem cannot be solved by these traditional CS-based algorithms directly. Therefore, in order to apply CS-based algorithms, Eq. (2.25) should be reformed a 1D problem by Eq. (2.33) first. This dimensional transform is easy to result in a new measurement matrix $\mathbf{\Phi} = \mathbf{E} \otimes \mathbf{HP\Psi}$ with a large size, which requires a huge storage space and high complexity. For example, for the data $\mathbf{X} \in \mathbb{C}^{30\times10}$ on matrix $\mathbf{HP\Psi} \in \mathbb{C}^{30\times100}$ and $\mathbf{E}^H \in \mathbb{C}^{10\times10}$, it leads to the measurement matrix $\mathbf{\Phi} \in \mathbb{C}^{3000\times100}$. The unnecessary data is a heavy burden for storage, transmission and processing. In order to avoid the dimensional transform operation, 2D-CS-based algorithms are proposed to estimate the delay-Doppler map in Eq.(2.25) by solving the following matrix optimization problem:

$$\mathbf{S} = \min \left\| \mathbf{X} - \mathbf{HP\Psi SE}^H \right\|_F^2 + \lambda \|\mathbf{S}\|_0, \tag{3.1}$$

It is obvious that the ℓ_F-norm is to guarantee $\mathbf{X} = \mathbf{HP\Psi SE}^H$ (data-fitting term), while sparse constraint ℓ_0-norm is guarantee the sparsity of the solution (sparse term). Let $\mathbf{G}(\mathbf{S}) = \lambda\|\mathbf{S}\|_0$, and $\mathbf{F}(\mathbf{S}) = \left\|\mathbf{X} - \mathbf{HP\Psi E}^H\right\|_F^2$. Eq.(3.1) can be simply rewritten as $\mathbf{S} = \min \mathbf{G}(\mathbf{S}) + \mathbf{F}(\mathbf{S})$. Firstly, we will focus on the problem of the data-fitting term: $\min \mathbf{F}(\mathbf{S})$.

In 1D CS-based algorithms, the linear least square (LS) method is used to find the LS solution of the data-fitting term in Eq.(2.33): $\boldsymbol{x} = \mathbf{\Phi}\boldsymbol{s} + \boldsymbol{w}$. The solution is searched by: $\boldsymbol{s}(i+1) = \boldsymbol{s}(i) + \mathbf{\Phi}^H(\boldsymbol{x} - \mathbf{\Phi}\boldsymbol{s}(i))$, where $\boldsymbol{s}(i)$ denotes the solution in ith iteration. At last, each element in $\boldsymbol{s}(i)$ is rearranged as the delay-Doppler map. In our proposed algorithm, the LS

method is modified as follows:

$$\begin{aligned}\mathbf{S}(i+1) &= \mathbf{S}(i) + \nabla \mathbf{F}\left(\mathbf{S}\left(i\right)\right) \\ &= \mathbf{S}(i) + (\mathbf{HP\Psi})^H \mathbf{RE},\end{aligned} \tag{3.2}$$

where $\mathbf{R} = \mathbf{X} - \mathbf{HP\Psi S}(i)\mathbf{E}^H$ denotes the residual at each iteration.

Theorem 1. Let $\mathbf{S}(i) = \mathbf{S}(i-1) + (\mathbf{HP\Psi})^H \mathbf{RE}$ and $\boldsymbol{s}(i) = \boldsymbol{s}(i-1) + \boldsymbol{\Phi}^H \boldsymbol{r}$, where $\boldsymbol{r} = (\boldsymbol{x} - \boldsymbol{\Phi s}(i-1))$. Then for any $i > 0$,

$$\boldsymbol{s}(i) = \mathrm{vec}\left(\mathbf{S}(i)\right). \tag{3.3}$$

Proof: Applying vec$(\cdot)$ on both sides of $\mathbf{S}(i) = \mathbf{S}(i-1) + (\mathbf{HP\Psi})^H \mathbf{RE}$, we have

$$\begin{aligned}\mathrm{vec}\left(\mathbf{S}(i)\right) &= \mathrm{vec}\left(\mathbf{S}(i-1) + (\mathbf{HP\Psi})^H \mathbf{RE}\right) \\ &= \mathrm{vec}\left(\mathbf{S}(i-1)\right) + \mathrm{vec}\left((\mathbf{HP\Psi})^H \mathbf{RE}\right)\end{aligned} \tag{3.4}$$

According to [67], we have

$$\begin{aligned}\mathrm{vec}\left((\mathbf{HP\Psi})^H \mathbf{RE}\right) &= \left(\mathbf{E} \otimes (\mathbf{HP\Psi})\right) \mathrm{vec}\left(\mathbf{R}\right) \\ &= \boldsymbol{\Phi}^H \mathrm{vec}\left(\mathbf{X} - \mathbf{HP\Psi S}(i)\mathbf{E}^H\right) \\ &= \boldsymbol{\Phi}^H \left[\mathrm{vec}\left(\mathbf{X}\right) - \mathrm{vec}\left(\mathbf{HP\Psi S}(i-1)\mathbf{E}^H\right)\right] \\ &= \boldsymbol{\Phi}^H \left[\boldsymbol{x} - \mathbf{E} \otimes (\mathbf{HP\Psi}) \mathrm{vec}\left(\mathbf{S}(i-1)\right)\right] \\ &= \boldsymbol{\Phi}^H \left[\boldsymbol{x} - \boldsymbol{\Phi} \mathrm{vec}\left(\mathbf{S}(i-1)\right)\right]\end{aligned} \tag{3.5}$$

Using Eq. (3.5) into Eq. (3.4), it follows that

$$\begin{aligned}\mathrm{vec}\left(\mathbf{S}(i)\right) &= \mathrm{vec}\left(\mathbf{S}(i-1)\right) + \boldsymbol{\Phi}^H \left[\boldsymbol{x} - \boldsymbol{\Phi} \mathrm{vec}\left(\mathbf{S}(i-1)\right)\right] \\ &= \left(1 - \boldsymbol{\Phi}^H \boldsymbol{\Phi}\right) \mathrm{vec}\left(\mathbf{S}(i-1)\right) + \boldsymbol{\Phi}^H \boldsymbol{x}\end{aligned} \tag{3.6}$$

Let $\mathrm{vec}\left(\mathbf{S}(i)\right) = \boldsymbol{c}(i)$, $\left(1 - \boldsymbol{\Phi}^H \boldsymbol{\Phi}\right) = \boldsymbol{a}$, and $\boldsymbol{\Phi}^H \boldsymbol{x} = \boldsymbol{b}$, Eq. (3.6) can be simplified as follows:

$$\boldsymbol{c}(i) = \boldsymbol{a}\boldsymbol{c}(i-1) + \boldsymbol{b}. \tag{3.7}$$

Then the difference equation can be rewritten as follows:

$$\boldsymbol{c}(i) = \boldsymbol{a}^{i+1} \boldsymbol{c}(0) + \boldsymbol{b} \sum_{n=0}^{i} \boldsymbol{a}^n \tag{3.8}$$

By the similar derivation, $\boldsymbol{s}(i) = \boldsymbol{s}(i-1) + \boldsymbol{\Phi}^H(\boldsymbol{x} - \boldsymbol{\Phi}\boldsymbol{s}(i-1))$ can be simplified as follow:

$$\boldsymbol{s}(i) = \boldsymbol{a}^{i+1}\boldsymbol{s}(0) + \boldsymbol{b}\sum_{n=0}^{i}\boldsymbol{a}^n \tag{3.9}$$

Due to $\boldsymbol{c}(0) = \boldsymbol{s}(0)$, $\mathrm{vec}(\mathbf{S}(i)) = \boldsymbol{s}(i)$ can be achieved.

Eq. (3.2) solves the ℓ_F-minimization. However, due to the lack of the sampled data, the sparse solution cannot be searched by the projection method directly. Hence, $\min \mathbf{G}(\mathbf{S})$ is then used to attract the solution to be sparse. In the next section, four algorithms with different approximate functions $\min \mathbf{G}(\mathbf{S})$ are proposed, i.e., 2D zero-attractive projection (2D-ZAP) algorithm, 2D iterate hard thresholding (2D-IHT) algorithm, 2D iterative shrinkage-thresholding algorithm (2D-ISTA) and 2D fast iterative shrinkage-thresholding algorithm (2D-FISTA) to search a sparse solution in Eq. (2.13), directly.

3.2 The proposed 2D-CS-based algorithms

3.2.1 2D zero-attractive projection (2D-ZAP) algorithm

Inspired by the traditional 1D-ZAP algorithm [41], the 2D-ZAP algorithm is proposed to estimate the delay-Doppler map by solving Eq.(3.1).

Since ℓ_0-minimization problem is NP hard, an approximate ℓ_0-norm is used instead of ℓ_0-norm in the proposed 2D-ZAP algorithm, which can be written as follows:

$$\|\mathbf{S}\|_0 \approx \sum_{n=1}^{N}\sum_{m=1}^{M}\left(1 - e^{-\sigma|s_{nm}|}\right), \tag{3.10}$$

where s_{nm} is the element in $\mathbf{S}$. The two sides of Eq. (3.10) is strictly equal when parameter $\sigma \to \infty$. Thus, Eq. (3.1) can be rewritten as follows:

$$\mathbf{S} = \min \underbrace{\left\|\mathbf{X} - \mathbf{HP\Psi SE}^H\right\|_F^2}_{\ell_F-\text{norm}} + \lambda \underbrace{\sum_{n=1}^{N}\sum_{m=1}^{M}\left(1 - e^{-\sigma|s_{nm}|}\right)}_{\text{optimization } \ell_0-\text{norm}}. \tag{3.11}$$

The optimization ℓ_0-norm term is $\mathbf{G}(\mathbf{S})$ which can be solved the gradient descent method.

In gradient descent method, the gradient (derivative) $\nabla\mathbf{G}(\mathbf{S})$ of the function $\mathbf{G}(\mathbf{S})$ at the

current point is calculated as follows:

$$\begin{aligned} g(s_{nm}) &= \nabla G\left(s_{nm}\right) \\ &= \frac{\partial 1 - e^{-\sigma|s_{nm}|}}{\partial s_{nm}} \\ &= \sigma \cdot \operatorname{sgn}(s_{nm}) e^{-\sigma|s_{nm}|}, \end{aligned} \tag{3.12}$$

where $\operatorname{sgn}(x)$ is a sign function defined as follows:

$$\operatorname{sgn}(x) = \begin{cases} 1, & x > 0; \\ -1, & x < 0; \\ 0, & x = 0. \end{cases} \tag{3.13}$$

To further reduce the complexity, the first two terms of Taylor series expansion of exponential function is used:

$$e^{-\sigma|s_{nm}|} = \sum_{q=0}^{\infty} \frac{\left(-\sigma\left|s_{nm}\right|\right)^q}{q!} \approx 1 - \sigma\left|s_{nm}\right|. \tag{3.14}$$

Then, a new point is updated along the negative of the gradient descent

$$\begin{aligned} g(s_{nm}) &= -\nabla G\left(s_{nm}\right) \\ &= \begin{cases} \sigma\left(\sigma s_{nm} + 1\right), & \frac{1}{\sigma} \leq s_{nm} < 0; \\ \sigma\left(\sigma s_{nm} - 1\right), & 0 < s_{nm} \leq \frac{1}{\sigma}; \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{3.15}$$

Finally, a local minimum of the function $\mathbf{F}(\mathbf{S})$ is achieved when the derivative of the function $\mathbf{F}(\mathbf{S}) = \left\|\mathbf{X} - \mathbf{HP\Psi SE}^H\right\|_F^2$ is close to 0.

Compared with the 1D-ZAP algorithm, the proposed 2D-ZAP algorithm reduces the memory requirement with the same estimation accuracy. Assuming that $\mathbf{HP\Psi} \in \mathbb{C}^{N_1 \times N_2}$ and $\mathbf{E} \in \mathbb{C}^{M_1 \times M_2}$. 1D-ZAP algorithm requires $N_1N_2 \times M_1M_2$ to store measurement matrix $\mathbf{\Phi}$, while 2D-ZAP algorithm only requires $N_1N_2 + M_1M_2$. In addition, in the 1D-ZAP algorithm, the complexity of the conjugate transpose $\mathbf{\Phi}_v$ is $\mathcal{O}\left(N_1N_2 \times M_1M_2\right)$, while in the 2D-ZAP algorithm, the conjugate transpose $\mathbf{E}$ and $\mathbf{HP\Psi}$ are required, that the complexity is only $\mathcal{O}\left(N_1N_2 + M_1M_2\right)$. The proposed 2D-ZAP algorithm reduces the memory requirement for measurement matrix and the complexity of the conjugate transpose, which is valuable in the pulse Doppler radar system. The proposed 2D-ZAP algorithm is listed in Algorithm 1.

3.2.2 2D iterate hard thresholding (2D-IHT) algorithm

Inspired by the traditional 1D-IHT algorithm [40], the 2D-IHT algorithm is proposed to estimate the delay-Doppler map by solving Eq.(3.1) as same as the 2D-ZAP algorithm. To further reduce

Algorithm 1 2D zero-attractive projection (2D-ZAP) algorithm

Input: measurement matrices $\mathbf{H}, \mathbf{P}, \mathbf{\Psi}, \mathbf{E}^H$, sampled signal $\mathbf{X}$, parameter λ, Maximum number of iterations i_{max},
Initialization: $\mathbf{S}^0 = \mathbf{0}$, $i = 0$, $\mathbf{R} = \mathbf{X}$,
Iteration:
$\mathbf{S}(i+1) \leftarrow \mathbf{S}(i) + \mathbf{HP\Psi}^H\mathbf{R}\boldsymbol{E}$,
$\boldsymbol{s}_{nm}(i+1) \leftarrow \boldsymbol{s}_{nm}(i+1) - g(\boldsymbol{s}_{nm}(i+1))$, for $n = 1, ..., N$, $m = 1, ..., M$.
$\mathbf{R} \leftarrow \mathbf{X} - \mathbf{HP\Psi S}(i+1)\mathbf{E}^H$,
$i \leftarrow i + 1$,
Until $i = i_{max}$,
Output: estimated delay Doppler map $\mathbf{S}^{\#} = \mathbf{S}(i)$.

the complexity of $\mathbf{G}(\mathbf{S})$, the above optimization problem can be solved as follows:

$$\mathbf{S}(i+1) = \mathbb{H}_\sigma \left[\mathbf{S}(i) + (\mathbf{HP\Psi})^H\mathbf{RE}\right], \tag{3.16}$$

where $\mathbb{H}_\sigma[s]$ is the nonlinear operation that sets s as zero when s is smaller than the threshold value σ while keeps s when s is larger or equal to σ, which is shown as follows:

$$\mathbb{H}_\sigma[s] = \begin{cases} s, & s \geq \sigma; \\ 0, & s < \sigma. \end{cases} \tag{3.17}$$

Another well-known nonlinear operator $\mathbb{H}^K[\mathbf{S}]$ is that keeps K largest absolute elements in $\mathbf{S}$ and sets the other elements to zero. It can also be used when the target number K is known. Otherwise, a too large K will lead to the high false alarm probability while a too small K will lead to the low detection probability. However, it is difficult to know the target number before the detection. Hence it can not be used for the pulse Doppler radar directly.

The 2D-IHT algorithm also reduces the memory requirement for the measurement matrix and the complexity of the conjugate transpose, which is similar to the 2D-ZAP algorithm. The proposed 2D-IHT algorithm is listed in Algorithm 2.

Algorithm 2 2D iterate hard thresholding (2D-IHT) algorithm

Input: measurement matrices $\mathbf{H}, \mathbf{P}, \mathbf{\Psi}, \mathbf{E}^H$, sampled signal $\mathbf{X}$, threshold value σ, Maximum number of iterations i_{max},
Initialization: $\mathbf{S}^0 = \mathbf{0}$, $i = 0$, $\mathbf{R} = \mathbf{X}$,
Iteration:
$\mathbf{S}(i+1) \leftarrow \mathbf{S}(i) + (\mathbf{HP\Psi})^H\mathbf{RE}$,
$\mathbf{S}(i+1) \leftarrow \mathbb{H}_\sigma[\mathbf{S}(i+1)]$,
$\mathbf{R} \leftarrow \mathbf{X} - \mathbf{HP\Psi S}(i+1)\mathbf{E}^H$,
$i \leftarrow i + 1$,
Until $i = i_{max}$,
Output: estimated delay Doppler map $\mathbf{S}^{\#} = \mathbf{S}(i)$.

3.2.3 2D iterative shrinkage-thresholding algorithm (2D-ISTA)

Inspired by the traditional 1D-ISTA algorithm [68], the 2D-ISTA algorithm is proposed to estimate the delay-Doppler map by solving Eq.(3.1) as same as the 2D-IHT algorithm. In the ISTA, $\mathbb{H}_\sigma[s]$ is replaced by the soft thresholding operator $\mathbb{S}_\sigma[s]$ which is defined as:

$$\mathbb{S}_\sigma[s] = \begin{cases} \operatorname{sgn}(s) \cdot (|s| - \sigma), & |s| > \sigma; \\ 0, & |s| \leq \sigma. \end{cases} \tag{3.18}$$

where σ is a small threshold value. Furthermore, the optimization problem can be solved as follows:

$$\mathbf{S}(i+1) = \mathbb{S}_\sigma\left[\mathbf{S}(i) + \frac{1}{L}(\mathbf{HP\Psi})^H \mathbf{RE}\right], \tag{3.19}$$

where $\frac{1}{L}$ plays the role of a step-size to control the convergence rate. When $\frac{1}{L} \to 0$, Eq.(3.1) can be solved accurately, while the convergence rate is very slow. In contrast, if a too large $\frac{1}{L}$ is chosen, overshoot will occur. It is well known that $\frac{1}{L}$ depends on the eigenvalues of $\mathbf{HP\Psi}$ and $\mathbf{E}$. For the 1D-ISTA algorithm, $L = 2\lambda_{max}\left(\mathbf{\Phi}^H\mathbf{\Phi}\right)$, where $\lambda_{max}(\cdot)$ denotes the max eigenvalues of the matrix. In the 2D-ISTA algorithm, $L = 2\lambda_{max}\left(\mathbf{\Phi}^H\mathbf{\Phi}\right)$ is replaced by $L = 2\lambda_{max}\left(\mathbf{E}^H\mathbf{E}\right) \times \lambda_{max}\left(\mathbf{HP\Psi\Psi}^H\mathbf{P}^H\mathbf{H}^H\right)$. Thus, instead of the large matrix $\mathbf{\Phi}^H\mathbf{\Phi}$, the max eigenvalue of two small matrices $\mathbf{E}^H\mathbf{E}$ and $\left(\mathbf{HP\Psi\Psi}^H\mathbf{P}^H\mathbf{H}^H\right)$ are calculated. As a result, the unnecessary complexity can be avoided. The proposed 2D-ISTA is listed in Algorithm 3.

Algorithm 3 2D iterative shrinkage-thresholding algorithm (2D-ISTA)

Input: measurement matrices $\mathbf{H}, \mathbf{P}, \mathbf{\Psi}, \mathbf{E}^H$, sampled signal $\mathbf{X}$, parameter σ, Maximum number of iterations i_{max}
Initialization: $\mathbf{S}^0 = \mathbf{0}$, $i = 0$, $\mathbf{R} = \mathbf{X}$, $L = 2\lambda_{max}\left(\mathbf{E}^H\mathbf{E}\right) \times \lambda_{max}\left(\mathbf{HP\Psi\Psi}^H\mathbf{P}^H\mathbf{H}^H\right)$,
Iteration:
$\mathbf{S}(i+1) \leftarrow \mathbf{S}(i) + \frac{1}{L}(\mathbf{HP\Psi})^H\mathbf{RE}$,
$\mathbf{S}(i+1) \leftarrow \mathbb{S}_\sigma[\mathbf{S}(i+1)]$,
$\mathbf{R} \leftarrow \mathbf{X} - \mathbf{HP\Psi S}(i+1)\mathbf{E}^H$,
$i \leftarrow i + 1$,
Until $i = i_{max}$,
Output: estimated delay Doppler map $\mathbf{S}^\# = \mathbf{S}(i)$.

3.2.4 2D fast iterative shrinkage-thresholding algorithm (2D-FISTA)

The ISTA algorithm has slow convergence in general. Therefore, some acceleration techniques have been proposed, such as an adaptive threshold value $\sigma(i)$, or an adaptive step-size $t(i)$ instead of $\frac{1}{L}$ [69, 70]. One of the most popular algorithms is the fast iterative shrinkage-thresholding algorithm (FISTA) [71]. Inspired by the 1D-FISTA algorithm, the 2D-FISTA algorithm is proposed in this paper to reduce the memory requirement and complexity. According to Theorem

1, the results of each step utilized the 2D-ISTA and 2D-FISTA algorithms are as the same as the 1D-ISTA and 1D-FISTA algorithms, respectively. Hence, similar to the relationship between the 1D-ISTA and 1D-FISTA algorithms, the 2D-ISTA and 2D-FISTA algorithms improve the worst-case complexity result of $\mathcal{O}\left(1/i^2\right)$ from 2D-ISTA complexity result of $\mathcal{O}\left(1/i\right)$. The proposed 2D-FISTA is listed in Algorithm 4.

Algorithm 4 2D fast iterative shrinkage-thresholding algorithm (2D-FISTA)

Input: measurement matrices $\mathbf{H}, \mathbf{P}, \mathbf{\Psi}, \mathbf{E}^H$, sampled signal $\mathbf{X}$, parameter σ, Maximum number of iterations i_{max}

Initialization: $\mathbf{S}^0 = \mathbf{0}$, $i = 0$, $\mathbf{R} = \mathbf{X}$, $L = 2\lambda_{max}\left(\mathbf{E}^H\mathbf{E}\right) \times \lambda_{max}\left(\mathbf{HP\Psi\Psi}^H\mathbf{P}^H\mathbf{H}^H\right)$, $t(1) = 1$.

Iteration:

$\mathbf{S}(i+1) \leftarrow \mathbf{S}(i) + \frac{1}{L}(\mathbf{HP\Psi})^H\mathbf{RE}$

$\mathbf{S}(i+1) \leftarrow \mathbb{S}_\sigma\left[\mathbf{S}(i+1)\right]$

$t(i+1) = \frac{1+\sqrt{1+4t^2(i)}}{2}$,

$\mu = \frac{t(i)-1}{t(i+1)}$,

$\mathbf{S}(i+1) \leftarrow \mathbf{S}(i+1) + \mu\left(\mathbf{S}(i+1) - \mathbf{S}(i)\right)$,

$\mathbf{R} \leftarrow \mathbf{X} - \mathbf{HP\Psi S}(i+1)\mathbf{E}^H$,

$i \leftarrow i+1$,

Until $i = i_{max}$,

Output: estimated delay Doppler map $\mathbf{S}^{\#} = \mathbf{S}(i)$.

3.3 Simulation results

The performances of the proposed algorithms are compared with those of the 1D-IHT [40], 1D-ZAP [41], 1D-ISTA [68], and 1D-FISTA [71] algorithms. Mean square error (MSE) is defined as:

$$MSE = \frac{1}{M \times N}\sum_{m=1}^{M}\sum_{n=1}^{N}\sqrt{\left(\tilde{s}_{mn} - s_{mn}\right)^2} \tag{3.20}$$

where $\tilde{s}_{mn}$ is the estimation of s_{mn}. A smaller MSE means a better estimation performance. The bandwidth of transmitted signal B is set at 1.5kHz. The pulse width T_p is set as 1ms. The T_{PRI} is set as 2ms. The Nyquist sampling rate should be set larger than $2.25 \times B$ to void the aliasing. $N = 16$ pulses are transmitted to estimate the Doppler shift frequency. The delay resolution of a grid is $1/B$, and the Doppler resolution of a grid is $2\pi/N$ [72]. Thus, the number of the delay grid is $\bar{M} = \lfloor(T_{PRI} - T_p) * 2.25B\rfloor = 33$, while the number of the Doppler shift grids is $N = 16$. Operator $\lfloor\cdot\rfloor$ denotes the largest integer. It is shown in Fig.3.1 (b) that the received signal should be sampled 528 times at Nyquist sampling rate. In the following experiments, the received signal is sampled by the RD at half of the Nyquist sampling rate. Fig.3.1 (c) shows that the received signal is only sampled 256 times. Thus, the matrix $\mathbf{HP\Psi} \in \mathbb{R}^{M\times N}$, where

$M = \lfloor 0.5 \times \bar{M} \rfloor = 19$ and $\mathbf{E} \in \mathbb{R}^{N \times N}$. According to 1D-CS algorithms, the requirement of the memory should be at least $(M \times N) \times N^2 = 160512$, while in the 2D-CS algorithms, it reduces to $(M' \times N) + P^2 = 883$. In the first experiment, there are $K = 5$ existing targets in the search field with the white Gaussian noise whose signal-to-noise ratio (SNR) is set as 10dB. The original delay-Doppler map is shown in Fig. 3.1 (a).

Figs.3.2–3.5 show that the estimated delay-Doppler map by the 1D-ZAP ($\lambda = 0.01$), 2D-ZAP ($\lambda = 0.01$), 1D-IHT ($\sigma = 0.03$), 2D-IHT ($\sigma = 0.03$), 1D-ISTA ($\lambda = 0.01$), 2D-ISTA ($\lambda = 0.01$), 1D-FISTA ($\lambda = 0.01$) and 2D-FISTA ($\lambda = 0.01$) algorithms, respectively. One can find that 2D-CS algorithms can estimate the delay-Doppler map as the same as 1D-CS algorithms.

Table.6.2 shows the calculating time of each algorithm. The CPU time is used as an index of complexity. The simulations are run in MATLAB R2017b with Intel Xeon E3-1270 v5, 3.60GHz processor and 16GB of memory under Microsoft Windows 10 Professional (64bit). The parameters of the proposed 2D-CS algorithms are set as the same as the 1D-CS algorithms. The table shows that the proposed 2D-CS algorithms have a faster calculation time than the 1D-CS algorithms, because our proposed algorithms have a low complexity.

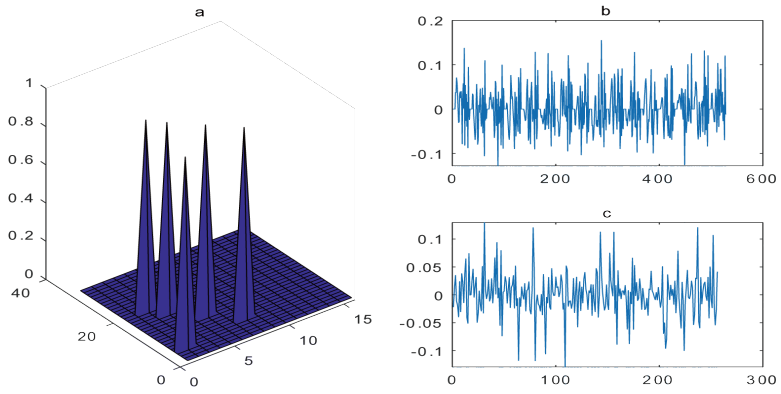


Figure 3.1: (a) The original delay-Doppler map; (b) The sampled data at Nyquist sampling rate; (c) The data sampled by the RD.

Table 3.1: The calculating time of each algorithm

	ZAP	IHT	ISTA	FISTA
1D	1.2397	0.8883	5.1217	1.7720
2D	0.2103	0.1386	0.2091	0.1728

Fig.3.6 shows the MSE performances of the proposed 2D-ZAP algorithms against the parameter λ. The MSE performances of the 1D-ZAP algorithm are plotted as a reference. The parameter α is set as 1. It can be found that the 2D-ZAP algorithm has the same MSE performance as the 1D-ZAP algorithm with the same λ. In other words, utilizing the 2D-ZAP

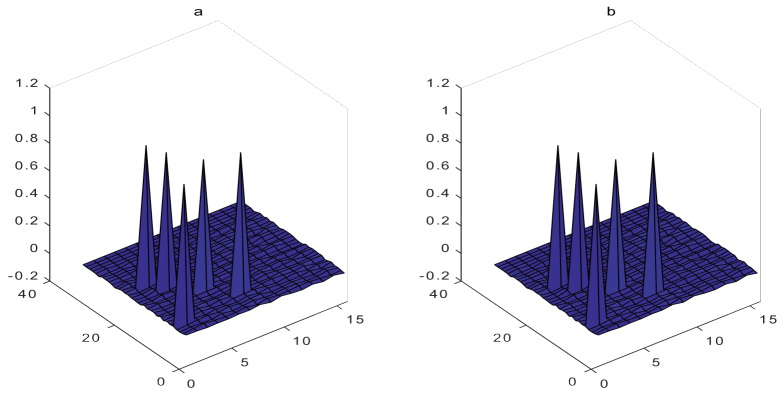


Figure 3.2: (a) The estimated delay-Doppler map by 1D-ZAP algorithm; (b) The estimated delay-Doppler map by 2D-ZAP algorithm.

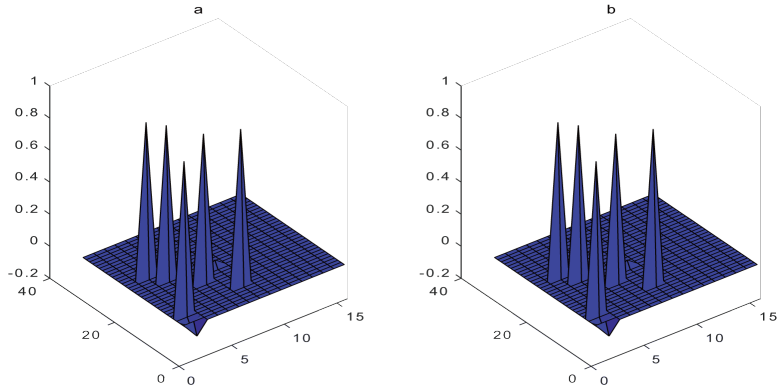


Figure 3.3: (a) The estimated delay-Doppler map by 1D-IHT algorithm; (b) The estimated delay-Doppler map by 2D-IHT algorithm.

algorithm does not reduce the estimation performance. Moreover, for either 1D or 2D-ZAP algorithm, a smaller λ results in a smaller steady state MSE at a slower convergence rate, while a larger λ results in a larger steady state MSE at a faster convergence rate.

Fig.3.7 shows the MSE performances of the proposed 2D-IHT algorithms against the parameter σ. The MSE performances of the 1D-IHT algorithm are plotted as a reference. It can be found that the 2D-IHT algorithm has the same MSE performance as the 1D-IHT algorithm with the same σ. In other words, utilizing the 2D-IHT algorithm does not reduce the estimation performance. Moreover, for either 1D or 2D-IHT algorithm, the MSE performance depends on the thresholding value σ. In this case, the least MSE can be achieved when $\sigma = 0.03$. Therefore, finding a suitable σ is the key problem in the 2D-IHT algorithm. It still needs to be studied deeply.

Fig.3.8 shows the MSE performances of the proposed 2D-ISTA algorithms against the parameter λ. The MSE performances of the 1D-ISTA algorithms with different λ are plotted as a reference. It can be found that the 2D-ISTA algorithm has the same MSE performance as the

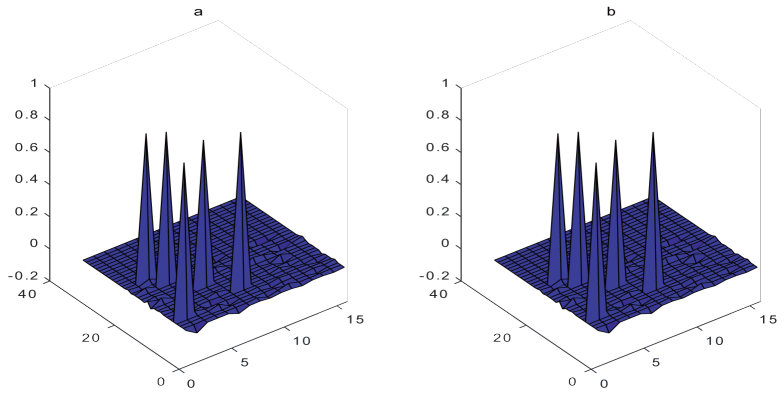


Figure 3.4: (a) The estimated delay-Doppler map by 1D-ISTA; (b) The estimated delay-Doppler map by 2D-ISTA.

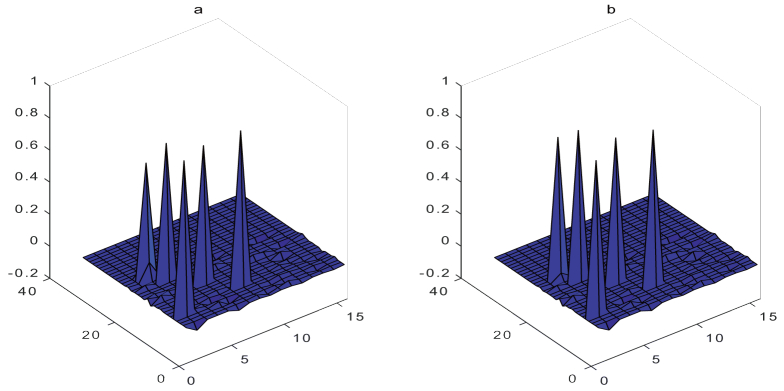


Figure 3.5: (a) The estimated delay-Doppler map by 1D-FISTA; (b) The estimated delay-Doppler map by 2D-FISTA.

1D-ISTA algorithm with the same λ. In other words, utilizing the 2D-ISTA algorithm does not reduce the estimation performance. Moreover, for either 1D or 2D-ISTA algorithm, a smaller λ results in a smaller steady state MSE and a slower convergence rate, while a larger λ results in a larger steady state MSE and a faster convergence rate. It is similar to 1D and 2D-ZAP algorithms.

Fig.3.9 shows the MSE performances of the proposed 2D-FISTA algorithm against the parameter λ. The MSE performances of the 1D-FISTA algorithms with different λ are plotted as a reference. It can be found that the 2D-FISTA algorithm has the same MSE performance as the 1D-FISTA algorithm with the same λ. In other words, utilizing the 2D-FISTA algorithm does not reduce the estimation performance. Since the 2D-FISTA algorithm is an improved 2D-ISTA algorithm, the two kinds algorithms have the similar feature that a smaller λ results in a smaller steady state MSE and a slower convergence rate, while a larger λ results in a larger steady state MSE and a faster convergence rate. Moreover, compared with the 2D-ISTA algorithm, the number of iterations is much smaller when the 2D-FISTA algorithm gets the convergence.

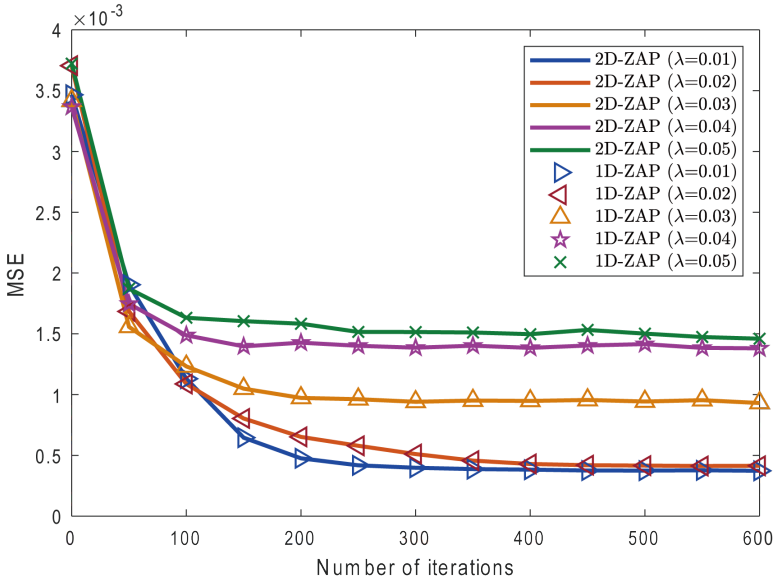


Figure 3.6: MSE curves of 1D-ZAP and 2D-ZAP algorithms with respect to λ .

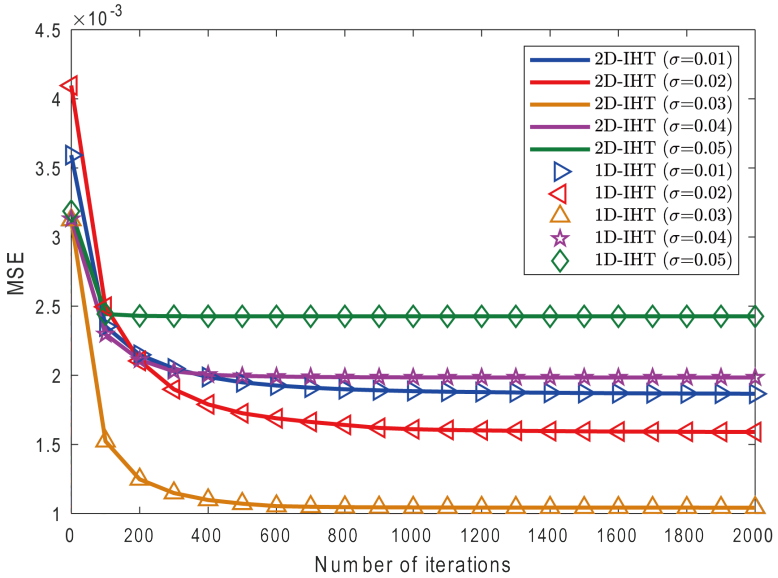


Figure 3.7: MSE curves of 1D-IHT and 2D-IHT algorithms with respect to σ.

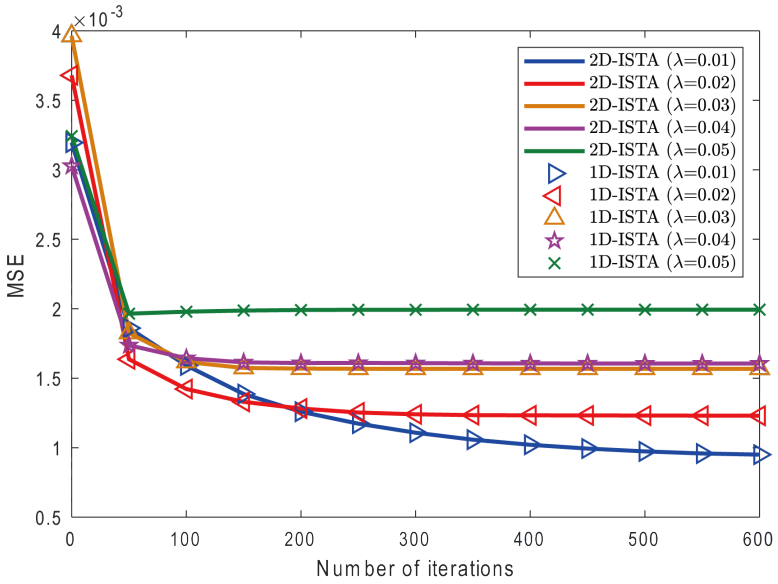


Figure 3.8: The MSE curves of 1D-ISTA and 2D-ISTA algorithms with respect to λ.

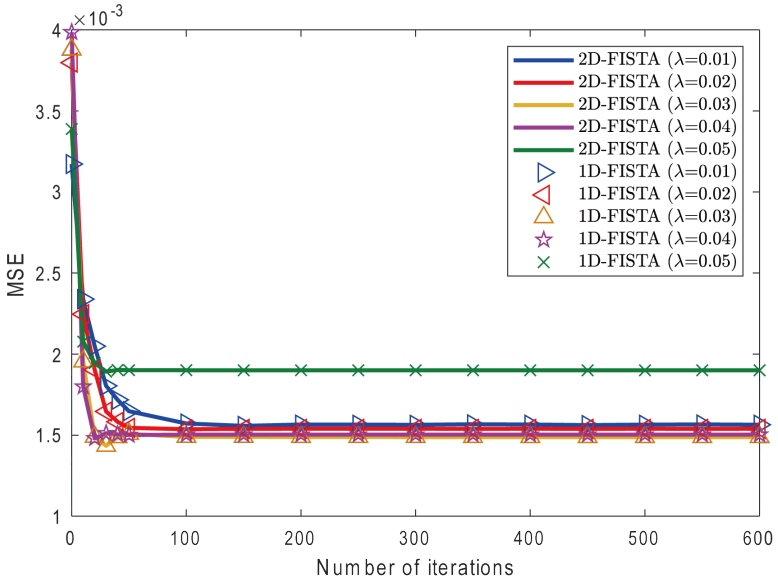


Figure 3.9: The MSE curves of 1D-FISTA and 2D-FISTA algorithms with respect to λ.

In the third experiment, the robustness to the Gaussian noise is considered. There are 5 existing targets within the search field. SNR is chosen from 0 to 10dB. For each SNR, all these algorithms are repeated 1000 times to calculate the MSEs. Other parameters are as the same as the first experiment. Fig. 3.10 shows that the MSEs of all algorithms decrease with the SNR increasing. Among these algorithms, the 2D-ZAP algorithm has the smallest MSE, and the 2D-ISTA algorithm has a smaller MSE than the 2D-FISTA algorithm, in each SNR environment. When SNR< 3dB, the 2D-FISTA algorithm has a smaller MSE than the 2D-IHT algorithm, while the 2D-FISTA algorithm has a larger MSE than the 2D-IHT algorithm when SNR> 3dB.

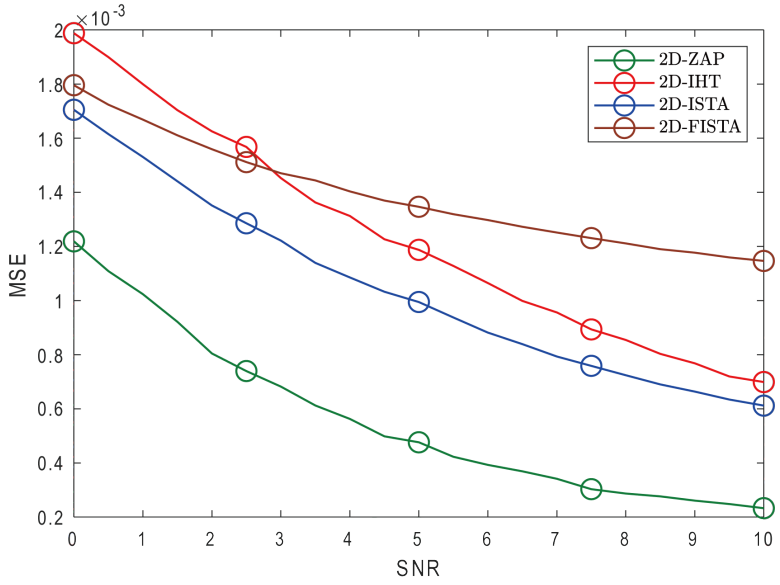


Figure 3.10: The MSE curves of 2D-CS algorithms against SNR.

In the fourth experiment, the robustness to the target number is considered. There are 1-10 existing targets within the search field. SNR is chosen as 10dB. For each target number, all these algorithms are repeated 1000 times to calculate the MSEs. Other parameters are set as the same as the first experiment. Fig.3.11 shows that the MSEs of these algorithms increase as the number of targets increases. Among all algorithms, the 2D-ZAP algorithm has the smallest MSE. The 2D-ISTA and 2D-IHT algorithms have smaller MSEs than the 2D-FISTA algorithm, in each SNR environment. When target number $K < 3$, the 2D-IHT algorithm has smaller MSE than the 2D-ISTA algorithm, while the 2D-IHT algorithm has a larger MSE than the 2D-ISTA algorithm when target number $K > 3$.

In the fifth experiment, the robustness to the compression ratio (CR) is considered. The CR is set from 0.2 to 1. CR = 1 means that the received signal is sampled at the Nyquist sampling rate. For each compression ratio, all these algorithms are repeated 1000 times to calculate the

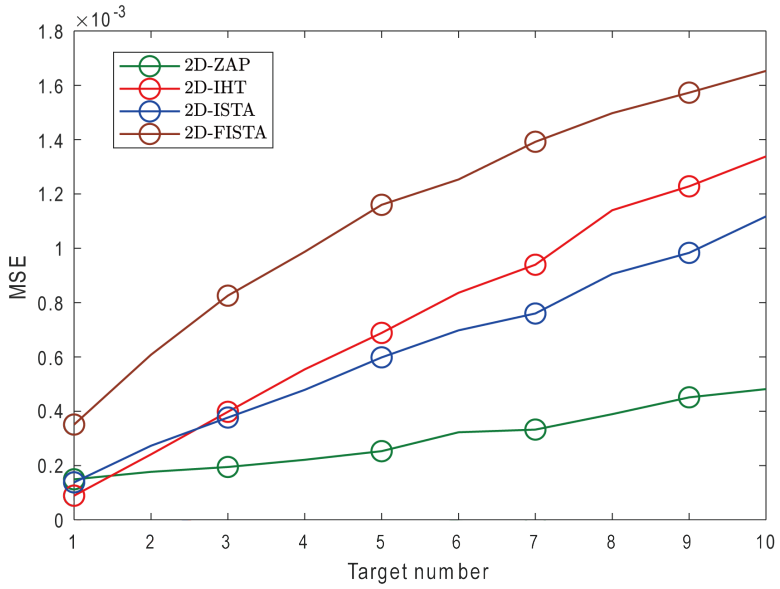


Figure 3.11: The MSE curves of 2D-CS algorithms against the number of targets.

MSEs. Other parameters are as the same as the first experiment. Fig.3.12 shows that the MSEs of these algorithms reduce as the compression ratio increases. When compression ratio is larger than 0.4, the 2D-ZAP algorithm has a similar MSE to the original sampling rate.

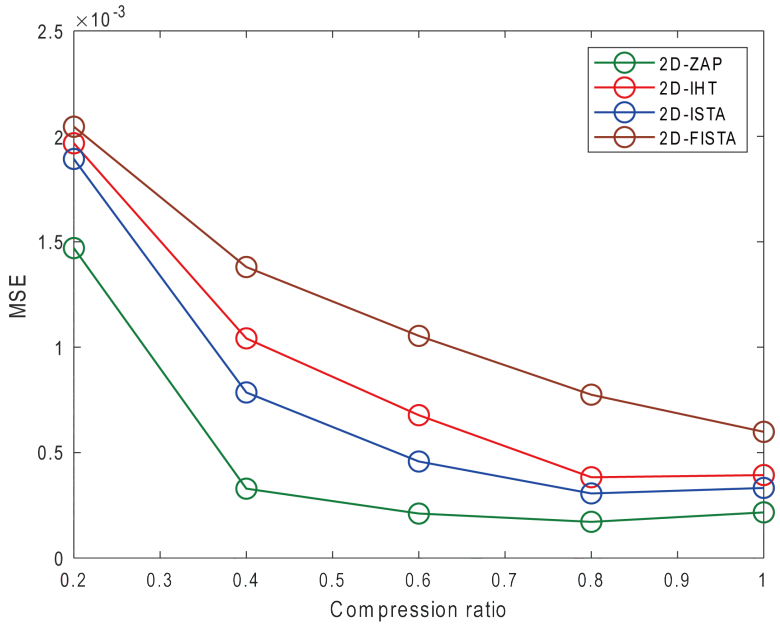


Figure 3.12: The MSE curves of 2D-CS algorithms against compressive radio.

We have proposed a 2D data model for the pulse Doppler radar system with the RD method. In this method, the data is under-sampled by a low rate ADC. Then the 2D-CS (i.e., 2D-ZAP, 2D-IHT, 2D-ISTA, and 2D-FISTA) algorithms have been proposed for detecting the sparse tar-

gets from the under-sampled data. Since the 2D-CS algorithms solve the 2D data model without vectorizing, the memory requirement and complexity are significantly reduced. Numerical simulations have been provided to validate the performances of our proposed algorithms.

Chapter 4

Robust 2D-CS-based algorithms for delay Doppler joint estimation

4.1 Non-Gaussian impulsive noise

Recall the sampled signal model which is given by: $\mathbf{X} = \mathbf{HPU} + \mathbf{W} = \mathbf{HP\Psi SE}^H + \mathbf{W}$, where $w_{ln} \in \mathbf{W}$ denotes the noise. As our previous discussion, $\mathbf{W}$ is assumed to satisfy the Gaussian distribution to simplify the model. However, the impulsive noise, such as environmental effects of atmospherics (lighting) and meteor train echoes [53], will occur in $\mathbf{W}$ in the practical radar system. The radar performance is easy to be degraded by the impulsive noise interference. These impulsive noise makes the statistical characteristics of noise significantly deviate from the Gaussian distribution. Its probability density function (PDF) decays more slowly than a Gaussian distribution, causing heavy tail. In order to evaluate the characteristics of the impulsive noise, the symmetric α-stable noise model [73–75] is used to describe the non-Gaussian impulsive noise. The characteristic function of α-stable distribution is defined as

$$r\left(z;\alpha,\beta,c,u\right) = e^{juz-|cz|^{\alpha}\times[1+j\beta\cdot\mathrm{sgn}(z)\varphi(z,\alpha)]} \tag{4.1}$$

where

- $\alpha \in (0,2]$ denotes the characteristic exponent. When $\alpha = 2$, the distribution becomes a Gaussian distribution. When α is closer to 0, the number of the impulse in noise is larger, and the amplitude of the impulse is larger;

- $\beta \in [-1,1]$ denotes the skewness parameter which controls the symmetry scenarios;

- $c \in [0, \infty)$ is a scale factor which plays a similar role as the variance of Gaussian distribution;

- $u \in (-\infty, \infty)$ is the location parameter.

When $\beta = 0$, the distribution is symmetric about u, and

$$\varphi(z, \alpha) = \begin{cases} \tan\left(\frac{\pi\alpha}{2}\right), & \alpha \neq 1; \\ -\frac{2\log|z|}{\pi}, & \alpha = 1. \end{cases} \tag{4.2}$$

In this thesis, only the symmetric α-stable noise model is considered, because it is the most common in a practical radar system [73], The PDF of the symmetric α-stable model is shown in Fig.4.1. The characteristic function of the symmetric α-stable distribution is simplified as $r(z; \alpha, c) = e^{-|cz|^{\alpha}}$. For convenience, the variance of the symmetric α-stable distribution is defined as $\sigma_n = c^{1/\alpha}$ which plays the role as the noise variance. The large variance results in that the approaches of the above 2D-CS algorithms are far from the desired original delay-Doppler map $\mathbf{S}$. Thus, the robust 2D-CS-based algorithms are expected.

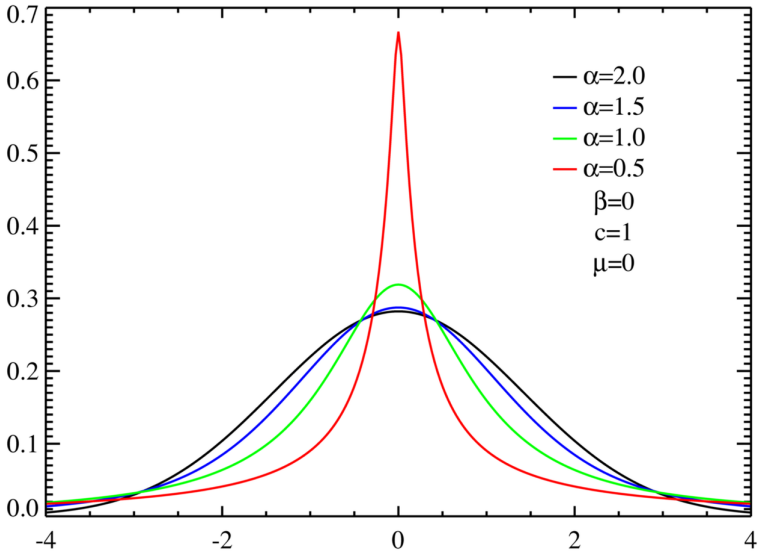


Figure 4.1: Symmetric α-stable distributions.

The reason for low accuracy is that ℓ_F-minimization for errors is not robust against the impulse noise. The key in our proposed robust 2D-CS-based algorithms is that ℓ_F-norm is replaced by more robust cost function, such as ℓ_1-norm, ℓ_p-norm $(0 < p < 1)$ and Lorentzian norm (LL_2-norm), which are presented in the following sections. These cost functions provide

the bias and efficiency when data is contaminated by a small noise and reliable albeit not optimal behavior when the data is contaminated by impulsive noise.

4.2 Proposed robust 2D-CS-based algorithms

4.2.1 The ℓ_1-based 2D methods

Using ℓ_1-norm instead of ℓ_F-norm in the data-fitting term in Eq. (3.1), the solving method can be written as

$$\mathbf{S} = \min \left\| \mathbf{X} - \mathbf{H}\mathbf{P}\mathbf{\Psi}\mathbf{E}^H \right\|_1^2 + \lambda \|\mathbf{S}\|_0. \tag{4.3}$$

Then the data fitting tern can be solved as:

$$\begin{aligned} \mathbf{S}(i+1) &= \mathbf{S}(i) + \nabla \mathbf{F}(\mathbf{S}(i)) \\ &= \mathbf{S}(i) + (\mathbf{H}\mathbf{P}\mathbf{\Psi})^H \mathbf{R}_{\ell_1} \mathbf{E}, \end{aligned} \tag{4.4}$$

where $\mathbf{R}_{\ell_1} = \operatorname{sgn}(\mathbf{R})$, and it provides low variance gradient estimates, with the effect of reducing the residual error. It means that ℓ_1-based 2D method does not pay attention to the amplitude of the residual error, because in the non-Gaussian impulsive noise environment, it is very possible that the large residual error is caused by impulsive noise. If we pay the indiscriminate attention to the amplitude of the residual error, the approaches of the above 2D-CS algorithms are far from the desired original delay-Doppler map.

The ℓ_1-based 2D-robust-ZAP (2D-RZAP(L1)) algorithm can be extended from 2D-ZAP algorithm as follows:

$$\mathbf{S}(i+1) = \mathbf{S}(i) + \mu (\mathbf{H}\mathbf{P}\mathbf{\Psi})^H \mathbf{R}_{\ell_1} \mathbf{E} + \lambda g(\mathbf{S}(i)), \tag{4.5}$$

where μ denotes the step-size which reduces the update rate of each element. With the same idea, the ℓ_1-based 2D-robust-IHT (2D-RIHT(L1)) algorithm can be extended from 2D-IHT algorithm as follows:

$$\mathbf{S}(i+1) = \mathbb{H}\left[\mathbf{S}(i) + \mu (\mathbf{H}\mathbf{P}\mathbf{\Psi})^H \mathbf{R}_{\ell_1} \mathbf{E}\right] \tag{4.6}$$

4.2.2 The ℓ_p-based 2D methods

The ℓ_1-based 2D method is a simple method to limit the influences of the too large residual error. However, it still cannot distinguish that the data is contaminated by a small noise or a large impulsive noise. Thus, this method does not utilize the data which is contaminated by a small noise effectively. The ℓ_p-based 2D method is proposed to reduce the influence of a large

residual error while the small residual error is utilized effectively. It is given by:

$$\mathbf{S} = \min \left\| \mathbf{X} - \mathbf{HP\Psi SE}^H \right\|_p^p + \lambda \|\mathbf{S}\|_0, \tag{4.7}$$

where $\mathbf{R}_{\ell_p}$ is the reweighted residual error matrix. The ith element can be written as

$$\mathbf{R}_{\ell_p}(i) = \frac{\|\mathbf{R}\|_p^{1-p} \operatorname{sgn}(\mathbf{R}(i))}{\varepsilon + |\mathbf{R}(i)|^{1-p}},$$

where $0 < p < 1$ and ε is a small constant bounding the term to avoid denominator becoming 0. It is a robust method since a large residual error does not influence the approach while a small residual error is utilized effectively.

The ℓ_p-based 2D-robust-ZAP (2D-RZAP (Lp)) algorithm is extended from 2D-ZAP algorithm as follows:

$$\mathbf{S}(i+1) = \mathbf{S}(i) + \mu (\mathbf{HP\Psi})^H \mathbf{R}_{\ell_p} \mathbf{E} + \lambda g(\mathbf{S}(i)), \tag{4.8}$$

and ℓ_p-based 2D-robust-IHT (2D-RIHT) algorithm is extended from 2D-IHT algorithm as follows

$$\mathbf{S}(i+1) = \mathbb{H}\left[\mathbf{S}(i) + \mu (\mathbf{HP\Psi})^H \mathbf{R}_{\ell_p} \mathbf{E}\right]. \tag{4.9}$$

The ℓ_p-based 2D methods is an effective method for non-Gaussian impulsive noise environment if the parameter p is selected properly. How to choose p depends on user experience. Hence, we will propose another norm for choosing the parameter automatically.

4.2.3 The LL_2-based 2D methods

The Lorentzian norm (LL_2-norm) is a robust norm that does not heavily penalized large deviations with the robustness depending on the scale parameter γ, which is chosen by the received signal automatically. Compared with the ℓ_p-based 2D methods, the LL_2-based 2D methods are able to adaptively select appropriate parameters to meet the need for high accuracy and fast convergence speed. The Lorentzian norm of a matrix $\mathbf{X} \in \mathbb{C}^{N \times M}$ is defined as:

$$\|X\|_{LL_2,\gamma} = \sum_{n=1}^{N} \sum_{m=1}^{M} \log\left(1 + \frac{x_{n,m}^2}{\gamma^2}\right), \tag{4.10}$$

where setting γ to half the data range of $\mathbf{X}$ as $\left(\mathbf{X}_{(1)} - \mathbf{X}_{(0)}\right)/2$, $\mathbf{X}_{(i)}$ is the i-quantile of vec($\mathbf{X}$). However, it often makes the Lorentzian norm to approximate ℓ_2-norm in simulation. According to [76], γ should be set as $\left(\mathbf{X}_{(0.875)} - \mathbf{X}_{(0.125)}\right)/2$ in general. It means that the measurement matrix with 25% of data corrupted and 75% well behaved by this value of γ.

Therefore, LL_2-based 2D methods are proposed to reduce the influence of a large residual error while the large residual error is utilized effectively. It is given as:

$$\mathbf{S} = \min \left\| \mathbf{X} - \mathbf{HP\Psi SE}^H \right\|_{LL_2,\gamma} + \lambda \|\mathbf{S}\|_0, \tag{4.11}$$

Then the data fitting term can be solved as:

$$\begin{aligned} \mathbf{S}\left(i+1\right) &= \mathbf{S}\left(i\right) + \nabla \mathbf{F}\left(\mathbf{S}\left(i\right)\right) \\ &= \mathbf{S}\left(i\right) + \left(\mathbf{HP\Psi}\right)^H \mathbf{R}_{LL_2,\gamma} \mathbf{E}, \end{aligned} \tag{4.12}$$

where

$$\mathbf{R}_{LL_2,\gamma}\left(i\right) = \frac{\gamma^2 \mathbf{R}\left(i\right)}{\gamma^2 + \mathbf{R}^2\left(i\right)}.$$

The LL_2-based 2D-robust-ZAP (2D-RZAP (LL2)) algorithm is extended from 2D-ZAP algorithm as follows:

$$\mathbf{S}\left(i+1\right) = \mathbf{S}\left(i\right) + \mu \left(\mathbf{HP\Psi}\right)^H \mathbf{R}_{LL_2,\gamma} \mathbf{E} + \lambda g\left(\mathbf{S}\left(i\right)\right), \tag{4.13}$$

and LL_2-based 2D-robust-IHT (2D-RIHT (LL2)) algorithm is extended from 2D-IHT algorithm as follows:

$$\mathbf{S}\left(i+1\right) = \mathbb{H}\left[\mathbf{S}\left(i\right) + \mu (\mathbf{HP\Psi})^H \mathbf{R}_{LL_2,\gamma} \mathbf{E}\right]. \tag{4.14}$$

In Fig. 4.2 shows the cost function of ℓ_1-norm (blue), ℓ_p-norm with $p = 0.5$ (yellow), and Lorentzian norm with $\gamma = 0.1$ (purple) and $\gamma = 1$ (green). The cost function of ℓ_2-norm (red) is also plotted as reference. Compared with ℓ_2 -norm, the functions of ℓ_1-norm, ℓ_p-norm and Lorentzian norm do not over penalize large deviations, which results in more robust for non-Gaussian impulsive noise. Moreover, ℓ_p-norm and Lorentzian norm are more robust to outliers than ℓ_1-norm because when $\mathbf{R}(i) \to \infty$, the formers do not increase their value as fast as the latter.

4.3 Simulations results

In this simulations, the robustness of the ZAP algorithms to the non-Gaussian noise is considered. The non-Gaussian impulsive noise satisfies the symmetric α-stable distribution, where $\alpha \in [0.5, 1, 1.5, 2]$. When α is close to 2, the noise is similar to a Gaussian noise, while when α is close to 0, the noise is similar to a non-Gaussian impulsive noise. The parameters are list in Table. 4.1

In Fig. 4.3, one can find that the MSE of the 2D-ZAP algorithm becomes large in the

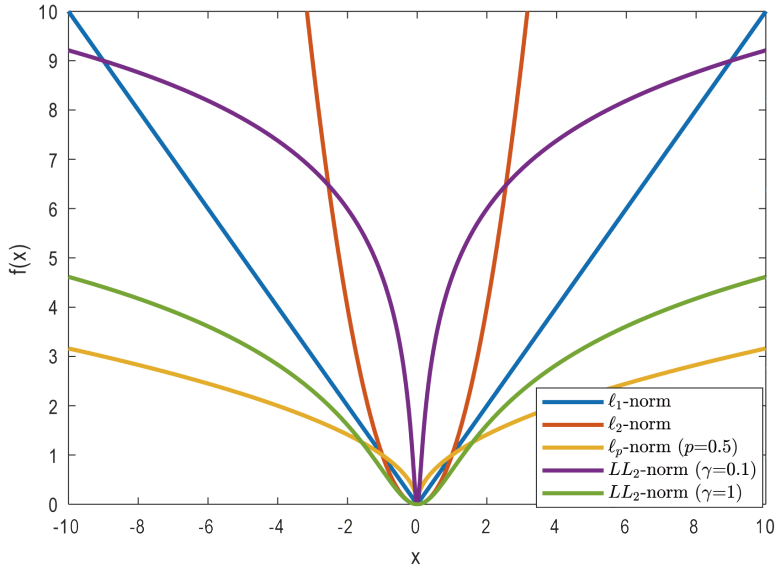


Figure 4.2: The cost function of ℓ_1-norm (blue), ℓ_p-norm with p=0.5 (yellow), and the Lorentzian norm with γ=0.1 (purple) and γ=1 (green). ℓ_2-norm (red) is plotted as reference..

	2D-ZAP	2D-RZAP(L1)	2D-RZAP(Lp),	2D-RZAP(LL2)
μ	/	0.01	0.0001	0.5
λ	0.01	0.01	0.001	0.0005

Table 4.1: The parameters in 2D-ZAP and 2D-RZAP algorithms

non-Gaussian impulsive noise environment. When $\alpha = 0.5$, the 2D-RZAP(L1) algorithm has the smallest MSE. When $1 \leq \alpha \leq 1.5$, the 2D-RZAP(Lp) and 2D-RZAP(LL2) algorithms also have the small MSE. And the MSEs of these two algorithms is smaller than the 2D-RZAP(L1) algorithm. Only when $\alpha = 2$, namely in the Gaussian noise environment, the 2D-ZAP algorithm has the smallest MSE. The main reason is that the cost function of the 2D-ZAP algorithm is most sensitive to the noise, while the cost function of the 2D-RZAP(L1) algorithm is most non-sensitive to noise.

In the simulation, the robustness of the IHT algorithms to the non-Gaussian noise is considered. The parameters are list in Table. 4.2.

	2D-IHT	2D-RIHT(L1)	2D-RIHT(Lp),	2D-RIHT(LL2)
μ	/	0.02	0.0002	0.1
σ	0.004	0.1	0.1	0.1

Table 4.2: The parameters in 2D-IHT and 2D-RIHT algorithms

In Fig. 4.4, as the same as the 2D-ZAP algorithm, the MSE of the 2D-IHT algorithm becomes large in the non-Gaussian impulsive noise environment. When $\alpha = 0.5$, the 2D-RIHT(Lp) and 2D-RIHT(LL2) algorithms have the small MSE. When $1 \leq \alpha \leq 1.5$, the 2D-RIHT(L1), 2D-

RIHT(Lp) and 2D-RIHT(LL2) algorithms also have the small MSEs. Only when $\alpha = 2$, namely in the Gaussian noise environment, the 2D-IHT algorithm has the smallest MSE. The main reason is that the cost function of the 2D-IHT algorithm is most sensitive to the impulsive noise, while the cost functions of the 2D-RIHT(L1, Lp and LL2) algorithms are non-sensitive to impulsive noise.

The robust 2D-CS algorithms (2D-RZAP(L1, Lp, LL2) and 2D-RIHT(L1, Lp, LL2)) have been given for non-Gaussian impulsive noise environment. Numerical simulations have been provided to validate the performances of our proposed algorithms.

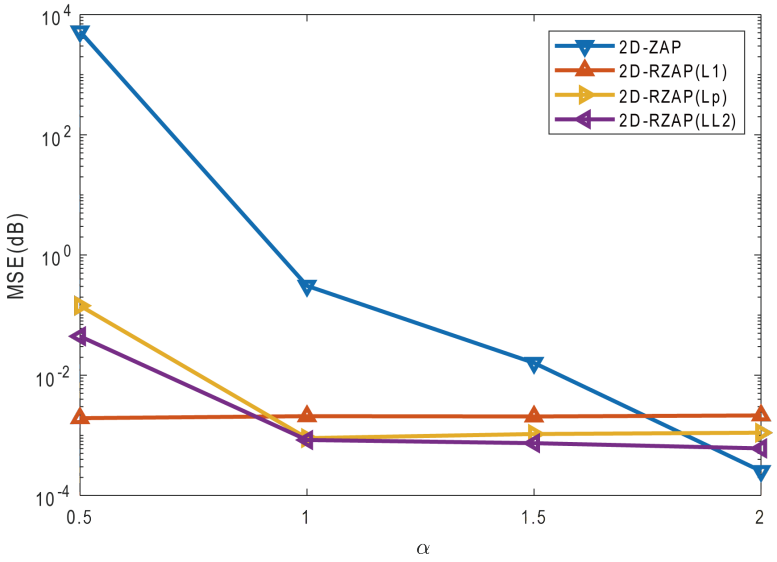


Figure 4.3: MSE curves of 2D-ZAP and 2D-RZAP algorithms with respect to α.

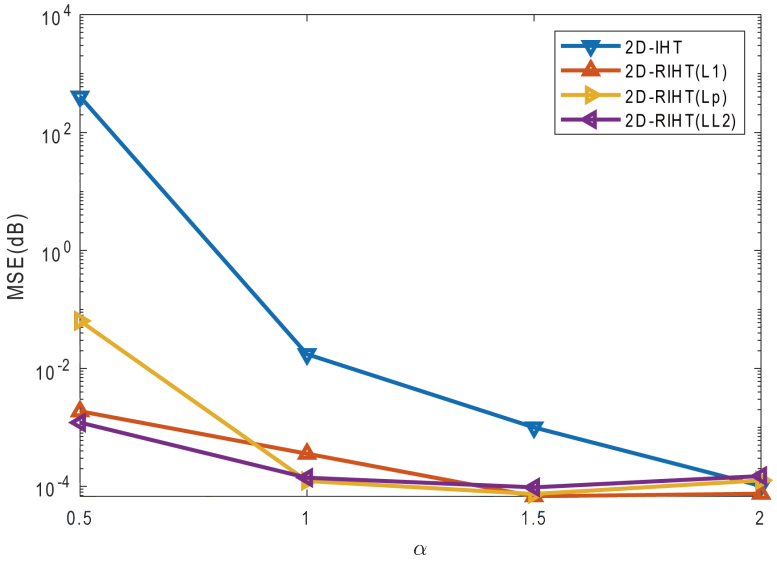


Figure 4.4: MSE curves of 2D-IHT and 2D-RIHT algorithms with respect to α.

Chapter 5

Direction-of-arrival estimation model for MIMO radar system

5.1 Mathematical model

In MIMO radar systems, multiple antennas also known as an array are used for target detection. There are many types of arrays for different needs, such as (the position of antennas in each array is shown in Fig. 5.1):

- Uniform Linear Array (ULA) [77]: all the M antennas lie on a line and the distance between the adjacent antennas is identical d for any two adjacent antennas. It is a simple case that can be used to estimate the azimuth angle of the target.
- Uniform Plane Array (UPA) [78]: Antennas are placed on a rectangular grid where the nodes are spaced $d1$ along x-axis and $d2$ along y-axis (if $d1 = d2 = d$, a square grid is got). It can be used to estimate both the azimuth and elevation angles of the target.
- Uniform Circular Array (UCA) [79]: The antennas are placed on a plane in a polar grid. For a fixed radial distance we have a circle on which the antennas are placed. It can also be used to estimate both the azimuth and elevation angles of the target.
- Sparse Array [80–82]: The distance between the adjacent antennas is non-uniform. Compared with the uniform array, the sparse array can achieve the same resolution, while the number of antennas, the complexity, size and weight of the system are reduced.

Other arrays can be seen as the expansion of the ULA. Hence, in this thesis, the ULA is introduced and used to validate the performances of our proposed algorithms due to its simplicity and typicality.

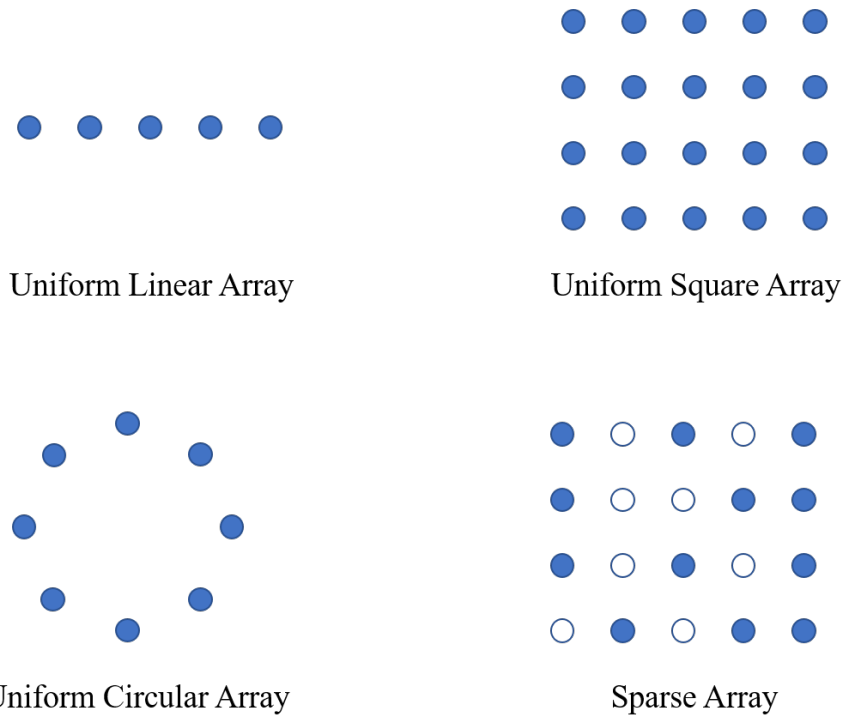


Figure 5.1: Schematic diagram of antenna positions of various arrays

Considering the two antennas in Fig. 5.2 and a far field point, the far field is the region beyond radial distance R which satisfies as

$$\begin{aligned} &R > D, \\ &R > \lambda, \\ &R > \tfrac{2D^2}{\lambda}, \end{aligned}$$

where and λ is the wavelength of signals. In this region, the radial component of the electric and magnetic fields are negligible and only the angles components matter. In general, especially in wireless communications, antennas and wavelengths are small enough that we can safely assume the far-field condition holds.

There is a relationship between the direction of targets and the phase shift of the received signals We assume that the target source signal comes from a far field point whose distances from Antenna 0 and Antenna 1 with R_0 and R_1 respectively. The direction of the target is θ. The time takes for the signal to arrival the Antenna 0 is $t_0 = \frac{R_0}{c}$, and to arrival the Antenna 1

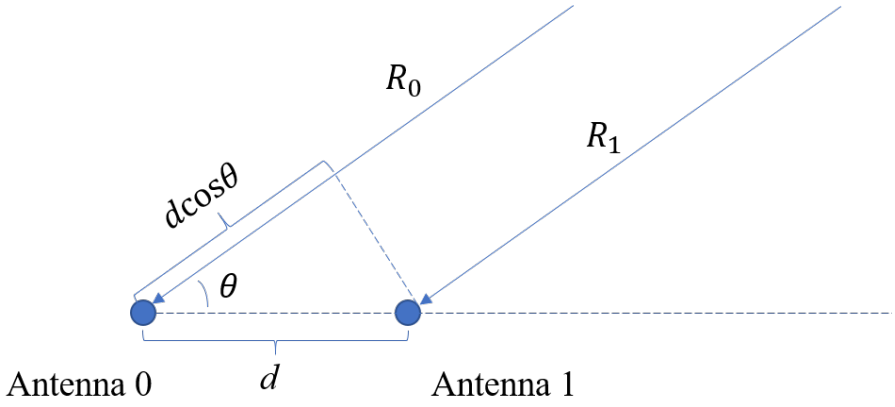


Figure 5.2: An array of two antennas

is $t_1 = \frac{R_1}{c}$, where c is the speed of light. The time difference is

$$\begin{aligned} \Delta t &= t_0 - t_1 \\ &= \frac{R_0 - R_1}{c} \\ &= \frac{d\cos\theta}{c}. \end{aligned} \tag{5.1}$$

For the signal with the frequency f, this time difference is equivalent to a phase shift:

$$\begin{aligned} e^{j2\pi f\Delta t} &= e^{j2\pi f d\cos\theta/c} \\ &= e^{j2\pi d\cos\theta/\lambda} \\ &= e^{jpd\cos\theta}, \end{aligned} \tag{5.2}$$

where $p = 2\pi/\lambda$. The signal at the Antenna 1 (s_1) and the signal at the Antenna 0 (s_0) are related by

$$s_1 = s_0 e^{jpd\cos\theta}. \tag{5.3}$$

The two-antenna steering vector can be written as

$$\mathbf{s}(\theta) = s_0 \begin{bmatrix} 1 & e^{jpd\cos\theta} \end{bmatrix}^T. \tag{5.4}$$

The signal s_0 is a constant across the steering vector. This constant can be safely ignored, because the information on the signal direction is only in the vector $\mathbf{a}(\theta) = \begin{bmatrix} 1 & e^{jpd\cos\theta} \end{bmatrix}^T$. Generalizing this to a ULA with M antennas, the steering vector associated with an angle θ can be written as

$$\mathbf{a}(\theta) = \begin{bmatrix} 1 & e^{jpd\cos\theta} & \cdots & e^{jpd(M-1)\cos\theta} \end{bmatrix}^T \tag{5.5}$$

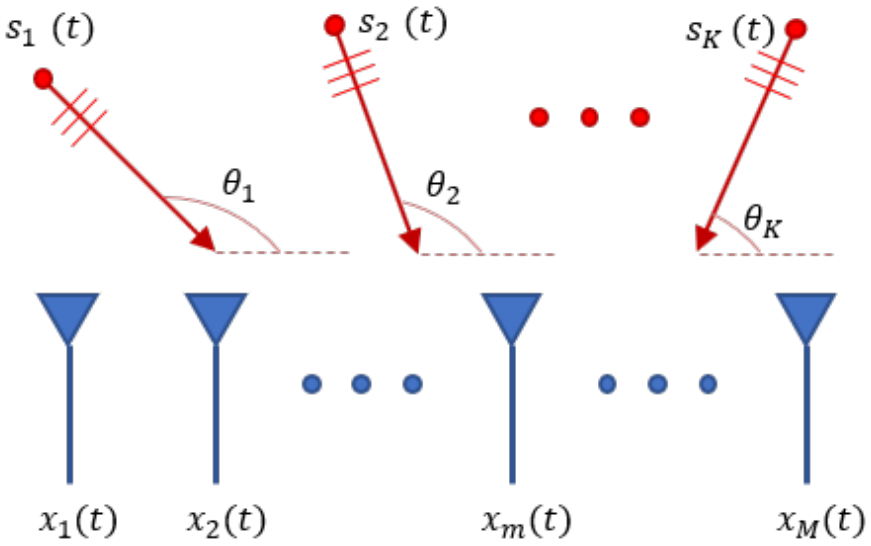


Figure 5.3: A reference scenario: source signals, denoted as $s_k(t)$, impinge on the array at angles θ_k, with the received signals denoted as $x_m(t)$.

In general, there are multiple targets at the same time, which is shown in Fig. 5.3. With the definition of the steering vector, the received signal model with multiple targets can be written as:

$$\mathbf{x}(t) = \sum_{k=1}^{K} s_k(t)\mathbf{a}(\theta_k) + \mathbf{w}(t), \tag{5.6}$$

where K is the number of targets, t is the sample time (snapshot), $\mathbf{x}(t) = \begin{bmatrix} x_1(t) & \cdots & x_M(t) \end{bmatrix}^T$ is the received signal, $\mathbf{w}(t) = \begin{bmatrix} w_1(t) & \cdots & w_M(t) \end{bmatrix}^T$ is the Gaussian noise, and $s_k(t)$ is the kth source signal at time t. The DOA estimation problem for massive MIMO systems can be stated as follows: Given $\mathbf{x}(t)$ and the mapping $\theta_k \to \mathbf{a}(\theta_k)$, find the unknown directions θ_k and the number K.

5.2 Single measurement vector and multiple measurement vector

However, in real MIMO radar systems, $\mathbf{a}(\theta_k)$ is unknown because it depends on θ_k which needs to be solved. To overcome this difficulty, an overcomplete representation of $\mathbf{x}(t)$ is developed in [56]. In this framework, the whole direction area of interest is divided into N distinct potential DOAs. Define the potential DOA vector as $\boldsymbol{\vartheta} = \begin{bmatrix} \theta_1 & \cdots & \theta_N \end{bmatrix}$. In general, the number of potential DOAs N is much larger than the number of the source signals K, or even the number of sensors M. The overcomplete dictionary consists of the steering vectors for each element in

$\boldsymbol{\vartheta}$ as

$$\mathbf{A} = \left[\begin{array}{ccc} \mathbf{a}(\theta)_1 & \cdots & \mathbf{a}\left(\theta_N\right) \end{array} \right]. \tag{5.7}$$

Since $\boldsymbol{\vartheta}$ is a given vector, the new manifold matrix $\mathbf{A}$ is known. Thus, the received signal can be formulated as a sparse representation problem:

$$\begin{aligned} \mathbf{x}\left(t\right) &= \sum_{n=1}^{N} \mathbf{a}\left(\theta_n\right) s_n\left(t\right) + \mathbf{w}\left(t\right) \\ &= \mathbf{A}\mathbf{s}\left(t\right) + \mathbf{w}\left(t\right), \end{aligned} \tag{5.8}$$

where $\mathbf{s}(t) = \left[\begin{array}{ccc} s_1\left(t\right) & \cdots & s_N\left(t\right) \end{array} \right]^T$ is represented as the source signals from $\left[\begin{array}{ccc} \theta_1 & \cdots & \theta_N \end{array} \right]$ at time t. Ideally, the nth component $s_n(t)$ of $\mathbf{s}(t)$ is nonzero if and only if $\theta_n = \theta_k$ for some k, and in that case $s_n(t) = s_k(t)$. It may not hold exactly that θ_k is equal to an element of $\boldsymbol{\vartheta}$ for any k in practice. But it can be found that when $\boldsymbol{\vartheta}$ is dense enough, θ_n will be approximately equal to an element of $\boldsymbol{\vartheta}$, and the remaining modeling error is absorbed in the residual term $\mathbf{w}\left(t\right)$ [10]. In general, K is much smaller than N, hence most elements in $\mathbf{s}(t)$ are zeros, and such a vector $\mathbf{s}(t)$ is a sparse vector.

If only a single snapshot can be used for DOA estimation, e.g., $t = T = 1$ where T is the total number of snapshots, $\mathbf{s}(t)$, $\mathbf{x}(t)$ and $\mathbf{w}(t)$ can be abbreviated as $\mathbf{s}$, $\mathbf{x}$ and $\mathbf{w}$, respectively. Thus, solving the vector $\mathbf{s}$ from $\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{w}$ can be formulated as an single measurement vector (SMV) problem, and it can be solved by CS method (refer to Eq. 2.30) directly.

In real applications, there are multiple snapshots for DOA estimation generally. The SMV problem cannot improve the accuracy for DOA estimation with multiple snapshot. Assuming aggregation over T snapshots with $1 \leq t \leq T$, where T is smaller than the requirement of the traditional DOA estimation algorithm, the observation model can be written as the multiple measurement vector (MMV) problem:

$$\mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{W}, \tag{5.9}$$

where

$$\begin{aligned} \mathbf{X} &= \left[\begin{array}{ccc} \mathbf{x}\left(1\right) & \cdots & \mathbf{x}\left(T\right) \end{array} \right], \\ \mathbf{S} &= \left[\begin{array}{ccc} \mathbf{s}\left(1\right) & \cdots & \mathbf{s}\left(T\right) \end{array} \right], \\ \mathbf{W} &= \left[\begin{array}{ccc} \mathbf{w}\left(1\right) & \cdots & \mathbf{w}\left(T\right) \end{array} \right]. \end{aligned}$$

In the MMV problem, if $\theta_n = \theta_k$ for some k, the elements in nth row of the $\mathbf{S}$ are nonzero, otherwise they are zeros. If matrix $\mathbf{S}$ can be solved from $\mathbf{X}$, the positions of the nonzero elements in $\mathbf{S}$ represent the true directions. When the true directions are time-invariant over

the period of observation, the nonzero elements share K same rows corresponding to the true directions. Therefore, only K rows in $\mathbf{S}$ are nonzero. Such an $\mathbf{S}$ is called jointly K-sparse matrix [83, 84]. As a result, the task of DOA estimation with the MMV model can be changed to determine the positions of nonzero elements in $\mathbf{S}$, and the multiple snapshots in the MMV model can improve the accuracy compared with the SMV model. The jointly sparse matrix $\mathbf{S}$ can be exactly estimated by [57]:

$$\min \|\mathbf{S}\|_{2,1}, \quad \text{s.t.} \|\mathbf{X} - \mathbf{A}\mathbf{S}\|_F \leq \varepsilon, \tag{5.10}$$

where ε is a small constant, $\|\cdot\|_F$ denotes the Frobenius norm, and $\|\cdot\|_{2,1} = \sum_i \sqrt{\sum_j [\cdot]_{i,j}^2}$ denotes the $\ell_{2,1}$-norm [85]. The $\ell_{2,1}$-norm ensures that the number of the nonzero rows in $\mathbf{S}$ is the least. Once $\mathbf{S}$ is computed by Eq. 5.10, the peaks of ℓ_2-norm of each row in $\mathbf{S}$ provide the true directions.

In practice, as the same as the case of the SMV model, equality $\theta_n = \theta_k$ may not hold exactly for any k. Nevertheless, making the grid dense enough, one can ensure $\theta_n \approx \theta_k$ closely, and the remaining modeling error can be absorbed in $\mathbf{W}$. However, we cannot make the grid very dense since it will increase the computation time. As a natural solution, the multi-resolution grid refinement method has been proposed in [56]. This method divides the range of interest into a coarse sampling grid first, and searches the potential DOAs where sources may exist by a sparse recovery method. Then the candidate grids obtained in the first stage are divided into a denser sampling grid and the same sparse recovery method is used again to search the sources.

In contrast, a coarse sampling grid results in a short computation time but low resolution estimation performance. In the worst case, the directions nearby the true sources cannot be obtained. An example for the results of different grid refinements solved by the sparse recovery algorithm is shown in Fig. 5.4, where the true signals come from $[44°, 79°, 88°]$. Fig. 5.4 (a) shows the ideal situation, where the interval between two sampling grids $\Delta\theta$ is set to $1°$. We see that all true directions can be found. In Fig. 5.4(b), when $\Delta\theta = 5°$, the some directions closest to the true directions can be estimated. Since the true directions are not on-grid, they cannot be achieved directly, but we see that the estimated directions are very close to these true directions. In Fig. 5.4(c), when $\Delta\theta = 10°$, the estimated directions are quite different from the true directions. That is, the sparse recovery method fails to find the candidates for true sources. Hence, if an effective method to find the true directions with a coarse sampling grid, the multi-resolution method will become more useful.

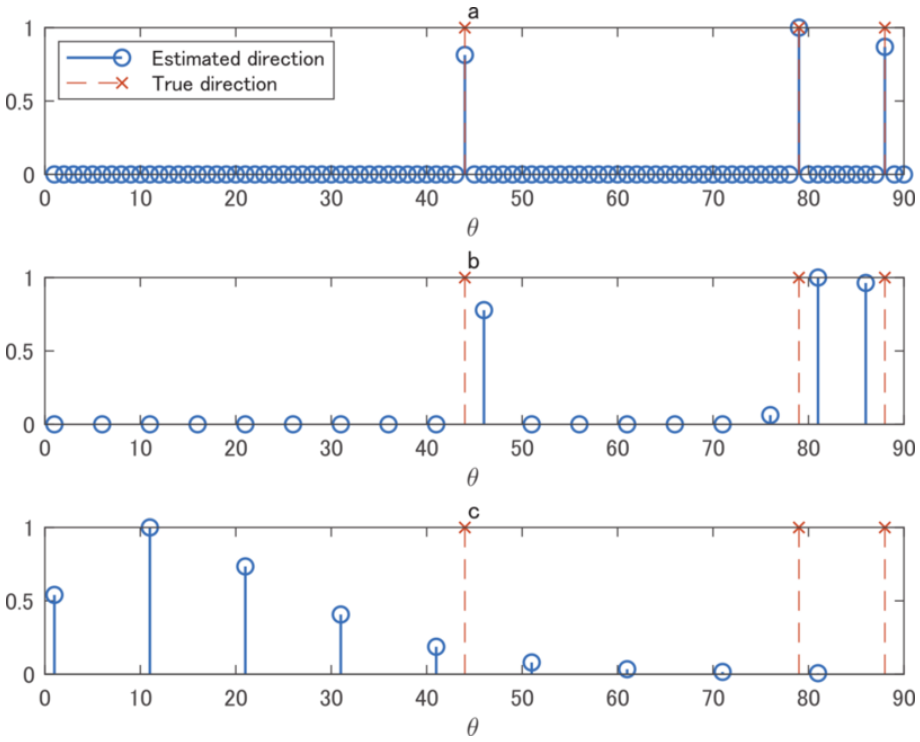


Figure 5.4: Results estimated by the CS-based algorithm for different $\Delta\theta$. (a) $\Delta\theta = 1°$; (b) $\Delta\theta = 5°$; (c) $\Delta\theta = 10°$.

Chapter 6

Dimension-reduced DOA estimation based on $\ell_{2,1}$-norm penalty

In this chapter, a dimension-reduced DOA estimation based on CS is proposed. First, a low-resolution method [86] is used to quickly identify the candidate or potential areas where true sources may exist. Then, $\ell_{2,1}$-norm minimization (a high-resolution method) is used to locate the sources.

6.1 Pre-estimation

The mismatch problem limits the performance of the CS-based DOA estimation with a coarse sampling grid as shown in Fig. 5.4. Therefore, under the CS framework, it is difficult to find the possible areas where targets exist. In order to avoid the fail to find the candidate or potential regions and, at the same time, to reduce the computational cost more effectively, a low-resolution method called the conventional beam former (CBF) is used to identify the potential DOA closest to the corresponding true sources. It can be used to quickly identify the candidate or potential areas even for the sampling grid coarser than the one in Fig .5.4.

The CBF identifies the candidate or potential areas by checking the correlation between an estimation and a source. Even when the sampling grid is coarse and the closest potential DOA is away from the source, the correlation between the estimation on the source and the closest grid to the source is still high. While the correlations between the source and the estimations on the other potential DOAs are near to zero. Therefore, the CBF can identify the candidate or potential areas where true sources may exist for rather coarse sampling grid. Thus, the areas that obviously contain no sources (where the correlation is near to zero) can be removed, while those that possibly include true sources (where the correlation is relatively high) can be kept for further detailed searching.

Assume that there is an observed signal $\mathbf{X} = \mathbf{a}(\theta)\mathbf{S}$ whose source comes from θ and a candidate signal $\bar{\mathbf{X}} = \mathbf{a}(\phi_j)\bar{\mathbf{S}}$ whose source comes from a candidate direction (ϕ_j) . For convenience, we set $\bar{\mathbf{S}}$ as 1 and $\theta = \phi_i$. The output power $V(\phi_j)$ of the CBF whose inputs are the observed signal $\mathbf{X}$ and the candidate signal $\bar{\mathbf{X}}$ is used to locate the sources. It is defined as:

$$\begin{aligned} W\left(\phi_j\right) &= \bar{\mathbf{X}}^H \mathbf{X}\mathbf{X}^H \bar{\mathbf{X}} \\ &= \mathbf{a}^H\left(\phi_j\right)\mathbf{X}\mathbf{X}^H\mathbf{a}\left(\phi_j\right) \\ &= \mathbf{a}^H\left(\phi_j\right)\mathbf{a}\left(\phi_i\right)\mathbf{R}_{ss}\mathbf{a}^H\left(\phi_i\right)\mathbf{a}\left(\phi_j\right), \end{aligned} \tag{6.1}$$

where $\mathbf{R}_{ss} = \mathbf{S}\mathbf{S}^H$. When $\mathbf{X}$ is given, the output power $W(\phi_j)$ is determined by the correlation between $\mathbf{a}(\phi_i)$ and $\mathbf{a}(\phi_j)$. Only when $\phi_i = \phi_j$, the absolute value of the correlation is the largest (see Fig. 6.1).

Assume that ϕ_i and ϕ_j is the i-th and j-th elements of $\{\phi_1, ..., \phi_P\}$ respectively, namely, $\mathbf{a}(\phi_i)$ and $\mathbf{a}(\phi_j)$ are the i-column and j-column of $\mathbf{\Phi}$ respectively. The correlation between $\mathbf{a}(\phi_i)$ and $\mathbf{a}(\phi_j)$, i.e., the i-th and the j-th columns of $\mathbf{\Phi}$ is defined as [87]

$$\begin{aligned} \chi\left(\phi_i, \phi_j\right) =& \frac{\mathbf{a}^H(\phi_i)\mathbf{a}(\phi_j)}{M} \\ =& \frac{1}{M}\sum_{m=0}^{M-1}\exp\left\{j2\pi d_m\left[\cos\left(\phi_i\right) - \cos\left(\phi_j\right)\right]\right\}. \end{aligned} \tag{6.2}$$

Due to that the source may come from any direction of interest, the measure of correlation between any two columns of $\mathbf{\Phi}$ is necessary. An intuitive measure is its mutual coherence defined as [88, 89]

$$\frac{\mathbf{\Phi}^H\mathbf{\Phi}}{M} = \begin{bmatrix} 1 & \cdots & \chi\left(\phi_1, \phi_P\right) \\ \chi\left(\phi_2, \phi_1\right) & \cdots & \chi\left(\phi_2, \phi_P\right) \\ \vdots & \vdots & \vdots \\ \chi\left(\phi_P, \phi_1\right) & \cdots & 1 \end{bmatrix}, \tag{6.3}$$

Then, the absolute Gram matrix is defined as

$$\mathbf{G} \triangleq \frac{\left|\mathbf{\Phi}^H\mathbf{\Phi}\right|}{M}. \tag{6.4}$$

The element in the i-th row and the j-th column of $\mathbf{G}$ can be written as $\mathbf{G}_{i,j} = \left|\chi\left(\phi_i, \phi_j\right)\right|$, which represents the absolute value of the correlation between $\mathbf{a}(\phi_i)$ and $\mathbf{a}(\phi_j)$. A large $\mathbf{G}_{i,j}$ denotes a high correlation, while a small $\mathbf{G}_{i,j}$ denotes a low correlation.

Fig. 6.1 (a) and (c) show the matrix $\mathbf{G}$ with $M = 10$ and 20 in θ space, respectively. Fig. 6.1 (b) and (d) show the corresponding beampattern with $M = 10$ and 20 when $\phi_i = 90°$, respectively. The red points denote the positions of the null points where $\mathbf{G}_{i,j} = 0$. In Fig. 6.1

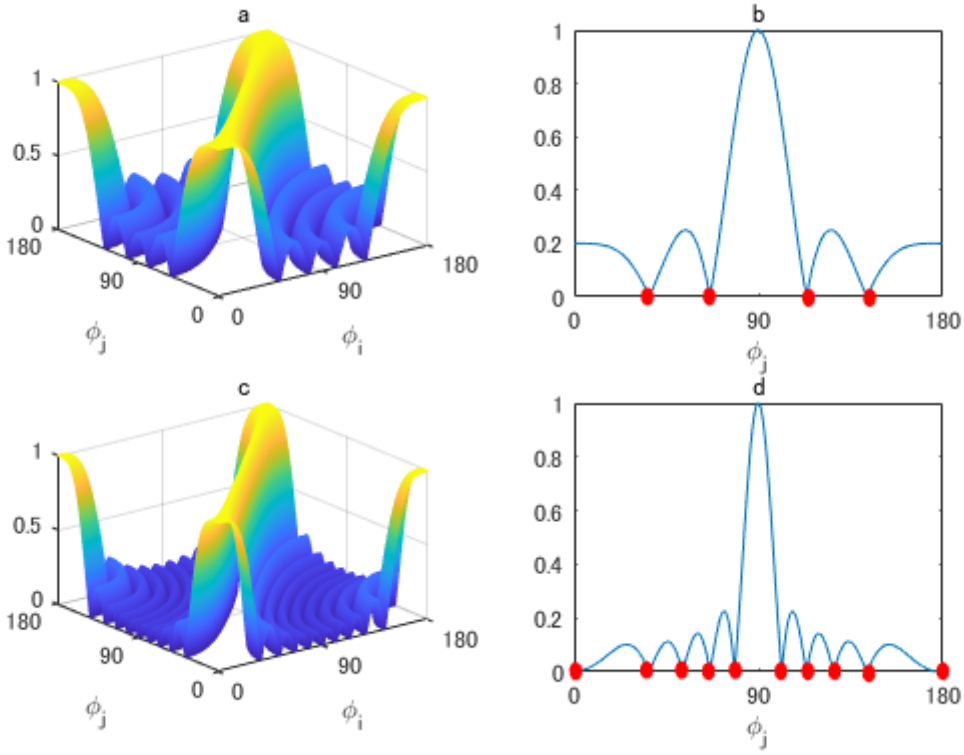


Figure 6.1: Gram matrix for a ULA with $M = 10$ sensors (a) and $M = 20$ sensors (c), with $d/\ell = 1/2$. The corresponding beam patterns when $\theta_k = 90°$ are shown in (b) ($M = 10$) and (d) ($M = 20$). The red points show the positions of the null points.

(b) and (d), it can be found that the position of the closest null point to ϕ_i is determined by the number of sensors M. A smaller M can yield less null points and a farther null point from ϕ_i, thus resulting in a wide beam width. When ϕ_j is in the area between two points which are the most closest to ϕ_i, the correlation also can be used to determine whether the target exists. According to Appendix A, the closest null point to ϕ_i is $\phi_j = \arccos\left[\cos\left(\phi_i\right) \pm \frac{2}{M}\right]$. In other words, the correlation between $\mathbf{a}(\phi_i)$ and $\mathbf{a}(\phi_j)$ is still high, although ϕ_j is away from ϕ_i. Hence, it is feasible to determine whether there is a source in one divided search area based on one ϕ_j, which belongs to this search area. In addition, for a small-scale ULA, all the observations $\mathbf{X}$ can be used directly due to its low resolution. However, for a large-scale ULA, the high resolution is not conducive to the search area division, as the correlation decreases rapidly when ϕ_j is slightly away from ϕ_i. Therefore, only the observations $\mathbf{X}$ measured by the first $\bar{M}$ sensors are required. In this way, the resolution can be reduced artificially for area search. $\bar{M}$ can be determined by the number of search areas according to the absolute Gram matrix.

In the first stage, the search range is divided into $\bar{P}$ ($\bar{P} < P$) sampling grids $\phi_{\bar{p}}$ coarsely, where $\bar{p} = 1, 2, ..., \bar{P}$. Thus, $\mathbf{\Phi} \triangleq [\mathbf{a}(\phi_1), ..., \mathbf{a}(\phi_{\bar{P}})]$. The interval between two adjacent search areas is $\Delta\phi = \phi_{\bar{p}+1} - \phi_{\bar{p}}$. Then, the largest correlation can be found as follows:

$$\phi = \arg\max_{\phi_{\bar{p}}} \mathbf{J}, \tag{6.5}$$

where $\mathbf{J} = \left[J\left(\phi_1\right), ..., J\left(\phi_{\bar{p}}\right), ..., J\left(\phi_{\bar{P}}\right)\right]$ is a correlation filter used to output the correlation between $\mathbf{a}\left(\phi_{\bar{p}}\right)$ and the observations $\mathbf{X}$. It can be defined as follows:

$$J\left(\phi_{\bar{p}}\right) = \mathbf{a}^H\left(\phi_{\bar{p}}\right) \mathbf{R}_{\mathbf{xx}} \mathbf{a}\left(\phi_{\bar{p}}\right), \tag{6.6}$$

where $\mathbf{R}_{\mathbf{XX}} = \mathbf{X}\mathbf{X}^H$ is the covariance matrix of the observations $\mathbf{X}$. The index set of the $\tilde{K}$ largest $J\left(\phi_{\bar{p}}\right)$ can be found as follows:

$$\mathbf{\Phi}_{\tilde{K}} = L_{\tilde{K}}\left(\mathbf{J}\right). \tag{6.7}$$

The true signal directions are located in the index set with a high probability. In other words, $\mathbf{\Phi}_{\tilde{K}}$ can be constructed as a new manifold matrix for the grid search.

6.2 Grid search with $\ell_{2,1}$-norm penalty

The areas whose correlations are near to zero can be ignored in the first state. Only the candidate areas are applied in the second stage to achieve the true direction with a high resolution. To

achieve a high resolution result, the $\mathbf{\Phi}_{\tilde{K}}$ found in the first stage is divided into denser sampling grids and construct a new matrix $\mathbf{\Phi}_s \triangleq [\mathbf{a}(\phi_{s1}), ..., \mathbf{a}(\phi_{sP_s})]$ ($M < P_s \ll P$). Thus, the problem can be modeled as

$$\mathbf{X} = \mathbf{\Phi}_s \mathbf{S} + \mathbf{W}. \tag{6.8}$$

As with Eq. (5.9), we can estimate the signal directions if we find the nonzero values in $\mathbf{S}$. This problem can be converted into the following $\ell_{2,1}$-norm minimization problem:

$$\min \|\mathbf{S}\|_{2,1}, \quad \text{s.t.} \, \|\mathbf{X} - \mathbf{\Phi}_s \mathbf{S}\|_F \leq \varepsilon, \tag{6.9}$$

where ε is a small constant, which can be set as $\varepsilon \geq \|\mathbf{W}\|_F$ [87, 90]. We can directly use a Matlab tool called CVX to solve the optimization problem in Eq. (6.9) [91].

Using the true signal $\mathbf{s}$, the constraint in Eq. (6.9) can be reconstructed as follows:

$$\begin{aligned} \|\mathbf{X} - \mathbf{\Phi}_s \mathbf{S}\|_F &= \|(\mathbf{AS} + \mathbf{W}) - \mathbf{\Phi}_s \mathbf{S}\|_F \\ &= \|(\mathbf{AS} - \mathbf{\Phi}_s \mathbf{S}) + \mathbf{W}\|_F \leq \varepsilon \end{aligned} \tag{6.10}$$

Using the triangle inequality ($|\|\mathbf{a}\|_F - \|\mathbf{b}\|_F| \leq \|\mathbf{a} \pm \mathbf{b}\|_F \leq \|\mathbf{a}\|_F + \|\mathbf{b}\|_F$) yields

$$|\|\mathbf{AS} - \mathbf{\Phi}_s \mathbf{S}\|_F - \|\mathbf{W}\||_F \leq \|(\mathbf{AS} - \mathbf{\Phi}_s \mathbf{S}) + \mathbf{W}\|_F \leq \varepsilon, \tag{6.11}$$

Considering the sign yields

$$-\varepsilon \leq \|\mathbf{AS} - \mathbf{\Phi}_s \mathbf{S}\|_F - \|\mathbf{W}\|_F \leq \varepsilon. \tag{6.12}$$

Since $\varepsilon \geq \|\mathbf{W}\|_F$, the above inequality can be simplified as follows:

$$0 \leq \|\mathbf{AS} - \mathbf{\Phi}_s \mathbf{S}\|_F \leq 2\varepsilon. \tag{6.13}$$

To achieve the max residual angle, we assume that there is only one nonzero value element in $\mathbf{s}$ and in $\mathbf{S}$. This yields

$$0 \leq \|\mathbf{AS} - \mathbf{\Phi}_s \mathbf{S}\|_F = \|\mathbf{a}(\theta)\mathbf{s}(\theta) - \mathbf{a}(\phi)\tilde{\mathbf{s}}(\phi)\|_F \leq 2\varepsilon, \tag{6.14}$$

where $\mathbf{s}(\theta)$ is the true signal from θ and $\tilde{\mathbf{s}}(\phi)$ is the estimated signal from ϕ. According to Eq.

(6.9), the solution satisfies the condition requiring that $f\left(\tilde{\mathbf{s}}(\phi)\right)$ be minimized, where

$$\begin{aligned} f\left(\tilde{\mathbf{s}}\left(\phi\right)\right) =& \left\|\mathbf{a}(\theta)\mathbf{s}(\theta)-\mathbf{a}(\phi)\tilde{\mathbf{s}}(\phi)\right\|_F^2 \\ =& \mathrm{tr}\left\{\left[\mathbf{a}(\theta)\mathbf{s}(\theta)-\mathbf{a}(\phi)\tilde{\mathbf{s}}(\phi)\right]^H\left[\mathbf{a}(\theta)\mathbf{s}(\theta)-\mathbf{a}(\phi)\tilde{\mathbf{s}}(\phi)\right]\right\} \\ =& \mathrm{tr}\left\{\mathbf{s}^H(\theta)\mathbf{a}^H(\theta)\mathbf{a}(\theta)\mathbf{s}(\theta)-2\mathbf{s}^H(\theta)\mathbf{a}^H(\theta)\mathbf{a}(\phi)\tilde{\mathbf{s}}(\phi)\right. \\ & \left.+\tilde{\mathbf{s}}^H(\phi)\mathbf{a}^H(\phi)\mathbf{a}(\phi)\tilde{\mathbf{s}}(\phi)\right\} \\ =& M\times\mathrm{tr}\left\{\mathbf{s}^H(\theta)\mathbf{s}(\theta)\right\} \\ & -2\times\mathrm{tr}\left\{\mathbf{s}^H(\theta)\mathbf{a}^H(\theta)\mathbf{a}(\phi)\tilde{\mathbf{s}}(\phi)\right\} \\ & +M\times\mathrm{tr}\left\{\tilde{\mathbf{s}}^H(\phi)\tilde{\mathbf{s}}(\phi)\right\}, \end{aligned} \tag{6.15}$$

where $\mathrm{tr}\left[\cdot\right]=\sum_{i}^{N}\left[\cdot\right]_{ii}$ denotes the trace of a matrix. For any θ and ϕ, the solution for $\min f\left(\tilde{\mathbf{s}}\left(\phi\right)\right)$ also satisfies

$$\frac{\partial f\left(\tilde{\mathbf{s}}\left(\phi\right)\right)}{\partial\tilde{\mathbf{s}}\left(\phi\right)}=0. \tag{6.16}$$

It yields

$$\frac{\partial f\left(\tilde{\mathbf{s}}\left(\phi\right)\right)}{\partial\tilde{\mathbf{s}}\left(\phi\right)}=M\tilde{\mathbf{s}}(\phi)-2\mathbf{s}^H(\theta)\mathbf{a}^H(\theta)\mathbf{a}(\phi)=0. \tag{6.17}$$

The relation between θ and ϕ can be achieved as

$$\frac{\partial f\left(\tilde{\mathbf{s}}\left(\phi\right)\right)}{\partial\tilde{\mathbf{s}}\left(\phi\right)}=2M\tilde{\mathbf{s}}(\phi)-2\mathbf{a}^H(\phi)\mathbf{a}(\theta)\mathbf{s}(\theta)=0. \tag{6.18}$$

This equation can be solved as

$$\tilde{\mathbf{s}}(\phi)=\frac{\mathbf{a}^H(\phi)\mathbf{a}(\theta)}{M}\mathbf{s}(\theta). \tag{6.19}$$

Inserting $\tilde{\mathbf{s}}(\phi)$ from (6.19) into (6.14) yields

$$\begin{aligned} &\left\|\mathbf{a}(\theta)\mathbf{s}(\theta)-\mathbf{a}(\phi)\tilde{\mathbf{s}}(\phi)\right\|_F \\ &=\left\|\mathbf{a}(\theta)\mathbf{s}(\theta)-\mathbf{a}(\phi)\frac{\mathbf{a}^H(\phi)\mathbf{a}(\theta)}{M}\mathbf{s}(\theta)\right\|_F\leq 2\varepsilon. \end{aligned} \tag{6.20}$$

Thus, we can achieve

$$\left\|\mathbf{a}(\theta)-\rho\mathbf{a}(\phi)\right\|_F\leq 2\upsilon, \tag{6.21}$$

where $\rho=\left[\mathbf{a}^H(\phi)\mathbf{a}(\theta)\right]/M=\chi\left(\phi,\theta\right)$ denotes the correlation and $\upsilon=\varepsilon/\left\|\mathbf{s}(\theta)\right\|_F\geq\left\|\mathbf{n}\right\|_F/\left\|\mathbf{s}(\theta)\right\|_F$ is related to the noise power required by the signal-to-noise ratio (SNR).

The proposed method is listed in Algorithm 5.

At last, the proposed method can also use the multi-resolution grid refinement method to further divide the regions where the sources may exist into a denser sampling grids to get a higher resolution.

Algorithm 5 Dimension-Reduced Direction-of-Arrival Estimation Based on $\ell_{2,1}$-Norm Penalty

Input:

The manifold matrix $\mathbf{\Phi}$,

The sampled signal $\mathbf{X}$,

The parameter $\varepsilon = \frac{\|\mathbf{N}\|_F}{P}$,

The number of possible search areas $\tilde{K}$,

Initialization:

1. Set $\mathbf{S}^0 = \mathbf{0}$,
2. Set $\mathbf{\Phi}_{\tilde{K}} = \emptyset$,

Pre-estimation:

3. $W(\phi_p) = \mathbf{a}^H(\phi_p)\,\mathbf{R_{xx}}\mathbf{a}(\phi_p)$,
4. $\mathbf{\Phi}_{\tilde{K}} = L_{\tilde{K}}(\mathbf{W})$,

Grid search:

5. Construct $\mathbf{\Phi}_s$ according to $\mathbf{\Phi}_{\tilde{K}}$,
6. $\mathbf{S} = \min \|\mathbf{S}\|_{2,1}$ s.t. $\|\mathbf{X} - \mathbf{\Phi}_s\mathbf{S}\|_F \leq \varepsilon$,

Output: The estimated DOA $\mathbf{S}^{\#} = \mathbf{S}$.

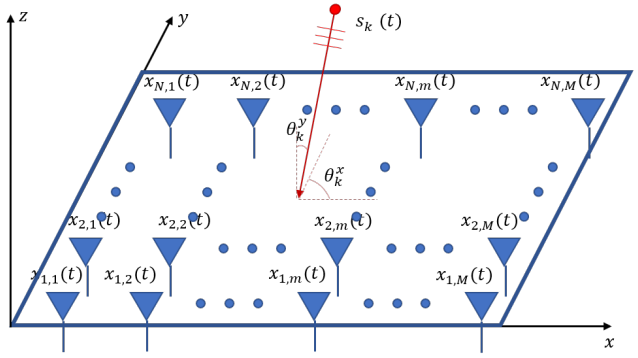


Figure 6.2: A reference scenario of a uniform plane array: a source, denoted as $s_k(t)$, impinges on the array at azimuth angle θ_k^x and elevation angle θ_k^y, with the received signals denoted as $x_{n,m}(t)$.

6.3 Extension for 2D-DOA

The proposed method can be extended for 2D-DOA estimation with a uniform plane array. Fig. 6.2 shows an example of a uniform plane array with $N \times M$ sensors. The signals come from (θ_1^x, θ_1^y), (θ_2^x, θ_2^y), ..., $\left(\theta_K^x, \theta_K^y\right)$, where θ_k^x and θ_k^y $(k = 1, ..., K)$ denote the azimuth angle and elevation angle of the k-th source, respectively. Thus, the wave path difference between the first sensor and the i-th sensor can be written as

$$\beta = 2\pi \left(d_{x_i} \cos\theta_k^y \sin\theta_k^x + d_{y_i} \sin\theta_k^y \sin\theta_k^x\right)/\ell, \tag{6.22}$$

where d_{x_i} and d_{y_i} denote the coordinates of the i-th sensor. The received signal can be written as

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{A}_x D_1(\mathbf{A}_y) \\ \mathbf{A}_x D_2(\mathbf{A}_y) \\ \vdots \\ \mathbf{A}_x D_N(\mathbf{A}_y) \end{bmatrix} \mathbf{s}(t) + \mathbf{w}(t), \tag{6.23}$$

where

$$\mathbf{A}_x \triangleq \left[\mathbf{a}_x(\theta_1^x, \theta_1^y), \mathbf{a}_x(\theta_2^x, \theta_2^y), ..., \mathbf{a}_x(\theta_K^x, \theta_K^y)\right],$$

$$\mathbf{a}_x(\theta_k^x, \theta_k^y) \triangleq \left[a_x^1(\theta_k^x, \theta_k^y), ..., a_x^m(\theta_k^x, \theta_k^y), ..., a_x^M(\theta_k^x, \theta_k^y)\right]^T,$$

$$a_x^m(\theta_k^x, \theta_k^y) \triangleq \exp\left(-j2\pi d_{x_m} \cos\theta_k^y \sin\theta_k^x\right)/\ell,$$

$$\mathbf{A}_y \triangleq \left[\mathbf{a}_y(\theta_1^x, \theta_1^y), \mathbf{a}_y(\theta_2^x, \theta_2^y), ..., \mathbf{a}_y(\theta_K^x, \theta_K^y)\right],$$

$$\mathbf{a}_y(\theta_k^x, \theta_k^y) \triangleq \left[a_y^1(\theta_k^x, \theta_k^y), ..., a_y^m(\theta_k^x, \theta_k^y), ..., a_y^M(\theta_k^x, \theta_k^y)\right]^T,$$

$$a_y^n(\theta_k^x, \theta_k^y) \triangleq \exp\left(-j2\pi d_{y_n} \sin\theta_k^y \sin\theta_k^x\right)/\ell.$$

$D_n(\mathbf{A}_y)$ denotes the diagonal matrix constructed by the n-th row of $\mathbf{A}_y$. It can be found that Eq. (6.23) is similar to Eq. (5.9). Our proposed method can be applied for 2D-DOA estimation.

First, the search range is divided into $\bar{P} \times \bar{Q}$ sampling grids $(\varphi_{\bar{p}}^x, \varphi_{\bar{q}}^y)$, as shown in Fig. 6.3, where

$$\mathbf{a}(\bar{p}, \bar{q}) = \mathbf{a}_y(\varphi_{\bar{p}}^x, \varphi_{\bar{q}}^y) \otimes \mathbf{a}_m(\varphi_{\bar{p}}^x, \varphi_{\bar{q}}^y). \tag{6.24}$$

Then, the correlation between $\mathbf{a}(\bar{p}, \bar{q})$ and $\mathbf{x}$ is calculated as

$$\mathbf{J}(\bar{p}, \bar{q}) = \mathbf{a}^H(\bar{p}, \bar{q}) \mathbf{R}_{\mathbf{xx}} \mathbf{a}(\bar{p}, \bar{q}). \tag{6.25}$$

The index set of the $\tilde{K}$ largest $\mathbf{J}(\bar{p}, \bar{q})$ can be found as follows:

$$\mathbf{\Phi}_{\tilde{K}} = L_{\tilde{K}}(\mathbf{W}). \tag{6.26}$$

The true signal directions are located in the index set with a high probability. Then, the new search areas are divided into denser sampling grids to construct the measurement matrix $\mathbf{\Phi}_s$. Finally, the nonzero values in $\mathbf{s}$ can be solved by Eq. (6.9).

6.4 Simulation Results

The performance of our proposed method is compared with the Capon [12], MUSIC [11] the ℓ_1-SVD [56], JLZA-DOA [10] and NSW-L1-L2 [57] algorithms. The simulations are performed

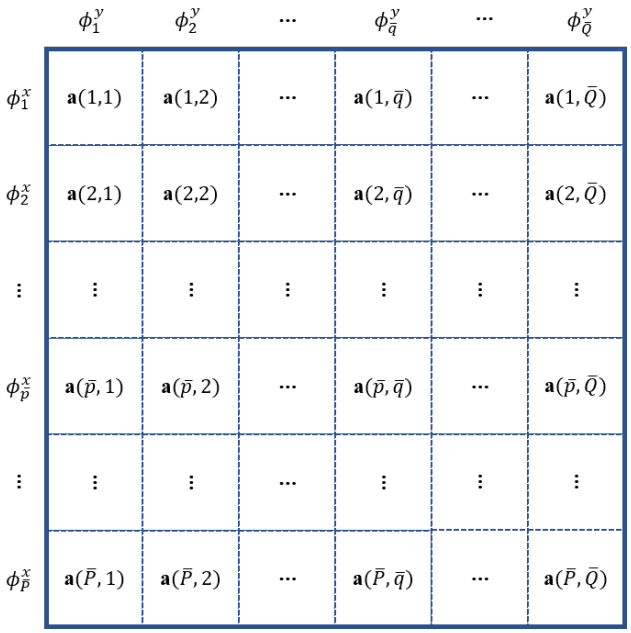


Figure 6.3: Sampling grids in Stage 1 for 2D-DOA estimation.

utilizing MATLAB R2017b with an Intel Xeon E3-1270 v5, 3.60 GHz processor and 16 GB of memory under Microsoft Windows 10 Professional (64 bit).

We consider a ULA with $M = 10$ sensors at $L = 5$ snapshots. The distance between the sensors is equal to half the wavelength. In the ℓ_1-SVD, JLZA-DOA and NSW-L1-L2 algorithms, a sampling grid with a range of 0° to 180° and 1° intervals is used. In our proposed method, a sampling grid with intervals of 10° is used for pre-estimation in the first stage. Then, denser sampling grids with an interval of 1° are set around the estimated directions for the grid search.

In the first simulation, there are $K = 3$ uncorrelated signals originating from $[40°, 110°, 120°]$. The true signal directions are on the sampling grid. The SNR is set to 10 dB. Fig. 6.4 (a) and (b) show the DOA estimation results obtained by the MUSIC and Capon algorithms. Since the number of snapshots is too small, these two algorithms fail to resolve the sources at 40° and 120°. Fig. 6.4 (c) (d) and (e) show the CS-based DOA estimation results obtained by ℓ_1-SVD, JLZA-DOA and NSW-L1-L2 algorithms. Fig. 6.4 (f) shows the $|W(\phi_p)|$ in each search area in the first stage of our proposed method. The output of the correlation filter shows that the sources come from search area 1 ($30° \sim 50°$) and search area 2 ($100° \sim 130°$). Thus, the other search areas can be ignored in the subsequent search. Fig. 6.4 (g) shows the final result of our proposed method. Clearly, all the source directions can be searched by the CS-based DOA estimation methods.

In the second simulation, there are $K = 3$ uncorrelated signals originating from $[40.5°, 110.8°, 120.5°]$. The true signal directions are not on the sampling grid. The other parameters are the same

as those used in the first simulation. The results are shown in Fig. 6.5. All the CS-based DOA estimation methods can locate three sources closest to the true signal directions, while the traditional methods are failed.

Table 1 shows the calculation time of the ℓ_1-SVD, JLZA-DOA, NSW-L1-L2 and our proposed method. The average times obtained after 100 Monte Carlo trials. The other parameters are the same as those used in the second simulation. It can be found that our proposed method is the fastest, because it ignores some search areas and focus on the most possible search areas.

Table 6.1: Calculation time of each CS-based algorithm

Parameter		Time (sec)			
M	L	ℓ_1-SVD	JLZA-DOA	NSW-L1-L2	Proposed method
10	5	3.4375	3.7344	5.8594	1.5000

In the third simulation, the relation between the frequency of detection and the number of snapshots is considered. The number of sensors is set to $M = 10$. The SNR is fixed at 10 dB. The frequency of detection is defined as the ratio between the number of the successful directions and the number of the total directions. The number of snapshots is set to $L \in \{1, 2, ..., 10\}$. There are $K = 3$ uncorrelated signals originating from $[40°, 85°, 95°]$. The results are shown in Fig. 6.6. When $L \leq 10$, the MUSIC, Capon and ℓ_1-SVD algorithms cannot obtain the true signal directions with a high frequency. $\ell_{2,1}$-norm, NSW-L1-L2 and our proposed method perform well and can achieve the true signal directions with a high probability. Particularly, our proposed method can obtain the true signal directions at a higher frequency compared to the NSW-L1-L2 algorithm. When $L = 3$, the frequency of direction of our proposed method is close to 98%, while the frequency of direction of the NSW-L1-L2 algorithm is close to 85%.

In the fourth simulation, the relation between the frequency of detection and the SNR is considered. The number of sensors is set to $M = 10$. The number of snapshots is fixed at $L = 10$. We set SNR $\in \{0, 2, 4, 6, 8, 10\}$ dB. There are $K = 3$ uncorrelated signals originating from $[40°, 85°, 95°]$. The results are shown in Fig. 6.7. When SNR ≤ 10 dB, the MUSIC, Capon ℓ_1-SVD and JLZA-DOA algorithms cannot obtain the true signal directions with a high frequency. NSW-L1-L2 and our proposed method perform well and can obtain the true signal directions with a high probability. Particularly, our proposed method can obtain the true signal directions at a higher frequency compared to the NSW-L1-L2 algorithm. When SNR $= 2$ dB, the frequency of direction of our proposed method is close to 100%, while the frequency of

direction of the NSW-L1-L2 algorithm is is only 80%.

In the fifth simulation, the relation between the frequency of detection and the number of sensors is considered. The number of snapshots is fixed at $L = 10$. The SNR is set at 10 dB. The number of sensors is set to $M \in \{6, 8, 10, 12, 14\}$. There are $K = 3$ uncorrelated signals originating from $[40°, 85°, 95°]$. The results are shown in Fig. 6.8. When $M > L = 10$, the Capon algorithm fails because the covariance matrix of the received signal is not a full rank matrix, namely, the inverse of the covariance matrix of the received signal cannot be achieved. The MUSIC, ℓ_1-SVD and JLZA-DOA algorithms also cannot obtain the true signal directions with a high frequency. Again, $\ell_{2,1}$-norm, NSW-L1-L2 and our proposed method perform well and can obtain the true signal directions with a high probability. Particularly, our proposed method can obtain the true signal directions at a higher frequency of detection compared to the NSW-L1-L2 algorithm.

In the sixth simulation, the sensitivity to $\tilde{K}$ is considered. In general, the number of true sources is usually unknown and needs to be estimated. Some source enumeration methods, such as the minimum description length (MDL) method [92], the smoothed rand profile (SRP) method [93], the gershgorin disks methd [94] and the cross-correlation transformation (CCT) method [95] can be used to estimate the number of true signal sources. However, the performances of these methods worsens when the number of snapshots is small and the SNR is low. Thus, the sensitivity of our proposed method to $\tilde{K}$ must be considered. The results are shown in Fig. 6.9. There are 3 uncorrelated sources originating from [40°, 85°, 95°]. $M = 10$, $L = 10$ and SNR=10 dB. It can be found that although $\tilde{K}$ is set to a value larger than that of K, the estimated results are not affected. Hence, our proposed method is not very sensitive to $\tilde{K}$. However, the calculation time will increase as $\tilde{K}$ increases, as shown in Table 2.

Table 6.2: Calculation time of the proposed method

$\tilde{K}$	3	4	5	6	7	8
Time (sec)	1.722	1.915	2.150	2.445	2.735	3.157

Finally, the proposed method is applied for 2D-DOA estimation by a uniform plane array with 10×10 sensors and $L = 10$ snapshots. The distance between the sensors is set as $d_x = d_y = \ell/2$. Sampling grids with a range from 0° to 90° with a 5° interval are used in the first stage. Then, the selected areas are divided into denser sampling grids with an interval of 1° in the second stage. There are $K = 2$ uncorrelated signals originating from (30°,60°) and (60°,70°). The SNR is 10 dB. Only the $\tilde{K} = 7$ largest elements in matrix $\mathbf{J}(p, q)$ is selected for the grid search. The

results are shown in Fig. 6.10. It can be found that the area in which the true sources exist can be pre-estimated in the first stage. Thus, the dimensions of the measurement matrix are reduced, as only the selected areas are used for the grid search. In the second stage, the signal sources can be found.

At last, the pre-estimation can be used before the ℓ_1-SVD, JLZA-DOA and NSW-L1-L2 algorithms to quickly identify the candidate or potential areas where true sources may exist. The pre-estimation is able to reduce the calculating time without the performance degradation. The simulation results are shown in Fig. 6.11, Fig. 6.12 and Fig. 6.13. We set $K = 3$ uncorrelated signals originating from $[40.5°, 95.0°, 125.5°]$, $L = 10$ and SNR $= 20$ dB.

Fig. 6.11 (a) shows the results solved by the NSW-L1-L2 algorithm without pre-estimation. We start with a sampling grid whose interval is 1° and obtain an approximate source direction. Then we make the sampling grid finer (0.1° interval) around the approximate source direction and refine the estimates. In each stage, the NSW-L1-L2 algorithm is used. Fig. 6.11 (b) shows the results solved by the NSW-L1-L2 algorithm with pre-estimation. We start with a sampling grid whose interval is 10° and obtain an approximate source direction by CBF. Then we refine the sampling grid as 1° interval in the second stage and 0.1° interval in the third stage. The NSW-L1-L2 algorithm is only used in the last two stages. It can be found that both the two methods obtain the true source directions. Fig. 6.11 (c) shows the calculation time of these two methods. The NSW-L1-L2 algorithm without pre-estimation takes 8.8692s to obtain the true source direction. Especially, the first stage takes 7.3003s. The NSW-L1-L2 algorithm with pre-estimation only takes 3.3242s to obtain the true source direction. CBF takes 0.0020s to identify the source areas. It results in the decreasing of the calculation time to identify the source direction by 1° grid resolution.

Fig. 6.12 (a) shows the results solved by the JLZA-DOA algorithm without pre-estimation. As the same as the previous simulation, we search the true source directions using 1° grid resolution first. Then 0.1° grid resolution is used around the approximate source direction. In each stage, the JLZA-DOA algorithm is used. Fig. 6.12 (b) shows the results solved by the JLZA-DOA algorithm with pre-estimation. We start with 10° grid resolution and obtain an approximate source direction by CBF. Then 1° grid resolution is used around the approximate source direction in the second stage and 0.1° grid resolution is used in the third stage. The JLZA-DOA algorithm is only used in the last two stages. It can be found that both the two methods obtain the true source directions. Fig. 6.12 (c) shows the calculation time of these two methods. The JLZA-DOA algorithm without pre-estimation takes 4.8531s to obtain the true source direction. Especially, the first stage takes 4.1933s. The JLZA-DOA algorithm with

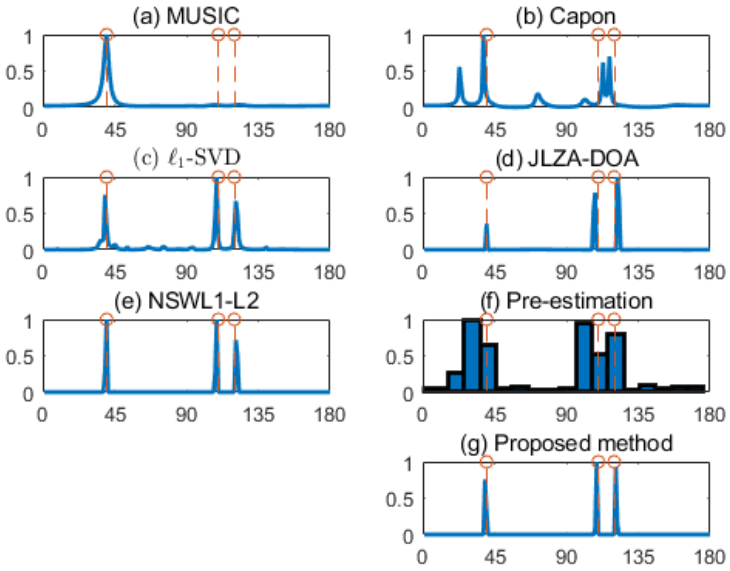


Figure 6.4: DOA estimation results estimated for three sources at 40°, 110° and 120° utilizing 10 sensors with 5 snapshots in an SNR=10 dB environment. Blue lines denote the DOA estimation results, and red dotted lines with round markers denote the true signal directions.(a) MUSIC. (b) Capon. (c) ℓ_1-SVD. (d) JLZA-DOA, (e) NSW-L1-L2. (f) Pre-estimation (the first stage of our proposed method). (g) The final result of our proposed method.

pre-estimation only takes 1.2425s to obtain the true source direction. CBF takes 0.0013s to identify the source areas. It results in the decreasing of the calculation time to identify the source direction by 1° grid resolution.

Fig. 6.13 (a) shows the results solved by the ℓ_1-SVD algorithm without pre-estimation. As the same as the previous experiment, we search the true source directions using 1° grid resolution first. Then 0.1° grid resolution is used around the approximate source direction. In each stage, the ℓ_1-SVD algorithm is used. Fig. 6.12 (b) shows the results solved by the ℓ_1-SVD algorithm with pre-estimation. We start with 10° grid resolution and obtain an approximate source direction by CBF. Then 1° grid resolution is used around the approximate source direction in the second stage and 0.1° grid resolution is used in the third stage. The ℓ_1-SVD algorithm is only used in the last two stages. It can be found that both the two methods obtain the true source directions. Fig. 6.12 (c) shows the calculation time of these two methods. The ℓ_1-SVD algorithm without pre-estimation takes 5.7136s to obtain the true source direction. Especially, the first stage takes 4.1048s. The ℓ_1-SVD algorithm with pre-estimation only takes 3.4813s to obtain the true source direction. CBF takes 0.0027s to identify the source areas. It results in the decreasing of the calculation time to identify the source direction by 1° grid resolution.

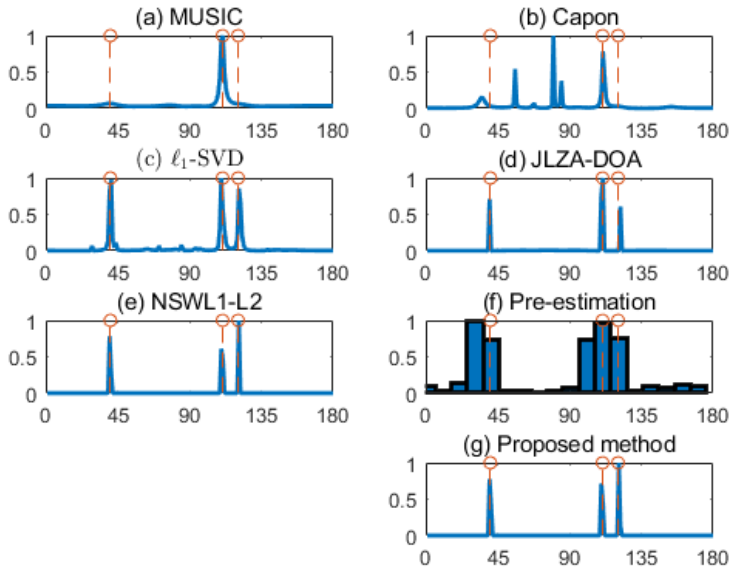


Figure 6.5: DOA estimation results estimated for three sources at 40.5°, 110.8° and 120.5° utilizing 10 sensors with 5 snapshots in the SNR=10 dB environment. Blue lines denote the DOA estimation results, and red dotted lines with round markers denote the true signal directions. (a) MUSIC. (b) Capon. (c) ℓ_1-SVD. (d) JLZA-DOA, (e) NSW-L1-L2. (f) Pre-estimation (the first stage of our proposed method). (g) The final result of our proposed method.

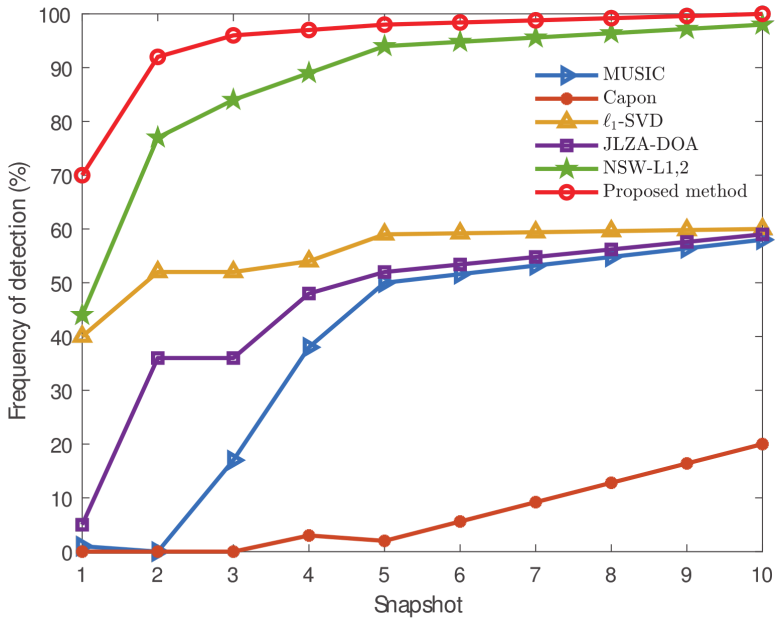


Figure 6.6: Frequency of direction against the number of snapshots, where $M = 10$ and SNR= 10 dB. There are $K = 3$ signal sources at $\theta_1 = 40°$, $\theta_2 = 85°$ and $\theta_3 = 95°$.

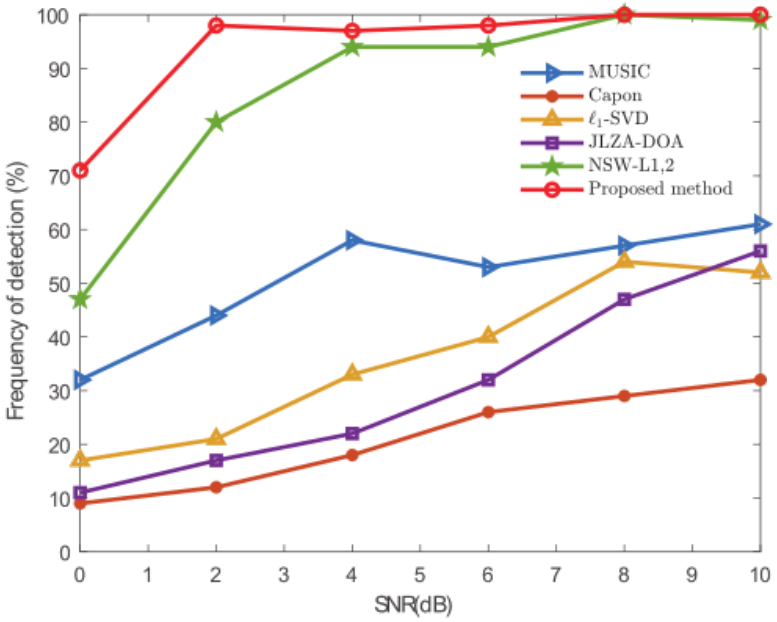


Figure 6.7: Frequency of direction against the SNR, where $M = 10$ and $L = 10$. There are $K = 3$ sources at $\theta_1 = 40°$, $\theta_2 = 85°$ and $\theta_3 = 95°$.

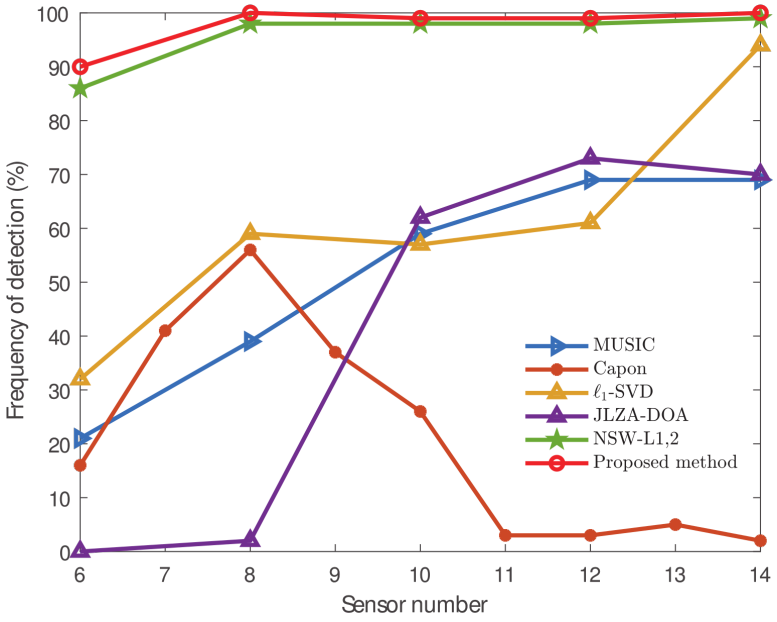


Figure 6.8: Frequency of direction against the number of sensors, where SNR= 10 dB and $L = 10$. There are $K = 3$ signal sources at $\theta_1 = 40°$, $\theta_2 = 85°$ and $\theta_3 = 95°$.

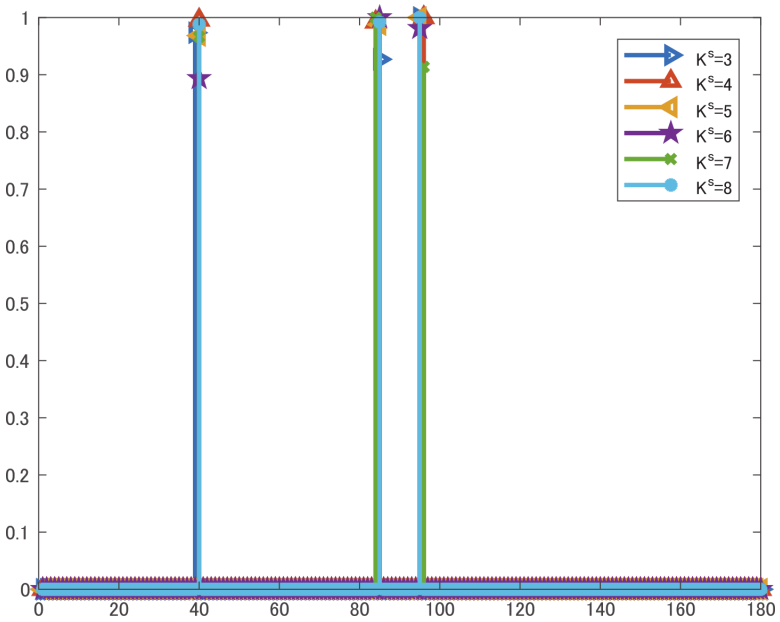


Figure 6.9: Results against $\tilde{K}$, where SNR= 10 dB, $L = 10$ and $M = 10$. There are $K = 3$ sources at $\theta_1 = 40°$, $\theta_2 = 85°$ and $\theta_3 = 95°$.

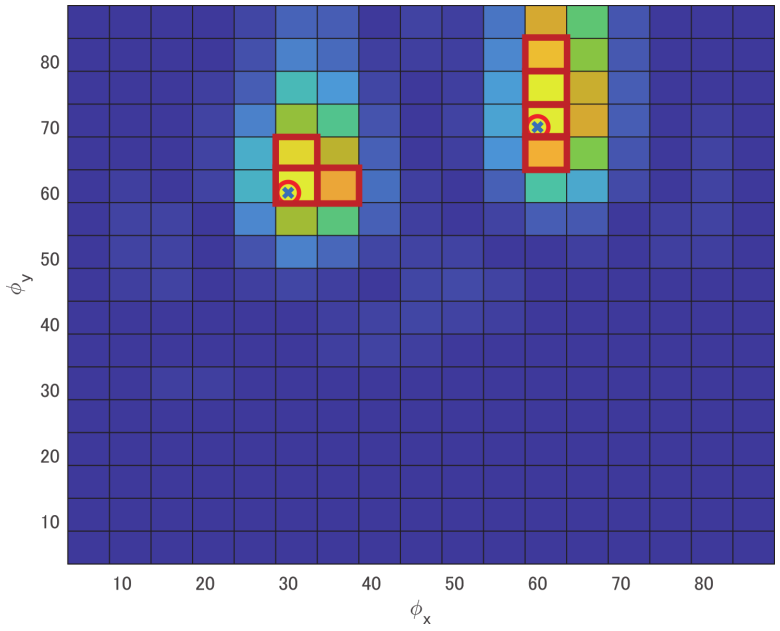


Figure 6.10: 2D-DOA estimation result for two signal sources at (30°,60°) and (60°,70°) utilizing 10×10 uniform plane array with 10 snapshots in the SNR=10 dB environment. Blue cross symbol denotes the true source. Red round symbol denotes the estimated source. Colored grids correspond to $\mathbf{J}(p, q)$. A yellow grid indicates a large value, while a cyan grid indicates a small value. The red box denotes the selected possible area for grid search.

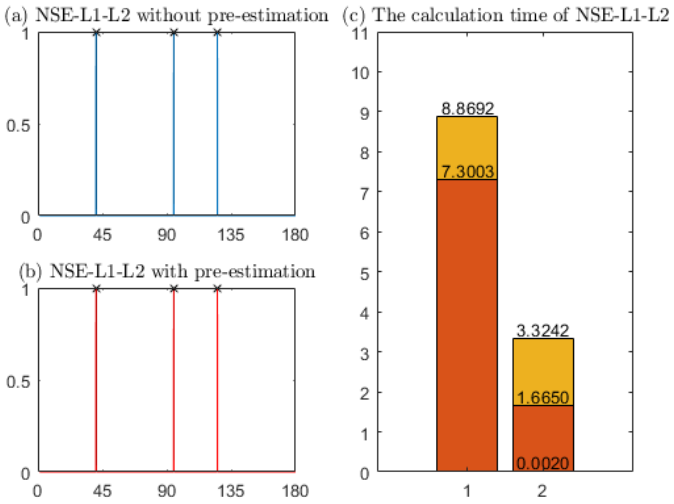


Figure 6.11: DOA estimation results estimated for three sources at $[40.5°, 95.0°, 125.5°]$ with 10 snapshots. (a) the NSW-L1-L2 algorithm without pre-estimation. (b) the NSW-L1-L2 algorithm algorithm with pre-estimation. (c) the calculation time.

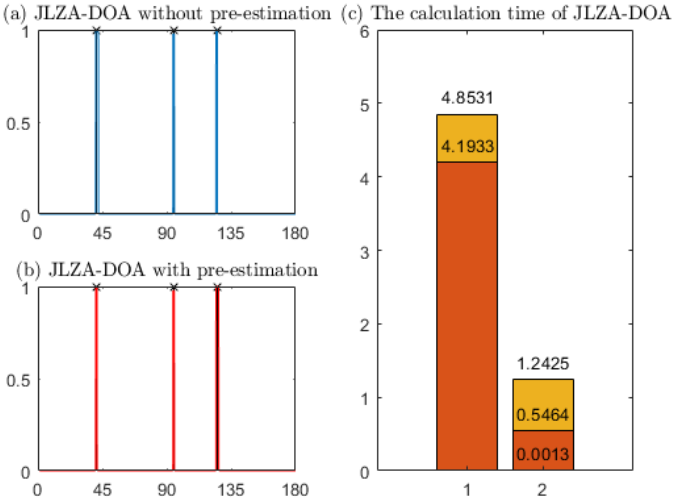


Figure 6.12: DOA estimation results estimated for three sources at $[40.5°, 95.0°, 125.5°]$ with 10 snapshots. (a) the JLZA-DOA algorithm without pre-estimation. (b) the JLZA-DOA algorithm with pre-estimation. (c) the calculation time.

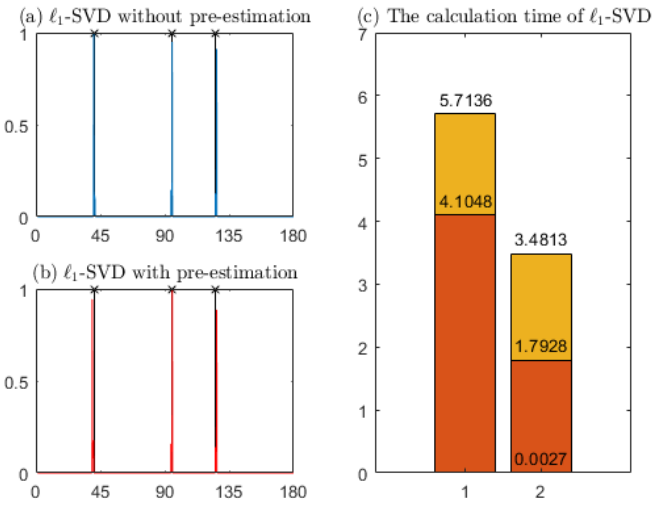


Figure 6.13: DOA estimation results estimated for three sources at $[40.5°, 95.0°, 125.5°]$ with 10 snapshots. (a) the ℓ_1-SVD algorithm without pre-estimation. (b) the ℓ_1-SVD algorithm with pre-estimation. (c) the calculation time

6.5 Conclusions

In this paper, we propose the dimension-reduced DOA estimation method for the case of a few snapshots based on CBF to focus on the most possible search areas. CBF is used to select the search areas in which the true signal directions exist for a coarse sampling grids because of its property of low-resolution. Then, it can decrease the calculation time for the grid search in the second stage. In addition, we extend our proposed method from 1D-DOA estimation to 2D-DOA estimation. The simulation results show that our proposed method has a higher frequency of direction and requires less calculation time compared to the existing DOA estimation algorithms. Furthermore, our proposed method is not sensitive to the parameter $\tilde{K}$. Hence, it can work even when $\tilde{K}$ is larger than the number of true sources. In addition, the simulation results verify the effectiveness of our proposed method for 2D-DOA estimation.

6.6 Proof of the position of the first null point

In this appendix, we derive the position of the first null point. First, the null point has the following properties:

$$|\chi(\phi,\theta)| = \left|\frac{\mathbf{a}^H(\phi)\mathbf{a}(\theta)}{M}\right| = 0, \tag{6.6.1}$$

where

$$\mathbf{a}^H(\phi)\mathbf{a}(\theta) = \sum_{m=1}^{M} \exp\left\{j2\pi(m-1)d\left[\cos(\phi)-\cos(\theta)\right]/\ell\right\}, \tag{6.6.2}$$

with m denoting the m-th sensor and d denoting the distance between two adjacent sensors. We assume that $d/\ell = 1/2$. Thus, 6.6.1 can be written as

$$\left| \sum_{m=1}^{M} \exp\left\{ j\pi(m-1)\left[\cos\left(\phi\right) - \cos\left(\theta\right)\right] \right\} \right| = 0. \tag{6.6.3}$$

We set $x = \cos\left(\phi\right) - \cos\left(\theta\right)$; thus, equation 6.6.3 can be simplified to

$$\left| \sum_{m=1}^{M} \exp\left\{ j\pi(m-1)x \right\} \right| = \left| \sum_{m=1}^{M} \left(\exp\left\{ j\pi x \right\}\right)^{m-1} \right| = 0. \tag{6.6.4}$$

$\left(\exp\left\{ j\pi x \right\}\right)^{m-1}$ is a geometric progression. Hence, the sum of $\left(\exp\left\{ j\pi x \right\}\right)^{m-1}$ can be written as follows:

$$\frac{1 - \left(\exp\left\{ j\pi x \right\}\right)^{M}}{1 - \exp\left\{ j\pi x \right\}} = 0. \tag{6.6.5}$$

Obviously, $\exp\left\{ j\pi x \right\} \neq 1$. Hence, the solution satisfies $\left(\exp\left\{ j\pi x \right\}\right)^{M} = 1$, namely,

$$x = \frac{2i}{M}, \quad \text{for } M > 1, i = \pm 1, \pm 2, \pm 3, ... \tag{6.6.6}$$

The position of the first null point closet to θ is given as follows:

$$\phi_{null} = \arccos\left[\cos\left(\theta\right) \pm \frac{2}{M} \right]. \tag{6.6.7}$$

Chapter 7

Adaptive filtering algorithm for DOA estimation with small snapshots

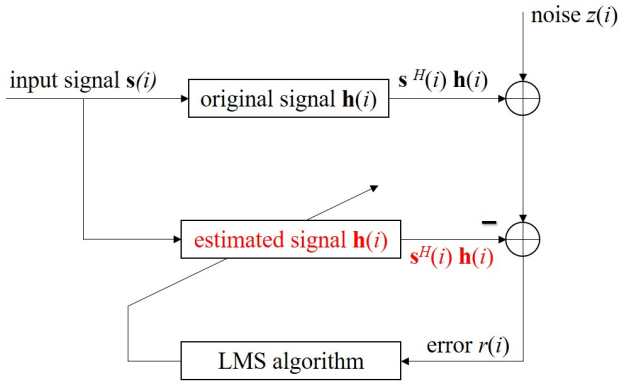
In the proceeding chapter, it has been shown that the CBF method can be used to find the search areas in which the true signal sources exist for a coarse sampling grids. And only the selected areas are used for grid search with high-resolution by $\ell_{2,1}$-norm penalty. It can reduce complexity effectively. However, the grid search processing still has a high resolution because the $\ell_{2,1}$-minimization based on the CS requires the matrix inversion operation in each iteration. This chapter is to further study the low-complexity CS algorithm for DOA estimation with small snapshots. In this chapter, we will propose a new algorithm to solve the $\ell_{2,1}$-minimization problem without the matrix inversion. Thus, the complexity can be further reduced by our proposed adaptive filter based CS algorithm.

The adaptive filter [96–98] has been widely used in signal processing for system identification due to its simplicity. In each iteration, only vector operation is used to adjust the weight parameters in the filter under an specific constraint. Motived by the adaptive filter, a novel adaptive filtering algorithm is proposed in this chapter for DOA estimation with small snapshots. The proposed algorithm is simple because it is based on the adaptive filter, and it is high accuracy because it utilizes the joint sparse characteristic of the CS-based DOA estimation model.

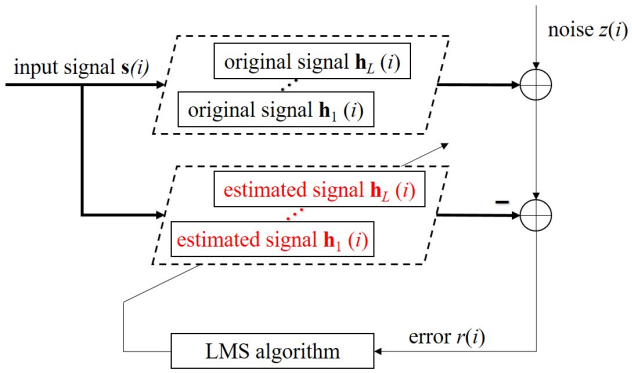
7.1 Adaptive filter framework for DOA estimation

7.1.1 Adaptive filter

The adaptive filtering algorithms have been widely used due to its good performance, low complexity, and reliable robustness. Fig. 7.1 (a) shows the diagram of the adaptive filter. Least mean squares (LMS) algorithm is a classical adaptive filtering algorithm [99]. It is used to find



(a)



(b)

Figure 7.1: The diagram of the adaptive filter (a) and matrix adaptive filter (b).

the the original signal that relates to producing the least mean square of the error signal between the input and output of a filter. The residual error signal $r(i)$ of the LMS algorithm for complex data is denoted by

$$r(i) = d(i) - \mathbf{s}^H(i)\mathbf{h}(i), \tag{7.1.1}$$

where $\mathbf{h}(i) = [h_1(i), ..., h_N(i)]^T$ is the original signal, which is often referred to as the filter coefficient in the adaptive filter, $\mathbf{s}(i) = [s(i), s(i-1), ..., s(i-N+1)]^T$ is the input signal, i is the iteration number, and N is the filter length. The output signal is constructed by $d(i) = \mathbf{s}^H(i)\mathbf{h}(i) + z(i)$, where $z(i)$ is the noise. By minimizing the residual error $r(i)$, $\mathbf{h}(i)$ contaminated by the noise $z(i)$ can be reconstructed iteratively by the LMS algorithm as:

$$\mathbf{h}(i+1) = \mathbf{h}(i) + \mu r(i)\mathbf{s}(i), \tag{7.1.2}$$

where μ is the step-size.

Recalling Eq. (5.8), the DOA estimation with a single snapshot can be formulated as an SMV problem. A link between the adaptive filter and the SMV problem has been proposed in [41]. The correspondences are listed in Table. 7.1.

Table 7.1: Correspondences between variables in DOA estimation problem with single snapshot (SMV problem) and adaptive filter

SMV ($t = 1$)	adaptive filter
$\mathbf{\Phi}_m$ (the mth row in matrix $\mathbf{\Phi}$) $m \in \{1, ..., M\}$	$\mathbf{s}^H(i)$
$\mathbf{x}(t)$	$\mathbf{h}(i)$
$y_m(t) = \mathbf{\Phi}_m \mathbf{x}(t) + \bar{e}_m(t)$	$d(i) = \mathbf{s}^H(i)\mathbf{h}(i) + z(i)$
$m = \mathrm{mod}(i, M) + 1$	iteration number i

In general, small snapshots can be used for DOA estimation in MIMO systems and achieve a better estimation performance than a single snapshot. Since $\mathbf{Y}$ and $\mathbf{S}$ are matrices, the adaptive filtering algorithm of [41] cannot be directly used for the DOA estimation with small snapshots.

In this paper, a matrix adaptive filter is used to extend the adaptive filter from SMV (original signal $\mathbf{x}(t)$ is a vector, where $t = L = 1$) to MMV (original signal $\mathbf{X}$ is a matrix, where $t \in [1, L], L > 1$). For the matrix adaptive filter, the source signal $\mathbf{x}(t)$ at each time t are treated as an independent original signal $\mathbf{h}_t(i)$. At different time t, the input signal $\mathbf{s}^H(i)$, namely $\mathbf{\Phi}_m$ is consistent. The diagram of the matrix adaptive filter is shown in Fig. 7.1 (b). The cost function of the LMS algorithm for the matrix filter can be defined as:

$$F(i) = \frac{1}{2}\left[y_m(t) - \mathbf{s}^H(i)\,\mathbf{h}_t(i)\right]^2. \tag{7.1.3}$$

By minimizing Eq. (7.1.3), the gradient descent recursion of $\mathbf{x}(i, t)$ can be written as

$$\begin{aligned} \mathbf{h}_t(i+1) &= \mathbf{h}_t(i) + \mu \frac{\partial F(i)}{\partial \mathbf{h}_t(i)} \\ &= \mathbf{h}_t(i) + \mu r(i)\,\mathbf{s}(i), \end{aligned} \tag{7.1.4}$$

where $r(i) = y_m(t) - \mathbf{s}^H(i)\,\mathbf{h}_t(i)$ is the residual error. Please note that although Eq. (7.1.4) cannot achieve a sparse solution directly. It is written here to show the framework of DOA estimation with small snapshots. Its improved form for solving the sparse solution will be described in detail in the following subsection.

Recalling Eq. (5.9), the DOA estimation with small snapshots can be formulated as an MMV problem. Thus, we list the link between the matrix adaptive filter and the MMV problem with small snapshots in Table. 7.2.

When the matrix adaptive filter is used to solve the MMV problem, there may not be

Table 7.2: Correspondences between variables in DOA estimation problem with small snapshots (MMV problem) and matrix adaptive filter

MMV ($t \in \{1, ..., L\}$)	matrix adaptive filter
$\mathbf{\Phi}_m$ (the mth row in matrix $\mathbf{\Phi}$) $m \in \{1, ..., M\}$	$\mathbf{s}^H(i)$
$\mathbf{x}(t)$	$\mathbf{h}_t(i)$
$y_m(t) = \mathbf{\Phi}_m \mathbf{x}(t) + \bar{e}_m(t)$	$d(i) = \mathbf{s}^H(i)\mathbf{h}_t(i) + z(i)$
$m = \mathrm{mod}(i, M) + 1$	iteration number i
$t = \lfloor i/M \rfloor + 1$	

enough data in $\mathbf{\Phi}$ for updating the signal $\mathbf{x}(t)$ to steady state. Hence, each row in $\mathbf{\Phi}$ and the corresponding elements $y_m(t)$ are utilized recursively.

7.1.2 Mixed norm penalty

The adaptive filtering algorithm cannot directly generate a sparse solution for Eq. (7.1.4). In fact, adding a suitable sparse penalty, such as ℓ_0-norm, into the cost function of the LMS algorithm can attract most entries in the solution to zero (see [17, 41, 98]). Note that both the real and imaginary parts of the matrix $\mathbf{X}$ are not just a sparse matrix, all its non-zero entries are concentrated in a few rows. Therefore, $\mathbf{X}$ exhibits joint sparse characteristic. In order to take the advantage of this characteristic, we propose a new mixed norm ($\ell_{2,0}$-norm) penalty for the adaptive filtering algorithm based on ℓ_0-norm penalty which has been used in the LMS algorithm [41, 98]. The $\ell_{2,0}$-norm is the number of the non-zero rows. $\ell_{2,0}$-minimizing can find a joint sparse solution which has the smallest number of rows that contain nonzero entries.

Denoting the solution at i iteration as $\mathbf{X}_i = [\mathbf{x}(i,1), ...\mathbf{x}(i,L)]$, an ideal $\ell_{2,0}$-norm of $\mathbf{X}_i$ can be written as [100]:

$$\|\mathbf{X}_i\|_{2,0} = \sum_{n=1}^{N} \mathcal{I}\left(\left\|\mathbf{X}_i^R(n,:)\right\|_2^2\right) + j\sum_{n=1}^{N} \mathcal{I}\left(\left\|\mathbf{X}_i^I(n,:)\right\|_2^2\right), \tag{7.1.5}$$

where $\mathbf{X}_i^R(n,:)$ and $\mathbf{X}_i^I(n,:)$ denote the real and imaginary parts of the nth row of $\mathbf{X}_i$, and the indicator function $\mathcal{I}$ is given by [10]

$$\mathcal{I}(x) = \begin{cases} 0, & x = 0 \\ 1, & \text{otherwise.} \end{cases} \tag{7.1.6}$$

As the indicator function $\mathcal{I}(x)$ is not smooth, we choose a popular approximation function [41, 101] to replace it:

$$\mathcal{I}(x) \approx 1 - e^{-\alpha|x|}. \tag{7.1.7}$$

The values of Eqs (7.1.6) and (7.1.7) will approach closer while $\alpha \to \infty$. Thus, $\|\mathbf{X}_i\|_{2,0}$ can be written as

$$\|\mathbf{X}_i\|_{2,0} \approx \sum_{n=1}^{N} 1 - e^{-\alpha \left\|\mathbf{X}_i^R(n,:)\right\|_2^2} + j \sum_{n=1}^{N} 1 - e^{-\alpha \left\|\mathbf{X}_i^I(n,:)\right\|_2^2}. \tag{7.1.8}$$

Incorporating Eq. (7.1.8) into Eq. (7.1.3), a new cost function can be given:

$$F_{new}(i) = \underbrace{\frac{1}{2}[y_m(t) - \Phi_m \mathbf{x}(i,t)]^2}_{\text{term1}} + \beta \underbrace{\|\mathbf{X}_i\|_{2,0}}_{\text{term2}}, \tag{7.1.9}$$

where term 1 is the mean square error term, term 2 is the sparse penalty term, β is the regularization parameter which trades off terms 1 and 2. By minimizing Eq. (7.1.9), the gradient descent recursion of $\mathbf{x}(i,t)$ can be rewritten as

$$\begin{aligned} x_n(i+1,t) &= x_n(i,t) + \mu \frac{\partial F_{new}(i)}{\partial \mathbf{x}(i,t)} \\ &= x_n(i,t) + \mu r(i) a_m^* \left(\bar{\theta}_n\right) \\ &- \kappa \left(e^{-\alpha \left\|\mathbf{X}_i^R(n,:)\right\|_2^2} x_n^R(i,t) + j e^{-\alpha \left\|\mathbf{X}_i^I(n,:)\right\|_2^2} x_n^I(i,t) \right), \end{aligned} \tag{7.1.10}$$

where $\kappa = 2\mu\beta\alpha$, $x_n^R(i,t)$ and $x_n^I(i,t)$ are the real and imaginary parts of $x_n(i,t)$. To reduce the computational complexity of the above formulation, especially the sparse penalty term, the first-order Taylor series expansion of exponential functions will be used instead as follows:

$$e^{-\alpha x^2} \approx \begin{cases} 1 - \alpha x^2 & x^2 \le \frac{1}{\alpha}; \\ 0 & \text{otherwise.} \end{cases} \tag{7.1.11}$$

Since the value of exponential function is obviously larger than zero, the approximation of the above formulation is limited to be positive. Then, substituting it into Eq. (7.1.10), a final gradient descent recursion for $\mathbf{X}_i$ can be written as

$$x_n(i+1,t) = x_n(i,t) + \mu r(i) a_m^* \left(\bar{\theta}_n\right) - \kappa \left(g^R(n,i) x_n^R(i,t) + j g^I(n,i) x_n^I(i,t) \right), \tag{7.1.12}$$

where $g^R(n,i) = 1 - \alpha \left\|\mathbf{X}_i^R(n,:)\right\|_2^2$ and $g^I(n,i) = 1 - \alpha \left\|\mathbf{X}_i^I(n,:)\right\|_2^2$, respectively. The proposed algorithm is listed in 6.

7.2 Performance analysis

Here, the real part of data is used to analyze the complexity and steady-state performance for simplicity. The analysis of the imaginary part of data can be done in the similar way. The data listed below are all real parts of the signal model.

Algorithm 6 The $\ell_{2,0}$-LMS algorithm

Input:

The manifold matrix $\mathbf{\Phi}$,

The sampled signal $\mathbf{Y}$,

The parameter μ, κ, α,

The limited error ε, the maximum iteration number C.

Initialization:

1. Set $i = 1$, 2. Set $\mathbf{X}(i) = \mathbf{0}$,

Iteration:

3. Set

$$y_m(t) = \mathbf{Y}(m, t), \tag{7.1.13}$$

$$r(i) = y_m(t) - \mathbf{\Phi}_m \mathbf{x}(i, t), \tag{7.1.14}$$

where $m = \bmod(i, M) + 1$, $t = \lfloor i/M \rfloor + 1$, $\mathbf{\Phi}_m$ is the mth row in matrix $\mathbf{\Phi}$, and $\mathbf{x}(i, t)$ is the tth column in matrix $\mathbf{X}(i)$.

4. Update $\mathbf{X}(i)$ by Eq. 7.1.10.

5. Increase i by one, and judge whether the following termination condition is satisfied:

$$\|\mathbf{X}(i) - \mathbf{X}(i-1)\|_2 < \varepsilon \quad \text{or} \quad i > C, \tag{7.1.15}$$

When Eq. 7.1.15 is not satisfied, go back to 3, otherwise go to output.

Output: The estimated DOA $\mathbf{S}^{\#} = \|\mathbf{X}_i\|_{2,0}$.

Table 7.3: Computational complexities of ℓ_0-LMS and $\ell_{2,0}$-LMS in each period

Methods	Equation	Multiplications	Additions
ℓ_0-LMS	Residual calculation	MNL	MNL
	Gradient descent recursion without sparse penalty	$2MNL$	MNL
	Sparse penalty	$3MNL$	MNL
$\ell_{2,0}$-LMS	Residual calculation	MNL	MNL
	Gradient descent recursion without sparse penalty	$2MNL$	MNL
	Sparse penalty	$MNL(L+3)$	$MNL(L+1)$

Computational complexity: The complexity of each iteration is analyzed in this part. For simplicity, we use a "period" instead of an iteration to analyze the complexity. The "period" is defined as that all the row vectors $\mathbf{\Phi}_m$ in the measurement matrix $\mathbf{\Phi}$ have been adopted once for all the original signal $\mathbf{h}_t$. The complexity of our proposed algorithm and the ℓ_0-LMS algorithm are compared. Note that the ℓ_0-LMS algorithm cannot be applied directly for the MMV form. To compare the complexity, it is used to solve each column in $\mathbf{X}$ as a SMV form. It means that in one "period", all columns in $\mathbf{X}$ will be solved like the $\ell_{2,0}$-LMS algorithm. The complexities are listed in Table. 7.3. The increased complexity is caused by calculating $\|\mathbf{X}_i(n,:)\|_2^2$ for sparse penalty. While it improves the estimation performance as it gives the joint sparsity of snapshots at all times. In general, the adaptive filtering algorithm is used for DOA estimation with a few snapshots, namely, L is very small. Hence, compared to the ℓ_0-LMS algorithm, the increase in complexity of the $\ell_{2,0}$-LMS algorithm is limited.

Steady-state performance analysis: In this part, we will analyze the steady-state mean square derivation between the original signal and the reconstructed signal. And the bound of step-size μ to guarantee convergence will be given.

Suppose that $\mathbf{x}_o$ is the wiener solution, thus the signal received by the mth sensor at time t can be written as [41]

$$y_m(t) = \mathbf{\Phi}\mathbf{x}_o(t) + \bar{e}_m(t), \tag{7.2.1}$$

where $\bar{e}_m(t)$ is the measurement noise at the mth sensor at time t. Assuming $\bar{e}_m(t)$ is Gaussian noise with zero mean, the misalignment can be defined as

$$\mathbf{v}_t(i) = \mathbf{x}(i,t) - \mathbf{x}_o(t). \tag{7.2.2}$$

For simplicity, we define that

$$\mathbf{g}_t(i) = [g(1,i)x_1(i,t), ..., g(N,i)x_N(i,t)]^T. \tag{7.2.3}$$

Subtract $\mathbf{x}_o(t)$ from both sides of Eq. (7.1.12):

$$\mathbf{x}(i+1,t) - \mathbf{x}_o(t) = \mathbf{x}(i,t) + \mu\mathbf{\Phi}_m^T\left[y_m(t) - \mathbf{\Phi}_m\mathbf{x}(i,t)\right] - \kappa\mathbf{g}_t(i) - \mathbf{x}_o(t). \tag{7.2.4}$$

It is equivalent to

$$\mathbf{v}_t(i+1) = \left[\mathbf{I} - \mu\mathbf{\Phi}_m^T\mathbf{\Phi}_m\right]\mathbf{v}_t(i) + \mu\mathbf{\Phi}_m^T\bar{e}_m(t) - \kappa\mathbf{g}_t(i). \tag{7.2.5}$$

The square of the ℓ_2-norm of the misalignment can be written as

$$\|\mathbf{x}(i+1,t)-\mathbf{x}_o(t)\|_2^2=\|\mathbf{v}_t(i+1)\|_2^2=\mathrm{tr}\left[\mathbf{v}_t(i+1)\mathbf{v}_t^T(i+1)\right]. \tag{7.2.6}$$

We have then that

$$\begin{aligned}
\mathbf{v}_t(i+1)\mathbf{v}_t^T(i+1)=&\left[\left[\mathbf{I}-\mu\mathbf{\Phi}_m^T\mathbf{\Phi}_m\right]\mathbf{v}_t(i)+\mu\mathbf{\Phi}_m^T\bar{e}_m(t)-\kappa\mathbf{g}_t(i)\right]\\
&\times\left[\left[\mathbf{I}-\mu\mathbf{\Phi}_m^T\mathbf{\Phi}_m\right]\mathbf{v}_t(i)+\mu\mathbf{\Phi}_m^T\bar{e}_m(t)-\kappa\mathbf{g}_t(i)\right]^T\\
=&\left[\mathbf{I}-\mu\mathbf{\Phi}_m^T\mathbf{\Phi}_m\right]\mathbf{v}_t(i)\mathbf{v}_t^T(i)\left[\mathbf{I}-\mu\mathbf{\Phi}_m^T\mathbf{\Phi}_m\right]^T\\
&+\mu\bar{e}_m(t)\left[\mathbf{I}-\mu\mathbf{\Phi}_m^T\mathbf{\Phi}_m\right]\mathbf{v}_t(i)\mathbf{\Phi}_m\\
&-\kappa\left[\mathbf{I}-\mu\mathbf{\Phi}_m^T\mathbf{\Phi}_m\right]\mathbf{v}_t(i)\mathbf{g}_t^T(i)\\
&+\mu\bar{e}_m(t)\mathbf{\Phi}_m^T\mathbf{v}_t^T(i)\left[\mathbf{I}-\mu\mathbf{\Phi}_m^T\mathbf{\Phi}_m\right]^T\\
&+\mu^2\bar{e}_m^2(t)\mathbf{\Phi}_m^T\mathbf{\Phi}_m\\
&-\mu\kappa\bar{e}_m(t)\mathbf{\Phi}_m^T\mathbf{g}_t^T(i)\\
&-\kappa\mathbf{g}_t(i)\mathbf{v}_t^T(i)\left[\mathbf{I}-\mu\mathbf{\Phi}_m^T\mathbf{\Phi}_m\right]^T\\
&-\mu\kappa\bar{e}_m(t)\mathbf{g}_t(i)\mathbf{\Phi}_m\\
&+\kappa^2\mathbf{g}_t(i)\mathbf{g}_t^T(i).
\end{aligned} \tag{7.2.7}$$

Let

$$\mathbf{T}(n)=E\left\{\mathbf{v}_t(i)\mathbf{v}_t^T(i)\right\} \tag{7.2.8}$$

denote a second moment matrix of the misalignment vector. Define the input correlation matrix and the minimum mean-square estimation error as

$$\mathbf{R}=E\left\{\mathbf{\Phi}_m^T\mathbf{\Phi}_m\right\}, \tag{7.2.9}$$

and

$$P_0=E\left\{\bar{e}_m^2(t)\right\}, \tag{7.2.10}$$

respectively. Since the manifold matrix $\mathbf{\Phi}$ satisfies Independence Assumption (see [88, 99]), taking expectations on both sides of Eq. (7.2.7) yields

$$\begin{aligned}
\mathbf{T}(i+1)\approx&\mathbf{T}(i)-\mu\left[\mathbf{T}(i)\mathbf{R}+\mathbf{R}\mathbf{T}(i)\right]+2\mu^2\mathbf{R}\mathbf{T}(i)\mathbf{R}\\
&+\mu^2\mathbf{R}\mathrm{tr}\left(\mathbf{R}\mathbf{T}(i)\right)+2\kappa\left(\mu\mathbf{R}-\mathbf{I}\right)E\left\{\mathbf{v}_t(i)\mathbf{g}_t^T(i)\right\}\\
&+\kappa^2E\left\{\mathbf{g}_t(i)\mathbf{g}_t^T(i)\right\}+\mu^2P_0\mathbf{R}.
\end{aligned} \tag{7.2.11}$$

According to [88], there is

$$\mathbf{R} \approx M\mathbf{I}, \tag{7.2.12}$$

where M is the number of the sensors. Thus, Eq. (7.2.11) can be written as

$$\begin{aligned}\mathbf{T}(i+1) \approx & \left[1 - 2M\mu + 2M^2\mu^2\right] \mathbf{T}(i) \\ & + \mu M^2 \mathrm{tr}\left(\mathbf{T}(i)\right) \mathbf{I} + 2\kappa\left(\mu M - 1\right) E\left\{\mathbf{v}_t(i)\mathbf{g}_t^T(i)\right\} \\ & + \kappa^2 E\left\{\mathbf{g}_t(i)\mathbf{g}_t^T(i)\right\} + \mu^2 M P_0 \mathbf{I}.\end{aligned} \tag{7.2.13}$$

Define

$$C(i) = \mathrm{tr}\left(\mathbf{T}(i)\right). \tag{7.2.14}$$

Taking the trace on both sides of Eq. (7.2.13), we have that

$$\begin{aligned}C(i+1) = & \left[1 - 2M\mu + \left(N+2\right) M^2\mu^2\right] C(i) \\ & + 2\kappa\left(\mu M - 1\right) V(i) + \kappa^2 G(i) + \mu^2 M N P_0,\end{aligned} \tag{7.2.15}$$

where

$$\begin{aligned}V(i) &= E\left\{\mathbf{v}_t^T(i)\mathbf{g}_t(i)\right\}, \\ G(i) &= E\left\{\mathbf{g}_t^T(i)\mathbf{g}_t(i)\right\}.\end{aligned} \tag{7.2.16}$$

Both $V(i)$ and $G(i)$ are bounded as follows:

$$\begin{aligned}|V(i)| &= \left|E\left\{\left(\mathbf{x}(i,t) - \mathbf{x}(t)\right)\mathbf{g}_t(i)\right\}\right| \\ &\leq E\left\{\left|\left(\mathbf{x}(i,t) - \mathbf{x}_o(t)\right)\mathbf{g}_t(i)\right|\right\} \\ &\leq \sum_{n=1}^{N} E\left\{\left|\left(x_n(i,t) - x_{on}(t)\right) g(n,i) x_n(i,t)\right|\right\} \\ &= \sum_{\|\mathbf{X}_i(n,:)\|_2^2 \leq \frac{1}{\alpha}} E\left|\left(x_n(i,t) - x_{on}(t)\right) g(n,i) x_n(i,t)\right| \\ &= \sum_{\|\mathbf{X}_i(n,:)\|_2^2 \leq \frac{1}{\alpha}} E\left\{\left|\left(x_n(i,t) - x_{on}(t)\right) x_n(i,t)\right| \left|g(n,i)\right|\right\} \\ &\leq \sum_{\|\mathbf{X}_i(n,:)\|_2^2 \leq \frac{1}{\alpha}} E\left\{\left|\left(x_n(i,t) - x_{on}(t)\right) x_n(i,t)\right|\right\} \\ &\leq \sum_{\|\mathbf{X}_i(n,:)\|_2^2 \leq \frac{1}{\alpha}} E\left|x_n^2(i,t)\right| + E\left|x_n(i,t) x_{on}(t)\right| \\ &\leq N + \left\|\mathbf{x}_o(t)\right\|_1,\end{aligned} \tag{7.2.17}$$

and

$$\begin{aligned} |G(i)| &= \left| E\left\{ \mathbf{g}_t^T(i)\mathbf{g}_t(i) \right\} \right| \\ &\leq E\left| \mathbf{g}_t^T(i)\mathbf{g}_t(i) \right| \\ &\leq \sum_{n=1}^{N} E\left| g^2(n,i)x_n^2(i,t) \right| \\ &\leq N. \end{aligned} \tag{7.2.18}$$

Therefore, in order to guarantee convergence of Eq. (7.2.15), the following condition should be satisfied:

$$\left| 1 - 2M\mu + (N+2)\, M^2\mu^2 \right| < 1. \tag{7.2.19}$$

Thus the step-size should be chosen by

$$0 < \mu < \frac{2}{M\,(N+2)}. \tag{7.2.20}$$

Let $i \to \infty$, the mean square derivation in steady state is

$$\begin{aligned} C\,(\infty) = &\left[1 - 2M\mu + (N+2)\, M^2\mu^2 \right] C(\infty) \\ &+ 2\kappa\,(\mu M - 1)\, V(\infty) + \kappa^2 G(\infty) + \mu^2 MNP_0. \end{aligned} \tag{7.2.21}$$

It is equivalent to

$$C\,(\infty) = B\left[2\kappa\,(\mu M - 1)\, V(\infty) + \kappa^2 G(\infty) + \mu^2 MNP_0 \right], \tag{7.2.22}$$

where

$$B = \frac{1}{2M\mu - (N+2)\, M^2\mu^2}. \tag{7.2.23}$$

By Eqs. (7.2.17) and (7.2.18), we have the upper bound of the derivation as follows

$$E\left\{ \left\| \mathbf{x}_F(t) - \mathbf{x}_o(t) \right\|_2^2 \right\} \leq B\left[2\kappa\,(\mu M - 1)\,(N + \left\| \mathbf{x}_o(t) \right\|_1) + \kappa^2 N + \mu^2 MNP_0 \right], \tag{7.2.24}$$

where $\mathbf{x}_F(t)$ denotes the reconstruction signal by the $\ell_{2,0}$-LMS algorithm.

7.3 Simulation results

We first consider a typical massive MIMO uniform linear array (ULA) with $M = 100$ sensors. The total number of snapshots is $L = 2$. There are $N = 91$ potential DOAs and these are distributed uniformly over the range $[0°, 180°]$. The frequency of signals f_o is 900MHz, and the sampling frequency f_s is 450MHz. The distance between the sensors is equal to half the wavelength $\lambda = c/f_o$, where c is the velocity of light. The $\ell_{2,0}$-LMS algorithm is compared

with the JLZA-DOA [10], M-FCOUSS [102], MSMPDOA [103] ℓ_0-LMS [41] algorithms and the traditional DOA methods (Capon's method [12] and MUSIC [11]), the reduced-rank DOA estimation [104] and the NLMS algorithm [105]. Note that the JLZA-DOA, M-FCOUSS and MSMPDOA algorithms can be compared directly as they are proposed for DOA estimation with small snapshots. The ℓ_0-LMS algorithm [41] cannot be applied directly for DOA estimation with small snapshots, because it is proposed for the SMV problem. In order to compare the estimation performance, we use the ℓ_0-LMS algorithm to solve each column in $\mathbf{X}$ as a SMV problem. Thus it can also be used to solve the matrix $\mathbf{X}$ as other algorithms.

In the first experiment, there are $K = 2$ uncorrelated source signals originating from $[40°, 80°]$. We set SNR as 10 dB. The parameter for each algorithm are set as follows:

- JLZA-DOA: $\rho = 0.3$, $\eta = 0.5$, $\gamma = 0.5$, $\beta = 0.5$;
- M-FCOUSS: $p = 0.5$;
- MSMPDOA: $K = N/4$;
- ℓ_0-LMS: $\alpha = 5$, $\mu = 4 \times 10^{-3}$, $\kappa = 1 \times 10^{-4}$
- $\ell_{2,0}$-LMS: $\alpha = 5$, $\mu = 4 \times 10^{-3}$, $\kappa = 1 \times 10^{-4}$
- MUSIC: the number of source signals $K_{MUSIC} = 2$
- Reduced-rank DOA estimation: the number of source signals $K_{RD} = 2$
- NLMS: $\mu = 0.1$

The results are shown in Fig. 7.2. It can be found that the MSMPDOA, ℓ_0-LMS, $\ell_{2,0}$-LMS, NLMS algorithms and the MUSIC can reconstruct the source signals at $[40°, 80°]$, which are the true DOAs. However, due to the noise, the MSMPDOA algorithm will bring some false signals at the other potential DOAs. In order to show the good performance of our algorithm under limit of the number of sensors in more detail , we set as $M = 10$ and the result is shown in Fig. 7.3. It can be found that except the MUSIC, Capon's method, reduced-rank DOA estimation and NLMS algorithm, all CS-based algorithms can reconstruct the source signals at $[40°, 80°]$. However, the JLZA-DOA, M-FOCUSS and MSMPDOA algorithms will bring some false signals at the other potential DOAs.

By taking the ℓ_2-norm of the source signals at each potential DOAs as,

$$\text{Power}\left(\bar{\theta}_n\right) = \log\left[\sum_{t=1}^{L} x_n^2(t)\right], \tag{7.3.1}$$

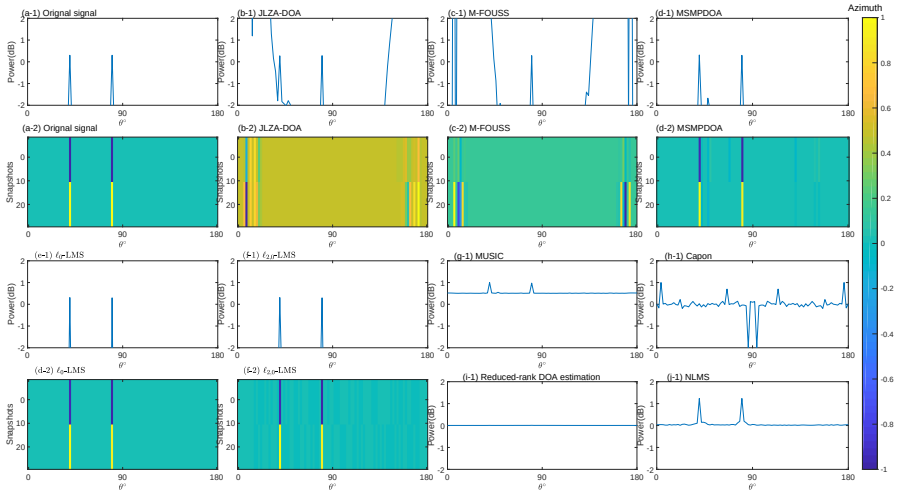


Figure 7.2: ($M = 100$) (1) DOA estimation results and (2) Multiple snapshot reconstruction with the ULA for original signal (a), JLAZ-DOA (b), M-FCOUSS (c), MSMPDOA (d), ℓ_0-LMS (e), $\ell_{2,0}$-LMS (f), MUSIC (g), Capon's method (h), Reduced-rank DOA estimation (i) and NLMS (j).

the DOA estimation results are achieved. It can be found that some false source signals exist in the results of the JLZA-DOA, M-FCOUSS and MSMPDOA algorithms due to the false signals.

In the second experiment, the reconstruction accuracy against sensor number (M) is studied. In order to analyze the limit of the number of sensors, M is set as $[10, 20, 40, 60, 80, 100]$. SNR is set as 10 dB. The average RMSEs are obtained after 100 Monte Carlo trials. The results are shown in Fig. 7.4. It can be found that when $M \leq 60$, the RMSE of all algorithms decreases as M increases. When $M > 60$, the RMSEs of the JLZA-DOA and M-FOCUSS algorithms increases rapidly, while the RMSEs of the MSMPDOA, ℓ_0-LMS and $\ell_{2,0}$-LMS algorithms decrease. Hence, our proposed algorithm has the lowest RMSE than other algorithms for massive MIMO systems.

In the third experiment, the reconstruction accuracy against noise is studied. The sensor number is set as $M = 24$. The total number of snapshots is $L = 5$. SNR is set as $[0, 5, 10, 15, 20]$ dB. We use root mean square error (RMSE) to evaluate the reconstruction accuracy which is defined as

$$\text{RMSE} = \frac{\sqrt{\sum_{i=1}^{L}\sum_{j=1}^{N}\left(\hat{x}_{i,j} - x_{i,j}^{\text{true}}\right)^2}}{L}, \tag{7.3.2}$$

where $\hat{x}_{i,j}$ denotes the element of the jth column and the ith row in the estimated $\mathbf{X}$, and $\hat{x}_{i,j}^{\text{true}}$ denotes the element of the jth column and the ith row in the true $\mathbf{X}$. The average RMSE obtained after 100 Monte Carlo trials. The results are shown in Fig. 7.5. It can be found that the RMSE decreases as SNR increases. When SNR is small (SNR<20 dB), the adaptive filtering

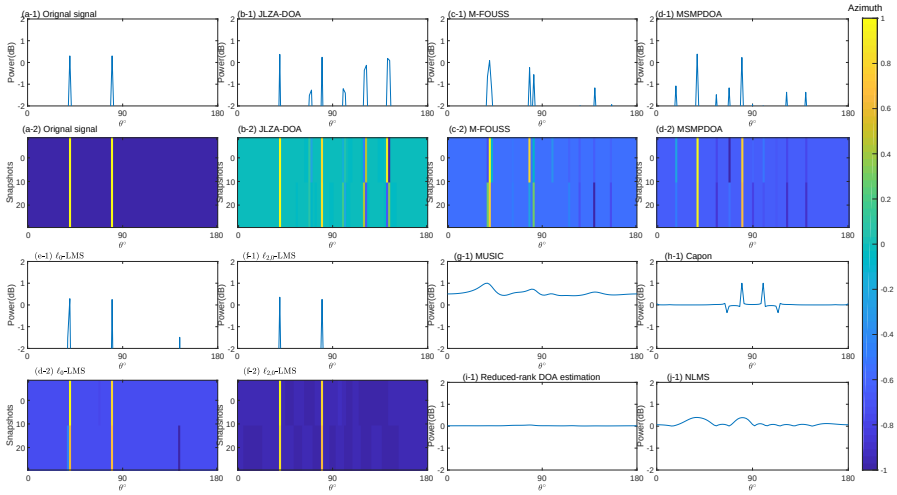


Figure 7.3: ($M = 10$) (1) DOA estimation results and (2) Multiple snapshot reconstruction with the ULA for original signal (a), JLAZ-DOA (b), M-FCOUSS (c), MSMPDOA (d), ℓ_0-LMS (e), $\ell_{2,0}$-LMS (f), MUSIC (g), Capon's method (h), Reduced-rank DOA estimation (i) and NLMS (j).

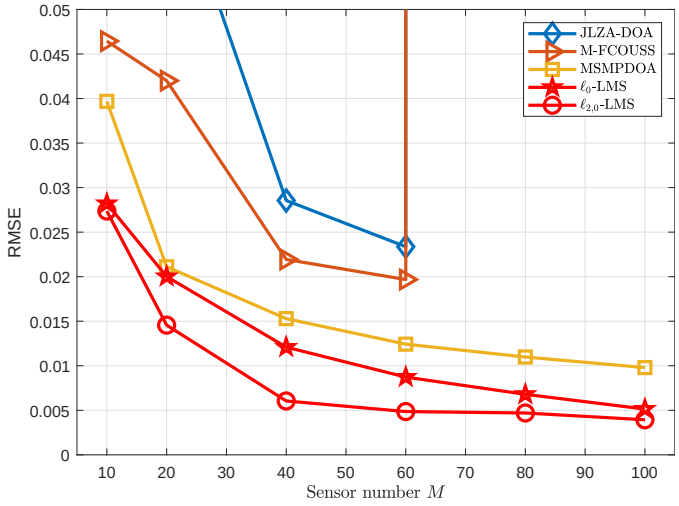


Figure 7.4: RMSE against sensor number.

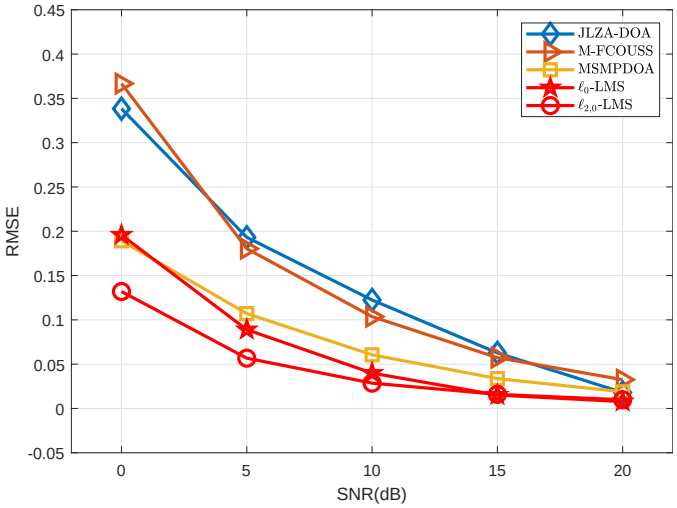


Figure 7.5: RMSE against SNR.

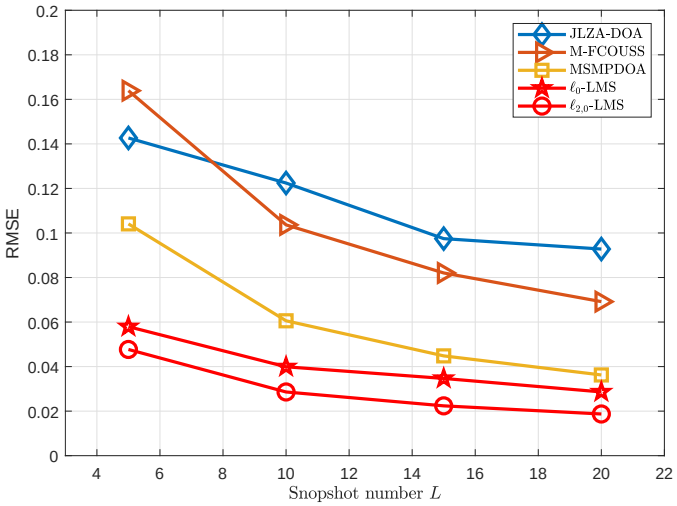


Figure 7.6: RMSE against snapshots.

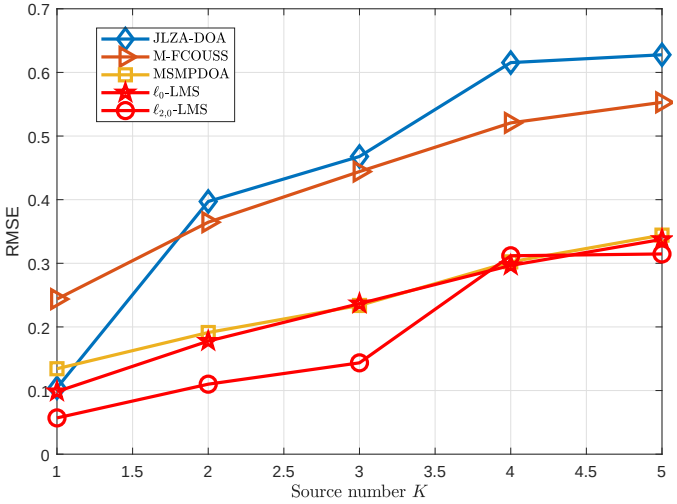


Figure 7.7: RMSE against source number.

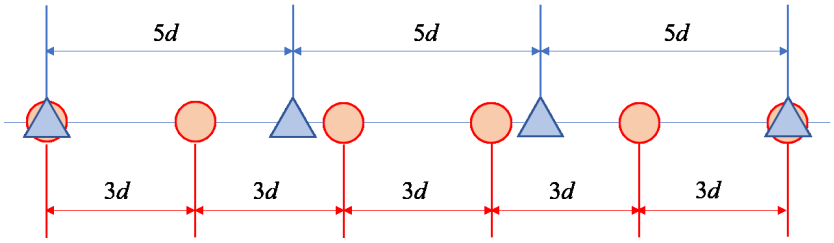


Figure 7.8: The structure of the coprime array.

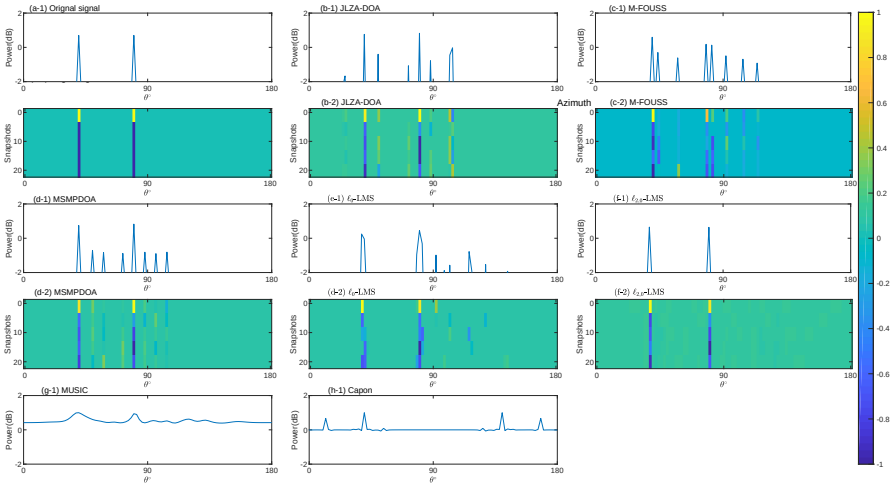


Figure 7.9: (1) DOA estimation results and (2) Multiple snapshot reconstruction with the coprime array for original signal (a), JLAZ-DOA (b), M-FCOUSS (c), MSMPDOA (d), ℓ_0-LMS (e), $\ell_{2,0}$-LMS (f), MUSIC (g) and Capon's method (h).

algorithms and the MSMPDOA algorithm can achieve the lower RMSE than the JLZA-DOA and M-FOCUSS algorithms. Our proposed algorithm ($\ell_{2,0}$-LMS) has the lowest RMSE when SNR is smaller than 15 dB.

In the forth experiment, the reconstruction accuracy against snapshots (L) is studied. SNR is set as 10 dB and the number of snapshots is set as $[5, 10, 15, 20]$. The other parameters are set as the same as the second experiment. The average RMSE obtained after 100 Monte Carlo trials. The results are shown in Fig. 7.6. It can be found that the RMSEs of all algorithms decrease as L increases. Our proposed algorithm ($\ell_{2,0}$-LMS) has the lowest RMSE.

In the fifth experiment, the reconstruction accuracy against source number (K) is studied. SNR is set as 10 dB and the number of sources is set as $[1, 2, 3, 4, 5]$. The other parameters are set as the same as the second experiment. The average RMSE obtained after 100 Monte Carlo trials. The results are shown in Fig. 7.7. It can be found that the RMSEs of all algorithms increase as K increases. Our proposed algorithm ($\ell_{2,0}$-LMS) has the lowest RMSE in most cases.

In the sixth experiment, the case with a coprime array with $M = 8$ sensors is considered [106, 107]. The structure of the coprime array is shown in Fig. 7.8. It is constructed by two subarrays. The distance between the two adjacent detectors in one subarray is $3d$, and the distance between the two adjacent detectors in anther subarray is $5d$. The first sensors in both subarrays are co-located at the zeroth position, and the last sensors in both subarrays are co-located at the terminal position. The total number of snapshots is $L = 5$. SNR is set as 0 dB. The other parameters are set as the same as the first experiment. The results are shown in Fig. 7.9. It can be found that except the MUSIC and Capon's method, all these algorithms can reconstruct the source signals at $[40°, 80°]$, which are the true DOAs. However, some false signals at the other potential DOAs are estimated in the JLZA-DOA, M-FCOUSS, MSMPDOA and ℓ_0-LMS algorithms because of the noise. Only the proposed $\ell_{2,0}$-LMS algorithm can achieve the real DOAs.

Chapter 8

Conclusions and future works

This chapter summarizes the main results of this thesis and provides some possible future works.

8.1 Conclusions

This thesis has systematically studied the CS-based high resolution radar signal processing algorithms. In order to reduce the complexity and improve the accuracy of the existing CS-based radar signal processing algorithms, the low-complexity CS-based algorithms have been proposed. The main contributions are drawn as follows.

- **2D CS-based algorithms for delay Doppler joint estimation**

  We have proposed a 2D data model for the pulse Doppler radar system with the RD method. In this method, the data is under-sampled by a low rate ADC. Then the 2D CS-based algorithms (i.e., 2D-ZAP, 2D-IHT, 2D-ISTA, and 2D-FISTA) have been proposed for detecting the sparse targets from the under-sampled data. Since the 2D-CS algorithms solve the 2D data model without vectorizing, the memory requirement and complexity are significantly reduced. Numerical simulations have been provided to validate the performances of our proposed algorithms.

- **Robust 2D CS-based algorithms for delay Doppler joint estimation**

  In order to reduce interference from non-Gaussian impulse noise, the robust 2D CS-based algorithms which are improved from 2D CS-based algorithms by robust cost function, e.g., ℓ_1-norm, ℓ_p-norm and LL_2-norm. The simulation results show that the proposed robust 2D-CS based algorithms have higher accuracy than the 2D-CS based algorithms in non-Gaussian impulse noise environment.

- **Dimension-reduced DOA estimation based on $\ell_{2,1}$-norm penalty**

The dimension-reduced DOA estimation method has been proposed for the case of small snapshots. The proposed algorithm can decrease the calculation time for the grid search. In addition, we extend our proposed method from 1D-DOA estimation to 2D-DOA estimation. The simulation results show that our proposed method has a higher frequency of direction and requires less calculation time than the existing DOA estimation algorithms.

- **Adaptive filtering algorithm for DOA estimation with small snapshots**

  A novel adaptive filtering algorithm has been proposed for DOA estimation with small snapshots. In order to utilize the joint sparsity of MMV, a mixed norm penalty is used in the proposed $\ell_{2,0}$-LMS algorithm. The proposed algorithm inherits the low complexity of the LMS algorithm, and improves the robustness against noise. The simulation results show that, compared with the existing DOA estimation algorithms, our proposed algorithm has a greater accuracy in low SNR environment.

8.2 Future works

Some directions for future research are as follows.

- In this thesis, we have shown the delay Doppler joint estimation and DOA estimation based on CS theory, respectively. In the next step, the problem of how to estimate the delay, the Doppler shift and the angle at the same time will be considered. Furthermore, the relationship between the delay, the Doppler shift, and the angle will be researched and used to reduce the algorithm complexity.

- This thesis only focuses on the ideal case that the target is a point. In practice, with the resolution of radar improving, the high-resolution radar imaging becomes possible. To further study the compressive sensing, the CS-based radar imaging problem would be very interesting.

- In practice, such as self-driving, jamming and clutter will degrade the performance of the CS-based algorithms. How to eliminate the interference of jamming and clutter in CS framework would also be very interesting.

Publication list

Journal Papers:

1. B. Liu, G. Gui, S. Matsushita, L. Xu, "Dimension-reduced direction-of-arrival estimation based on L2,1-norm penalty," *IEEE Access*, vol.6, pp.44433-44444, 2018.

2. B. Liu, Y. Zhao, X. Zhu, S. Matsushita, L. Xu, "Sparse detection algorithms based on two-dimensional compressive sensing for sub-Nyquist pulse Doppler radar systems," *IEEE Access*, vol.7, pp.18649-18661, 2019.

3. B. Liu, G. Gui, S. Matsushita, L. Xu, "Adaptive filtering algorithm for direction-of-arrival (DOA) estimation with small snapshots," *Digital Signal Processing*, vol. 94, pp. 84-95, 2019.

4. B. Liu, G. Gui, S. Matsushita, L. Xu, "Compressive sensing-based adaptive sparse multi-path channel estimation," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol.21, Issue 1, pp.153-158, 2017.

5. S. Wang, B. Liu, L. Xu, T. Tamura, N. Kyouno, X. Liu, J. Chen, "Application of compressed sensing for selecting relevant variables for a model to predict the quality of Japanese fermented soy sauce," *Innovative Food Science & Emerging Technologies*, vol. 59, p. 102241, 2020.

6. G. Gui, W. Peng, L. Xu, B. Liu, F. Adachi, "Variable-step-size based sparse adaptive filtering algorithm for channel estimation in broadband wireless communication systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no.1, p.195, 2014.

International Conference Papers:

1. B. Liu, G. Gui, S. Matsushita, L. Xu, "Anti-jamming DOA estimation based on compressive sensing via blocking matrix," *IEEE International Conference on Digital Signal Processing (DSP)*, Shanghai, China, Nov. 2018.

2. B. Liu, G. Gui, S. Matsushita, L. Xu, "Compressive sensing based direction-of-arrival estimation in MIMO Radars in presence of strong jamming via blocking matrix," *IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Sheffield, UK, Jul. 2018, pp.292-296.

3. B. Liu, G. Gui, S. Matsushita, L. Xu, "Two-dimensional compressive sensing based DOA estimation for single snapshot," *International Workshop on Multidimensional (nD) systems (nDS)*, Zielona Gora, Poland, Sep. 2017.

4. B. Liu, G. Gui, S. Matsushita, L. Xu, "Sparse target detection of pulse Doppler Radar based on two dimensional iterative hard thresholding algorithm," *Chinese Control and Decision Conference (CCDC)*, Chongqing, China, May 2017, pp.4856-4860.

5. B. Liu, G. Gui, Y. Zhu, L. Xu, "Frequency-domain adaptive sparse signal reconstruction at sub-Nyquist rate," *IEEE/CIC International Conference on Communications in China (ICCC)*, Chengdu, China, Jul. 2016.

6. G. Gui, B. Liu, L. Xu, W. Ma, "Fast NLMF-type algorithms for adaptive sparse system identifications," Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Hong Kong, China, Dec. 2015, pp.958-962.

7. B. Liu, G. Gui, L. Xu, N. Shimoi, "Iterative-promoting variable step-size least mean square algorithm for accelerating adaptive sparse channel estimation," *Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, Hangzhou, China, Jul. 2015.

8. B. Liu, G. Gui, L. Xu, "Iterative-promoting variable step-size least mean square algorithm for adaptive sparse channel estimation," *Asia-Pacific Conference on Communications (APCC)*, Kyoto, Japan, Oct. 2015.

References

[1] M. A. Richards, J. Scheer, W. A. Holm, and W. L. Melvin, *Principles of modern radar*. Citeseer, 2010.

[2] C. Warren, A. Giannopoulos, and I. Giannakis, "gprMax: Open source software to simulate electromagnetic wave propagation for Ground Penetrating Radar," *Computer Physics Communications*, vol. 209, pp. 163–170, 2016.

[3] W. J. Boord and J. B. Hoffman, *Air and missile defense systems engineering*. CRC Press, 2016.

[4] M. Aki, T. Rojanaarpa, K. Nakano, Y. Suda, N. Takasuka, T. Isogai, and T. Kawai, "Road surface recognition using laser radar for automatic platooning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2800–2810, 2016.

[5] M. A. Hadi, S. Alshebeili, K. Jamil, and F. E. A. El-Samie, "Compressive sensing applied to radar systems: an overview," *Signal, Image and Video Processing*, vol. 9, no. 1, pp. 25–39, 2015.

[6] J. H. G. Ender, "On compressive sensing applied to radar," *Signal Processing*, vol. 90, no. 5, pp. 1402–1414, 2010.

[7] W. Hirt, "Ultra-wideband radio technology: overview and future research," *Computer Communications*, vol. 26, no. 1, pp. 46–52, 2003.

[8] C. Fowler, J. Entzminger, J. A. M. E. S. Corum *et al.*, "Assessment of ultra-wideband (UWB) technology," *IEEE Aerospace and Electronic Systems Magazine*, vol. 5, no. 11, pp. 45–49, 1990.

[9] J. Li and P. Stoica, "MIMO radar with colocated antennas," *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 106–114, 2007.

[10] M. M. Hyder and K. Mahata, "Direction-of-arrival estimation using a mixed $\ell_{2,0}$ norm approximation," *IEEE Transactions on Signal processing*, vol. 58, no. 9, pp. 4646–4655, 2010.

[11] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[12] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[13] P. Stoica, H. Li, and J. Li, "A new derivation of the APES filter," *IEEE Signal Processing Letters*, vol. 6, no. 8, pp. 205–206, 1999.

[14] B. M. Radich and K. M. Buckley, "Single-snapshot DOA estimation and source number detection," *Ieee Signal Processing Letters*, vol. 4, no. 4, pp. 109–111, 1997.

[15] P. Häcker and B. Yang, "Single snapshot DOA estimation," *Advances in Radio Science*, vol. 8, pp. 251–256, 2010.

[16] S. Fortunati, R. Grasso, F. Gini, M. S. Greco, and K. LePage, "Single-snapshot DOA estimation by using compressed sensing," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, p. 120, 2014.

[17] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[18] E. J. Candes, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9-10, pp. 589–592, 2008.

[19] Y. Tsaig and D. L. Donoho, "Extensions of compressed sensing," *Signal processing*, vol. 86, no. 3, pp. 549–571, 2006.

[20] E. Candes and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, no. 3, p. 969, 2007.

[21] J. Romberg, "Compressive sensing by random convolution," *SIAM Journal on Imaging Sciences*, vol. 2, no. 4, pp. 1098–1128, 2009.

[22] R. G. Baraniuk, M. Wakin, M. F. Duarte, J. A. Tropp, and D. Baron, "Random filters for compressive sampling and reconstruction," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, May 2006.

[23] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk, "Beyond Nyquist: Efficient sampling of sparse bandlimited signals," *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 520–544, 2010.

[24] Z. Pang, M. Yuan, and M. B. Wakin, "A random demodulation architecture for subsampling acoustic emission signals in structural health monitoring," *Journal of Sound and Vibration*, vol. 431, pp. 390–404, 2018.

[25] S. Kirolos, J. Laska, M. Wakin, M. Duarte, D. Baron, T. Ragheb, Y. Massoud, and R. Baraniuk, "Analog-to-information conversion via random demodulation," in *IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*, Richardson, TX, USA, Oct. 2006, pp. 71–74.

[26] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[27] M. Mishali and Y. C. Eldar, "From theory to practice: Sub-Nyquist sampling of sparse wideband analog signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 375–391, 2010.

[28] M. Mishali, Y. C. Eldar, O. Dounaevsky, and E. Shoshan, "Xampling: Analog to digital at sub-Nyquist rates," *IET Circuits, Devices & Systems*, vol. 5, no. 1, pp. 8–20, 2011.

[29] M. Mishali and Y. C. Eldar, "Sub-Nyquist sampling," *IEEE Signal Processing Magazine*, vol. 28, no. 6, pp. 98–124, 2011.

[30] M. A. Lexa, M. E. Davies, and J. S. Thompson, "Reconciling compressive sampling systems for spectrally sparse continuous-time signals," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 155–171, 2011.

[31] D. King, R. Packard, and R. Thomas, "Unequally-spaced, broad-band antenna arrays," *IRE Transactions on Antennas and Propagation*, vol. 8, no. 4, pp. 380–384, 1960.

[32] M. Skolnik, "Resolution of angular ambiguities in radar array antennas with widely-spaced elements and grating lobes," *IRE Transactions on Antennas and Propagation*, vol. 10, no. 3, pp. 351–352, 1962.

[33] P. E. Pace, R. E. Leino, and D. Styer, "Use of the symmetrical number system in resolving single-frequency undersampling aliases," *IEEE Transactions on Signal Processing*, vol. 45, no. 5, pp. 1153–1160, 1997.

[34] A. C. Gilbert, M. J. Strauss, and J. A. Tropp, "A tutorial on fast Fourier sampling," *IEEE Signal processing magazine*, vol. 25, no. 2, pp. 57–66, 2008.

[35] D. E. Bellasi, L. Bettini, C. Benkeser, T. Burger, Q. Huang, and C. Studer, "VLSI design of a monolithic compressive-sensing wideband analog-to-information converter," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 4, pp. 552–565, 2013.

[36] H. Chi, Y. Mei, Y. Chen, D. Wang, S. Zheng, X. Jin, and X. Zhang, "Microwave spectral analysis based on photonic compressive sampling with random demodulation," *Optics Letters*, vol. 37, no. 22, pp. 4636–4638, 2012.

[37] T. Ragheb, J. N. Laska, H. Nejati, S. Kirolos, R. G. Baraniuk, and Y. Massoud, "A prototype hardware for random demodulation based compressive analog-to-digital conversion," in *51st Midwest Symposium on Circuits and Systems*, Knoxville, TN, USA, Aug. 2008.

[38] W. Xu, Y. Cui, Y. Wang, S. Wang, and J. Lin, "A hardware implementation of random demodulation analog-to-information converter," *IEICE Electronics Express*, vol. 13, no. 16, pp. 20 160 465–20 160 465, 2016.

[39] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[40] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[41] J. Jin, Y. Gu, and S. Mei, "A stochastic gradient approach on compressive sensing signal reconstruction based on adaptive filtering framework," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 409–420, 2010.

[42] Y. Li, Y. Lin, X. Cheng, Z. Xiao, F. Shu, and G. Gui, "Nonconvex penalized regularization for robust sparse recovery in the presence of SαS noise," *IEEE Access*, vol. 6, pp. 25 474–25 485, 2018.

[43] Y. Li, X. Cheng, and G. Gui, "Co-robust-ADMM-net: Joint ADMM framework and DNN for robust sparse composite regularization," *IEEE Access*, vol. 6, pp. 47 943–47 952, 2018.

[44] L. Xiao, X.-G. Xia, and W. Wang, "Multi-stage robust Chinese remainder theorem," *IEEE Transactions on Signal Processing*, vol. 62, no. 18, pp. 4772–4785, 2014.

[45] L. Xiao and X.-G. Xia, "Frequency determination from truly sub-Nyquist samplers based on robust Chinese remainder theorem," *Signal Processing*, vol. 150, pp. 248–258, 2018.

[46] J. Xu, Z.-Z. Huang, Z.-R. Wang, L. Xiao, X.-G. Xia, and T. Long, "Radial velocity retrieval for multichannel SAR moving targets with time–space Doppler deambiguity," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 1, pp. 35–48, 2017.

[47] M. A. Herman and T. Strohmer, "High-resolution radar via compressed sensing," *IEEE transactions on signal processing*, vol. 57, no. 6, pp. 2275–2284, 2009.

[48] G. Shi, J. Lin, X. Chen, F. Qi, D. Liu, and L. Zhang, "UWB echo signal detection with ultra-low rate sampling based on compressed sensing," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 4, pp. 379–383, 2008.

[49] G. E. Smith, T. Diethe, Z. Hussain, J. Shawe-Taylor, and D. R. Hardoon, "Compressed sampling for pulse Doppler radar," in *2010 IEEE Radar Conference*, Washington, DC, USA, May 2010, pp. 887–892.

[50] R. A. Sevimli, M. Tofighi, and A. E. Cetin, "Range-Doppler radar target detection using denoising within the compressive sensing framework," in *22nd European Signal Processing Conference (EUSIPCO)*, Lisbon, Portugal, Nov. 2014, pp. 1950–1954.

[51] Y. Fang, J. Wu, and B. Huang, "2D sparse signal recovery via 2D orthogonal matching pursuit," *Science China Information Sciences*, vol. 55, no. 4, pp. 889–897, 2012.

[52] B. Liu, G. Gui, S. Matsushita, and L. Xu, "Sparse target detection of pulse Doppler radar based on two dimensional iterative hard thresholding algorithm," in *29th Chinese Control And Decision Conference (CCDC)*, Chongqing, China, May 2017, pp. 4856–4860.

[53] D. G. Khairnar, S. N. Merchant, and U. B. Desai, "Radar signal detection in non-Gaussian noise using RBF neural network." *Journal of Computer*, vol. 3, no. 1, pp. 32–39, 2008.

[54] S. M. S. Zobly and Y. M. Kadah, "Multiple measurements vectors compressed sensing for Doppler ultrasound signal reconstruction," in *International Conference on Computing, Electrical and Electronic Engineering (ICCEEE)*, Khartoum, Sudan, Aug. 2013, pp. 319–322.

[55] J. Chen and X. Huo, "Theoretical results on sparse representations of multiple-measurement vectors," *IEEE Transactions on Signal processing*, vol. 54, no. 12, pp. 4634–4643, 2006.

[56] D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3010–3022, 2005.

[57] C. Zheng, G. Li, H. Zhang, and X. Wang, "An approach of DOA estimation using noise subspace weighted ℓ_1 minimization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011, pp. 2856–2859.

[58] X. Xu, X. Wei, and Z. Ye, "DOA estimation based on sparse signal recovery utilizing weighted ℓ_1-norm penalty," *IEEE Signal Processing Letters*, vol. 19, no. 3, pp. 155–158, 2012.

[59] S. Ghofrani, M. G. Amin, and Y. D. Zhang, "High-resolution direction finding of non-stationary signals using matching pursuit," *Signal Processing*, vol. 93, no. 12, pp. 3466–3478, 2013.

[60] G. Su, J. Jin, Y. Gu, and J. Wang, "Performance analysis of ℓ_0 norm constraint least mean square algorithm," *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2223–2235, 2012.

[61] Q. Shen, W. Liu, W. Cui, and S. Wu, "Underdetermined DOA estimation under the compressive sensing framework: A review," *IEEE Access*, vol. 4, pp. 8865–8878, 2016.

[62] J. Yin and T. Chen, "Direction-of-arrival estimation using a sparse representation of array covariance vectors," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4489–4493, 2011.

[63] W. U. Bajwa, K. Gedalyahu, and Y. C. Eldar, "Identification of parametric underspread linear systems and super-resolution radar," *IEEE Transactions on Signal Processing*, vol. 59, no. 6, pp. 2548–2561, 2011.

[64] J. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Transactions on Circuits and Systems*, vol. 25, no. 9, pp. 772–781, 1978.

[65] A. Graham, *Kronecker products and matrix calculus with applications.* Courier Dover Publications, 2018.

[66] H. Ozcelik, M. Herdin, W. Weichselberger, J. Wallace, and E. Bonek, "Deficiencies of 'Kronecker' MIMO radio channel model," *Electronics Letters*, vol. 39, no. 16, pp. 1209–1210, 2003.

[67] K. B. Petersen, M. S. Pedersen *et al.*, *The matrix cookbook*, 2008.

[68] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point Algorithms for Inverse Problems in Science and Engineering*. New York,: Springer, 2011, vol. 49, pp. 185–212.

[69] S. Dahlke, M. Fornasier, and T. Raasch, "Multilevel preconditioning and adaptive sparse solution of inverse problems," *Mathematics of Computation*, vol. 81, no. 277, pp. 419–446, 2012.

[70] E. Van Den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008.

[71] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[72] B. R. Mahafza, *Radar systems analysis and design using MATLAB*. Chapman and Hall/CRC, 2005.

[73] M. Shao and C. L. Nikias, "Signal processing with fractional lower order moments: stable processes and their applications," *Proceedings of the IEEE*, vol. 81, no. 7, pp. 986–1010, 1993.

[74] D. Middleton, "Non-Gaussian noise models in signal processing for telecommunications: new methods an results for class A and class B noise models," *IEEE Transactions on Information Theory*, vol. 45, no. 4, pp. 1129–1149, 1999.

[75] G. Gui, L. Xu, W. Ma, and B. Chen, "Robust adaptive sparse channel estimation in the presence of impulsive noises," in *IEEE International Conference on Digital Signal Processing (DSP)*, Singapore, Singapore, Jul. 2015, pp. 628–632.

[76] R. E. Carrillo and K. E. Barner, "Lorentzian iterative hard thresholding: Robust compressed sensing with prior information," *IEEE Transactions on Signal Processing*, vol. 61, no. 19, pp. 4822–4833, 2013.

[77] K. Adhikari, J. R. Buck, and K. E. Wage, "Extending coprime sensor arrays to achieve the peak side lobe height of a full uniform linear array," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, p. 148, 2014.

[78] Y.-M. Chen, J.-H. Lee, and C.-C. Yeh, "Two-dimensional angle-of-arrival estimation for uniform planar arrays with sensor position errors," in *IEE Proceedings F (Radar and Signal Processing)*, vol. 140, no. 1. IET, 1993, pp. 37–42.

[79] J.-A. Tsai, R. M. Buehrer, and B. D. Woerner, "BER performance of a uniform circular array versus a uniform linear array in a mobile radio environment," *IEEE Transactions on Wireless Communications*, vol. 3, no. 3, pp. 695–700, 2004.

[80] N. Hu, Z. Ye, X. Xu, and M. Bao, "DOA estimation for sparse array via sparse signal reconstruction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 760–773, 2013.

[81] C. Ma, T. S. Yeo, H. S. Tan, J. Wang, and B. Chen, "Three-dimensional ISAR imaging using a two-dimensional sparse antenna array," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 3, pp. 378–382, 2008.

[82] G. Prisco and M. D'Urso, "Maximally sparse arrays via sequential convex optimizations," *IEEE Antennas and Wireless Propagation Letters*, vol. 11, pp. 192–195, 2012.

[83] M. Mishali and Y. C. Eldar, "Reduce and boost: Recovering arbitrary sets of jointly sparse vectors," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4692–4702, 2008.

[84] S. Sarvotham, D. Baron, M. Wakin, M. F. Duarte, and R. G. Baraniuk, "Distributed compressed sensing of jointly sparse signals," in *Asilomar conference on signals, systems, and computers*, Pacific Grove, CA, USA, Nov. 2005, pp. 1537–1541.

[85] M. Kowalski, "Sparse regression using mixed norms," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 303–324, 2009.

[86] D. H. Johnson and D. E. Dudgeon, *Array signal processing: concepts and techniques*. PTR Prentice Hall Englewood Cliffs, 1993.

[87] S. Foucart and H. Rauhut, *A mathematical introduction to compressive sensing*. Birkhäuser Basel, 2013, vol. 1, no. 3.

[88] A. Xenaki, P. Gerstoft, and K. Mosegaard, "Compressive beamforming," *The Journal of the Acoustical Society of America*, vol. 136, no. 1, pp. 260–271, 2014.

[89] S. Foucart and H. Rauhut, "A mathematical introduction to compressive sensing," *Bull. Am. Math*, vol. 54, pp. 151–165, 2017.

[90] M. Golbabaee and P. Vandergheynst, "Hyperspectral image compressed sensing via low-rank and joint-sparse matrix recovery," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 2741–2744.

[91] M. Grant, S. Boyd, and Y. Ye, "Cvx: Matlab software for disciplined convex programming," 2008.

[92] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[93] J. H. Cozzens and M. J. Sousa, "Source enumeration in a correlated signal environment," *IEEE Transactions on Signal Processing*, vol. 42, no. 2, pp. 304–317, 1994.

[94] H.-T. Wu, J.-F. Yang, and F.-K. Chen, "Source number estimator using Gerschgorin disks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, Adelaide, SA, Australia, Apr. 1994, pp. 261–264.

[95] K. M. Wong, Q. Wu, and P. Stoica, "Generalized correlation decomposition applied to array processing in unknown noise environments," in *Advances in spectrum analysis and array processing*, vol. III. Prentice-Hall, Inc., 1995, pp. 219–323.

[96] P. L. Feintuch, "An adaptive recursive LMS filter," *Proceedings of the IEEE*, vol. 64, no. 11, pp. 1622–1624, 1976.

[97] G. Gui, W. Peng, and F. Adachi, "Adaptive system identification using robust LMS/F algorithm," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 2956–2963, 2014.

[98] Y. Gu, J. Jin, and S. Mei, "$\ell 0$ norm constraint LMS algorithm for sparse system identification," *IEEE Signal Processing Letters*, vol. 16, no. 9, pp. 774–777, 2009.

[99] S. S. Haykin, *Adaptive filter theory*. Pearson Education India, 2008.

[100] M. M. Hyder and K. Mahata, "A robust algorithm for joint-sparse recovery," *IEEE Signal Processing Letters*, vol. 16, no. 12, pp. 1091–1094, 2009.

[101] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping, "Use of the zero-norm with linear models and kernel methods," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1439–1461, 2003.

[102] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2477–2488, 2005.

[103] S. F. Cotter, "Multiple snapshot matching pursuit for direction of arrival (DOA) estimation," in *Signal Processing Conference*, Poznan, Poland, May 2007, pp. 247–251.

[104] L. Qiu, Y. Cai, R. C. de Lamare, and M. Zhao, "Reduced-rank doa estimation algorithms based on alternating low-rank decomposition," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 565–569, 2016.

[105] G. Bakhshi and K. Shahtalebi, "Role of the NLMS algorithm in direction of arrival estimation for antenna arrays," *IEEE Communications Letters*, vol. 22, no. 4, pp. 760–763, 2017.

[106] C. Zhou, Y. Gu, S. He, and Z. Shi, "A robust and efficient algorithm for coprime array adaptive beamforming," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1099–1112, 2017.

[107] C. Zhou, Y. Gu, X. Fan, Z. Shi, G. Mao, and Y. D. Zhang, "Direction-of-arrival estimation for coprime array via virtual array interpolation," *IEEE Transactions on Signal Processing*, vol. 66, no. 22, pp. 5956–5971, 2018.