ROBUST AND EFFICIENT ALGORITHMS FOR MILLIMETER-WAVE RADAR LOCALIZATION AND IMAGING

ROBUST AND EFFICIENT ALGORITHMS FOR MILLIMETER-WAVE RADAR LOCALIZATION AND IMAGING

BY

WEI ZHAO, M.A.Sc.

a Thesis

submitted to the Computing & Software

and the School of Graduate Studies

of McMaster University

in partial fulfilment of the requirements

for the degree of

Doctor of Philosophy

Doctor of Philosophy (2024)	McMaster University
(Computing & Software)	Hamilton, Ontario, Canada

TITLE:	Robust and Efficient Algorithms for Millimeter-Wave Radar Localization and Imaging
AUTHOR:	Wei Zhao M.A.Sc. (Electrical & Computer Engineering), McMaster University, Hamilton, Canada
SUPERVISOR:	Dr. Rong Zheng
NUMBER OF PAGES:	xxi, 158

Lay Abstract

Millimeter wave radars have been widely used in both civilian and military applications. In this thesis, we begin by improving the target localization accuracy. Then, we build a real-world robotic rotating radar system. Next, we propose an efficient imaging approach and apply it on the built system for indoor environment detection. Last, an autofocusing algorithm for rotating radar imaging is given to improve the quality of the obtained images.

Abstract

Millimeter wave radars have been widely used in military reconnaissance and remote sensing, since they can acquire data in all-weather and all-day conditions. However, they still face several challenges, such as array shape limitations, high computation complexity and array manifold errors. In this thesis, we propose three new algorithms to address these challenges and achieve better performance.

First, we consider the problem of localizing multiple targets with a trapezoid virtual antenna array. The goal is to estimate both the number and the 3-D locations of the targets. The proposed algorithm consists of two steps: 1) estimating the number of targets and their ranges by extending Barone's method to handle data from multiple antennas and 2) estimating the angle of arrival of each target by a Least-Square algorithm.

Second, we propose an efficient imaging method based on robust sparse array synthesis. It first performs range-dimension matched filtering, followed by azimuth-dimension matched filtering using a selected sparse aperture and filtering weights. The aperture and weights are computed offline in advance to ensure robustness to array manifold errors caused by the imperfect radar rotation. We introduce robust constraints on the mainlobe and sidelobe levels of the filter design. The resulting robust SAS problem is a nonconvex optimization problem with quadratic constraints.

We devise an algorithm based on feasible point pursuit and successive convex approximation to solve the optimization problem.

Third, due to the unforeseen disturbance and imperfect measurement, radars may not be at their ideal locations for Synthetic Aperture Radar (SAR) imaging. We consider two error models of radar movement in Rotating SAR (ROSAR) systems. To overcome the blurring induced by location deviations of virtual phase centers, we employ an autofocusing algorithm, named Minimum Entropy Algorithm, to improve the image sharpness. The corresponding optimization problem is solved by gradient descent and interior-point methods.

To validate the effectiveness of the proposed algorithms in practice, we built a real-world radar localization system and a robotic ROSAR system. Experimental results show that the systems can localize the targets with higher accuracy and generate sharper SAR images compared with a 2D-FFT based algorithm and the Back Projection Algorithm, respectively.

Acknowledgements

I would like to thank my supervisor, Dr. Rong Zheng, who provided me extremely careful guidance. I also appreciate her strict requirements that helped me correct many inappropriate habits.

Then, I would like to thank my friends who have given me countless help in Canada. Their support has helped me better adapt to the life here, allowing me to focus on academic research.

Finally, I would like to thank my parents for their encouragement. Their words have equipped me with strong spiritual support in my mind when facing challenges.

Contents

List of Figures

List of Tables

Notation, Definitions, and Abbreviations

Notation

T	Number of targets
R_i, θ_i, ϕ_i	Polar coordinates of target i: range, azimuth angle, elevation angle
x_i, y_i, z_i	Cartesian coordinates of target i
$x(t)$	The transmitted waveform (chirp signal) from the virtual transmitter antenna at time t
f_c	Carrier frequency
λ	Carrier wavelength
K	Chirp slope
c	The speed of the light
F_s	Fast-time sampling frequency

$N_{\mathbf{SAMP}}$	Fast-time sampling number in one chirp
t_s	Fast-time sampling interval. $t_s = 1/F_S$
$T_{\mathbf{Start}}$	Sampling start time in one chirp
$y_n(t)$	The signal received by the n-th virtual received antenna at time t
$\omega_{x,i}$	The phase shift between the neighboring azimuth receiver antennas for target i
$\omega_{z,i}$	The phase shift between one of the virtual receiver antennas on the top row and the corresponding virtual receiver antenna below it for target i
α_i	The backscatter coefficient of target i
τ_i	The interval between the time when a chirp signal is emitted and received for target i
$v(t)$	Gaussian white noise at the receiver side
$\mathbf{a}_i$,$\mathbf{b}_i$	Steering vector of the real transmitter antennas and receiver antennas for target i
M	The total number of samples in one chirp
m	Fast time sample index
N	# of virtual phase centers per rotation round
n	The phase centers index

ϕ_n	Bore-sight direction of the n-th phase center
ϕ_t, R_t	The direction and range of target t
ϕ_v	The angle such that if $\phi_n \in (\pi/2 - \phi_v, \pi/2 + \phi_v)$, the target is visible to the n-th phase center
θ_n, R_n	The distance and direction observed from the n-th phase center to the target
$p(\cdot)$	Antenna radiation pattern
$y_{\mathbf{IF},n}(t)$	IF signal at the n-th phase center
$y_{\mathbf{IF},n}(m)$	Sampled IF signal at the n-th phase center
$y_{\mathbf{1D},n}(l)$	The data in the l-th range bin at the n-th phase center
$\mathbf{y}_{\mathbf{1D},n}$	Data vector after applying range-FFT at the n-th phase center
$\mathbf{Y}_{\mathbf{IF}}$, $\mathbf{Y}_{\mathbf{1D}}$	Valid data matrix for SAR
$\mathbf{a}(\phi; R)$	Steering vector
$F(\phi; R)$	Array pattern
$\mathbf{w}$	Weight vector
$\mathbf{e}$	Array error vector
$(\cdot)^T$	Matrix transpose
$(\cdot)^H$	Matrix conjugate-transpose

$\widehat{(\cdot)}$	Estimated value
$\lVert\cdot\rVert$	l^2-norm
$\mathbf{Re}\,(\cdot)$	Real part of a complex number
$\mathbf{c}^{i:j}$	A new vector consisting of the i-th to the j-th element of a vector $\mathbf{c}$

Definitions

Phase center The point from which the electromagnetic radiation spreads spherically outward.

Aperture A surface, near or on an antenna, on which it is convenient to make assumptions regarding the field values for the purpose of computing fields at external points. The aperture is often taken as that portion of a plane surface near the antenna, perpendicular to the direction of maximum radiation, through which the major part of the radiation passes. [6]

Array pattern

The directional (angular) dependence of the strength of the radio waves from an antenna array.

Abbreviations

ADC	Analog-to-digital converter
AoA	Angle of arrival

BPA	Back-Projection algorithm
CDF	Cumulative distribution function
CFAR	Constant false alarm rate
COTS	Commercial off-the-shelf
FFT	Fast Fourier transform
FMCW	Frequency modulated continuous wave
FoV	Field-of-view
FPP	Feasible point pursuit
IF	Intermediate frequency
LoS	Line-of-sight
MDL	Minimum description length
MEA	Minimum entropy algorithm
mmWave	Millimeter wave
MUSIC	Multiple signal classification
PRF	Pulse repetition frequency
ROSAR	Rotating synthetic aperture radar
RX	Receiver antenna
SAR	Synthetic aperture radar

SAS	Sparse array synthesis
SCA	Successive convex approximation
SNR	Signal-to-noise ratio
ToF	Time-of-flight
TX	Transmit antenna
UCA	Uniform circular array
ULA	Uniform linear array

Declaration of Academic Achievement

This thesis contains my research outcome from 2019$\sim$2023.

Chapter 1

Introduction

1.1 Motivation

When the wavelength of an electromagnetic wave falls between 1mm and 10mm, it is called a millimeter wave [18]. mmWave radios have been widely used in communications, localization, and microwave imaging, taking advantage of its properties that arise from its short wavelength. For instance, the short wavelength directly implies short-length antennas for efficient mmWave transmissions. This allows multiple antennas to be integrated into a small area, and many advanced techniques, such as MIMO and beamforming, can be realized even on small IoT devices. The frequency range of the mmWave is from 30GHz to 300GHz. Such a large bandwidth can provide higher transmission speeds in 5G communications and better range resolutions in target detection and localization. Together with the high Doppler sensitivity to track slow-moving targets, one of the most important mmWave applications – mmWave radars – provide excellent performance in determining targets' locations, speed and directions.

mmWave technologies reveal target characteristics in a unique way. Unlike visible light, which senses color information, or infrared, which senses temperature information, mmWave senses the target backscatter coefficient, which is closely related to the target's conductivity and geometry. High-conductivity materials, such as metals, have high backscatter coefficients, and vice versa. Additionally, mmWave has the advantages of both visible light and low-frequency radio signals. For example, its propagation mainly follows a line-of-sight path, its amplitude fades significantly when bouncing more than once, the beamwidth can be made very thin, and the signal can penetrate thin materials. All of these features enable highly accurate target detection and sensing through mmWave radars.

mmWave radars typically can be employed in two remote sensing tasks: multi-target localization and environment imaging, in which the MIMO technique are usually employed in different ways. To locate targets in an unknown space, the task is divided into two main steps: target number estimation and target location estimation. After the number of targets is determined, the signal reflected from each target is extracted from the received signals among different antennas. The target range, speed, and direction are then estimated by analyzing the extracted signals. Many modern radars use Frequency Modulated Continuous Waveform (FMCW) as the transmit signal. This waveform tends to result in more precise and reliable range estimation than other pulse signals through directly measuring the time-of-flight in time domain, but must occupy a wide range of frequencies. However, this is not a problem when utilizing mmWave, which has a spectrum of hundreds of gigahertz.

The second task is to image an environment, which results in a picture showing the scanned area. Although mmWave radars have small antennas and can be equipped with multiple physical antennas to determine the direction of incoming signals, their apertures are still very small, which limits the angular resolution of the radar system [61]. The smaller the aperture, the lower the angular resolution, which means the radar cannot capture the target details, such as shapes. Our radar localization results show that each target is represented by only one or a few points, which cannot reflect its actual shape or the backscatter coefficient of each part. Radars with uniform linear arrays use more antennas to achieve finer resolution, but this increases the circuit complexity and cost. Since the middle of 20th century, Synthetic Aperture Radar (SAR) [67] has become an alternative that uses fewer physical antennas. It moves the radar continuously while transmitting and receiving signals, creating a large virtual antenna array by emitting signals from different positions. This is another way to implement MIMO. As long as the surrounding environment is static, this method can provide an even higher cross-range resolution that allows distinguishing targets by their shapes. SAR has several working modes, such as “stripmap”, “spotlight” and “scan” [47], which are suitable for different application purposes. The Rotating SAR (ROSAR) mode mounts a radar on the edge of a rotationary platform. Through platform rotation, a 360° view of the surrounding environment is obtained. This is difficult to achieve with the aforementioned modes, which may require several radars facing different directions. In indoor environments, ROSAR can be used for mapping and localization in the case of fire emergencies or situations where other sensors fail due to high heat and low visibility.

Synthesizing a large virtual aperture is a great way to obtain high cross-range resolutions. It not only saves the cost of building a large array, but also enables the creation of an array manifold that meets different purposes. However, it may introduce manifold errors because the precise control of radar movements is often difficult to achieve. In actual usage, phase centers may deviate from their ideal locations due to unforeseen disturbances, and these deviations are not easy to measure. Continuing to use the presumed or ideal phase center locations will result in a blurry SAR image. Therefore, it is necessary to take extra steps to correct the errors and obtain a sharper image, which is a process called autofocusing.

Although mmWave radar is an excellent choice for target localization and imaging, several issues are still under explored.

- Issue 1: For mmWave radar target localization, algorithms estimating the number of targets, such as Constant False Alarm Rate (CFAR) detection, must be well-tuned for each scenario. To obtain the precise locations of targets, mmWave radars must capture a sufficiently large number of data samples. The time complexity of super-resolution methods for Angle of Arrival estimation, such as the MUSIC algorithm, is very high since they rely on a grid search in pseudo-spectrum. FFT-based algorithms are fast but require a regular array shape, such as linear or rectangular.

- Issue 2: Due to the highly nonlinear moving track of ROSAR, low-complexity imaging algorithms such as Range Migration Algorithm (RMA) [12] cannot be applied, as they assume linear and uniform array manifolds. Therefore, Back-Projection Algorithm (BPA) [17, 73] is typically employed. However, BPA suffers from heavy computational complexity, since the complexity is proportional to the number of the imaging area grids, the number of phase centers, and the number of the sampled radar data.

- Issue 3: Due to the unforeseen disturbances, precise and reliable control of the radar movements in the ROSAR systems is hard to achieve, which leads to the array geometry mismatch and blurred images. However, at this time, there is no ROSAR approach that is both robust and efficient.

- Issue 4: One way to handle the blurry images caused by array geometry mismatch is to estimate the mismatch followed by imaging with corrected phase center locations. However, there is currently no such approach designed specifically for ROSAR to improve the image quality for large imaging areas.

In this thesis, we aim to resolve the above issues and validate the proposed algorithms in a real-world testbed.

1.2 Contributions

The main contributions of this thesis are summarized as follows.

First, we propose a novel multi-target localization approach, which directly resolves the Issue 1. Such approach can be applied to FMCW radars with irregular antenna placement, e.g., trapezoid virtual antenna arrays. It starts with separating the received signals from multiple targets by exploiting the signal structure. Then, the signals from all antennas are cast into a special form by extending Barone's method [7]. Next, a Least-Square algorithm is used to estimate the AoA of each target. Simulation results and testbed experiments demonstrate that the proposed method outperforms the 2D-FFT algorithm in both ranging and localization. In high signal-to-noise ratio regimes, it achieves more accurate AoA estimations than both

2D-FFT and MUSIC algorithms. Furthermore, the proposed method has low computational complexity and achieves good performance with as few as 100 samples when the SNR is at 30 dB.

Second, we design and implement a real-world robotic ROSAR system, which is used as a testbed for imaging algorithm evaluation. To understand the design requirements, we analyze the array patterns of ULA and UCA in both far- and near-field scenarios. Note that UCA is a special case of ROSAR when the rotation speed is constant and there is no linear motion. A measurement study is conducted to quantify the extent of rotation and movement errors of the radar and the rover, which will give rise to array geometry mismatch and image blur.

Third, to address the 2nd and the 3rd issues, we propose a new sparse array synthesis technique to reduce the computation complexity of BPA. A key novelty of our design lies in the consideration of array geometry mismatch. To solve for the sparse complex weights of the virtual array elements, we formulate a robust constrained optimization problem and devise an algorithm based on feasible point pursuit [46] and successive convex approximation [9]. Compared with conventional methods such as BPA, the proposed method is optimal subject to sidelobe constraints and robust to a certain level of array manifold error. Besides, thanks to the symmetry of the circular array, the algorithm only needs to be executed in an offline manner for one azimuth direction per range bin in the radar coverage area, and the results can be used for range bins in any direction. The resulting sparse weights effectively reduce the number of received signals needed in BPA. To further reduce the complexity of the proposed algorithm, we perform range-dimension matched filtering by employing Fast Fourier Transform. For a specific target in space, only the signals from the appropriate range bins at each phase center are considered. The sparse array design constitutes an important step toward realizing ROSAR on mobile devices with limited in space, battery power and computation capacity. Simulation results and testbed experiments show that the proposed algorithm can reduce 90% of the total computation time and generate images with the quality comparable to that of BPA.

Fourth, we introduce an autofocusing technique to improve the ROSAR image quality (Issue 4) and to mitigate the imperfect control and measurement of radar locations. Based on the measurement study in Contribution 2, we consider two error models for stationary and moving ROSAR systems. In the first model, the errors are caused by unstable radar rotation, resulting in deviations of each virtual phase center from its intended position. In the second model, the errors arise from the deviation of the rover speed and the moving direction. If we use the ideal locations of the virtual phase centers for imaging, the final SAR image may blur. To address this issue, we employ the Minimum Entropy Algorithm to obtain a sharper image. Simulation results demonstrate the effectiveness of the autofocusing algorithm for ROSAR.

1.3 Organization

The rest of this thesis is organized as follows:

- Chapter 2: We present the background on mmWave radars and the related works on multiple-target localization and ROSAR.

- Chapter 3: We describe a new multiple-target localization algorithm using mmWave radars with trapezoid virtual antenna arrays.

- Chapter 4: We present the design and implementation of a real-world robotic

ROSAR system and quantify its performance through measurement studies.

- Chapter 5: A robust efficient sparse array synthesis for ROSAR is presented and evaluated.
- Chapter 6: A MEA-based autofocusing algorithm for ROSAR imaging is described together with simulation results.
- Chapter 7: We conclude the thesis and propose several directions for future works.

Chapter 2

Background

2.1 Commercial Off-The-Shelf mmWave Radar Hardware

Radar technologies have evolved significantly over the past century. Depending on the application scenario, radars have been used for detection and search, targeting (fire-control), navigation, mapping, and more. Radars can also be categorized by the antenna placement (monostatic, bistatic, or multistatic), the signal waveform, the signal wavelength, the antenna scanning type, etc. This thesis focuses on the civil applications of millimeter-wave (mmWave) monostatic radars. Several technology companies have developed commercial off-the-shelf (COTS) mmWave radars in this category, such as Texas Instruments (TI), Infineon, Qualcomm, IMEC, etc. Table 2.1 summarizes their main products and specifications. Most mmWave radars have integrated multiple antennas for transmission and reception, which facilitate the determination of target directions.

Table 2.1: COTS mmWave Radars

Provider	Name	Application	Frequency	# of TX	# of RX
Taxes Instruments	IWR1443	Industrial, short-range detection	76-81 GHz	3	4
	IWR1642			2	
	IWR1843			3	
	IWR6843		60-64 GHz		
	IWR6443				
Infineon	BGT24MTR12	Tracking, angle and direction of movement detection	24-24.25 GHz	1	2
	BGT24LTR22			2	
	BGT24MTR11		24-26 GHz	1	1
	BGT24MR2		24-24.25 GHz		
	BGT24LTR11N16				
Qualcomm	QTM527	5G communication & detection	24.25-27.5, 26.5-29.5 GHz, 27.5-28.35 GHz, 37-40GHz	2	2
	QTM052		26.5-29.5 GHz, 27.5-28.35 GHz, 37-40 GHz		
IMEC	60GHz radar	Contactless health tracking	60-66 GHz	1	1

	79GHz radar	Autonomous driving	79-81 GHz	2	2
	140GHz radar	Gesture recognition	140-150 GHz	1	1
Silicon	24 GHz radar Transceiver	angle and direction detection	23.2-26.3 GHz	2	4
	120 GHz Transceiver TRX_120_001	High-accuracy distance and speed measurement	119.1-125.9 GHz	4	
	120 GHz Transceiver TRX_120_002		119.1-125.4 GHz		
Calterah	CAL60S344-AE	In-car	59-64 GHz	4	4
	CAL77S344-AE	In-car	76-81 GHz		
Arbe	Lynx	Surround imaging radar	15.2-18.2 GHz	24	12
	Phoenix	Perception radar	76-81 GHz	48	48
INRAS	Radarbook2	Range & Doppler laboratories	10 GHz, 24 GHz, 77 GHz	8	16
Pharrowtech	PTM1060	Beamforming transceiver solution	57-71 GHz	32	32

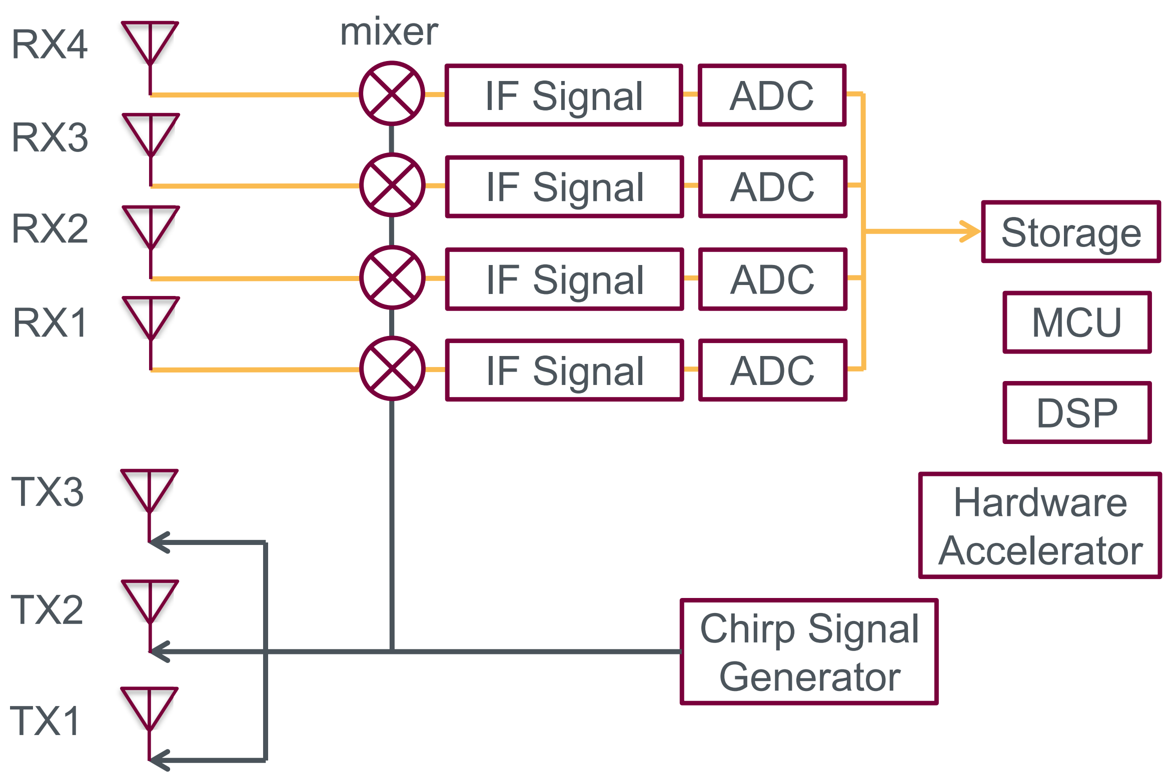


Figure 2.1: MIMO Radar System Architecture

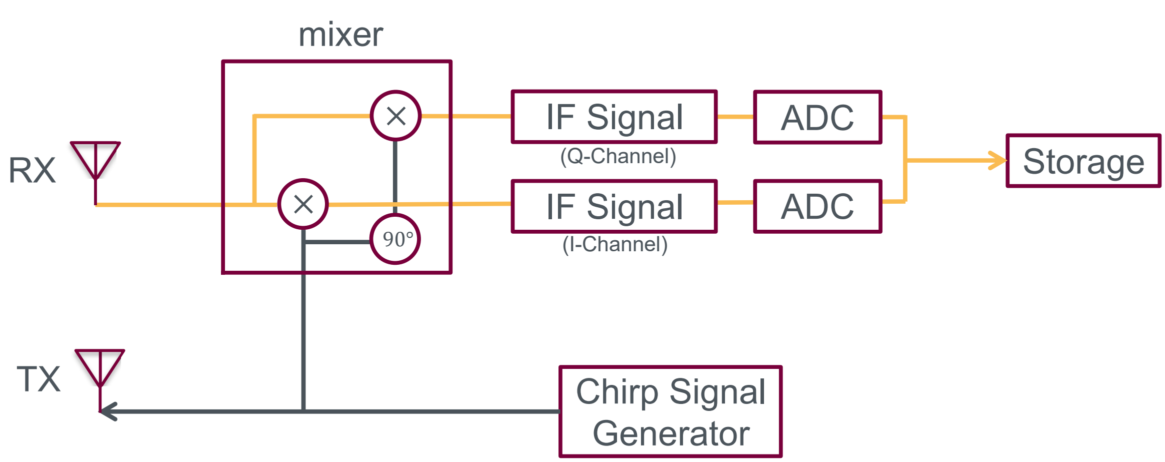
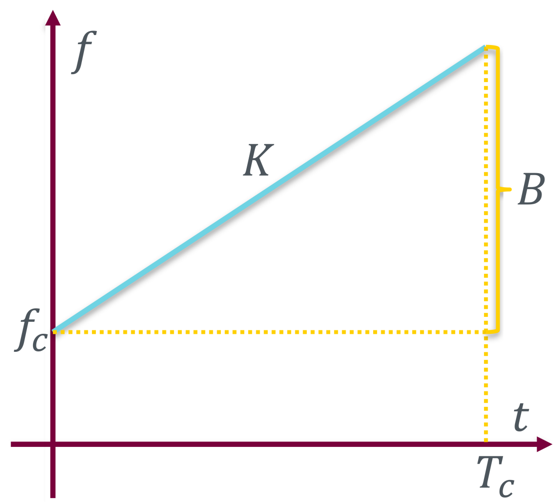
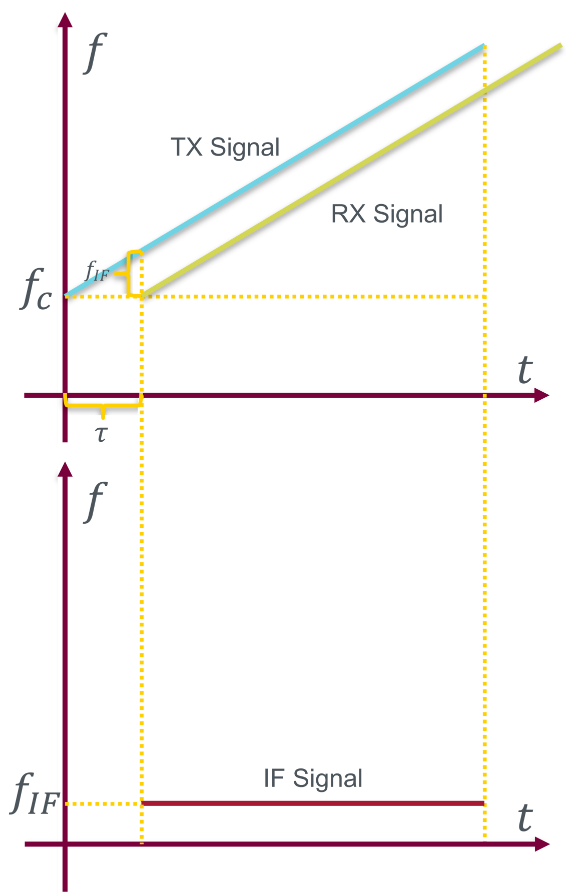


Figure 2.2: The Principle of a Mixer



(a) Chirp Signal



(b) IF Signal ($f_{\mathrm{IF}} = \tau K$)

Figure 2.3: Chirp Signal & Dechirp

2.2 Radar Signal Model and Processing

A MIMO FMCW radar example is shown in Figure 2.1. To maintain orthogonality in transmission channels, three transmit antennas operate in the time division multiplexing (TDM) mode. They take turns emitting the same signal waveform from a chirp signal generator during their designated time slots. Four receiver antennas capture the reflected signal, which is fed into mixers with the transmitted chirp signal to generate Intermediate Frequency (IF) signals. After being sampled by ADCs, the IF signal is stored for further processing. In addition, COTS radars usually equip with Micro Controller Unit (MCU), Digital Signal Processor (DSP) and hardware accelerator modules for on-board signal processing. Some advanced radars can produce complex IF signals (See Fig. 2.2), where the received signal is mixed with the original transmitted signal as well as the transmitted signal with an extra 90° phase, respectively. The resulted signals are put in the I-Channel and Q-Channel for sampling, which represent the real part and imaginary part of the IF signals.

Consider a chirp signal (shown in Figure 2.3a) transmitted by an antenna represented as

$$x(t) = e^{j2\pi\left(f_c t + \frac{1}{2}Kt^2\right)}, \tag{2.2.1}$$

where f_c is the carrier frequency; B is the bandwidth; T_c is the duration of one chirp signal, and K is the slope of the chirp. After being reflected from a target, the signal received by an antenna is given by

$$y(t) = \alpha e^{j2\pi\left[f_c(t-\tau) + \frac{1}{2}K(t-\tau)^2\right]} + v(t), \tag{2.2.2}$$

where α is the magnitude of the received signal; $\tau = \frac{2R}{c}$ is the round-trip delay, R is the distance between the target and the radar, $v(t)$ is the noise term. After being processed by the matched filter in a mixer, the output is the IF signal given by

$$\begin{aligned} y_{\mathrm{IF}}(t) &= y(t)^* \cdot x(t) \\ &= \alpha e^{j2\pi\left[\tau K t + \left(f_c\tau - \frac{1}{2}K\tau^2\right)\right]} + v_{\mathrm{IF}}(t) \\ &\approx \alpha e^{j2\pi(\tau K t + f_c\tau)} + v_{\mathrm{IF}}(t), \end{aligned} \tag{2.2.3}$$

where $v_{\mathrm{IF}}(t) = v(t)^* \cdot x(t)$. The principle of a mixer is shown in Fig. 2.3b. The IF signal must be sampled before further processing. Let the sampling frequency, the total number of samples, sampling interval, sampling start time in one chirp be F_s, M, t_s ($t_s = 1/F_s$), T_{Start}, respectively. The sampled IF signal is given by

$$y_{\mathrm{IF}}(m) = \alpha e^{j2\pi[\tau K(mt_s + T_{Start}) + f_c\tau]} + v_{\mathrm{IF}}(m), \tag{2.2.4}$$

where m is the sampling index, $0 \leq m \leq M-1$. Collecting all sampled data, we have

$$\mathbf{y}_{\mathrm{IF}} = \left[y_{\mathrm{IF}}(0), y_{\mathrm{IF}}(1), \ldots, y_{\mathrm{IF}}(M-1)\right]. \tag{2.2.5}$$

For simplicity, we omit the noise term. To measure the range R of a target, take FFT to $\mathbf{y}_{\mathrm{IF}}$, we have

$$\begin{aligned} Y_{1D}(l) &= \sum_{m=0}^{M-1} y_{\mathrm{IF}}(m) e^{-j2\pi\frac{l}{L}m} \\ &= \alpha e^{j2\pi(\tau K T_{Start} + f_c\tau)} \sum_{m=0}^{M-1} e^{j2\pi\left(\tau K t_s - \frac{l}{L}\right)m}, \end{aligned} \tag{2.2.6}$$

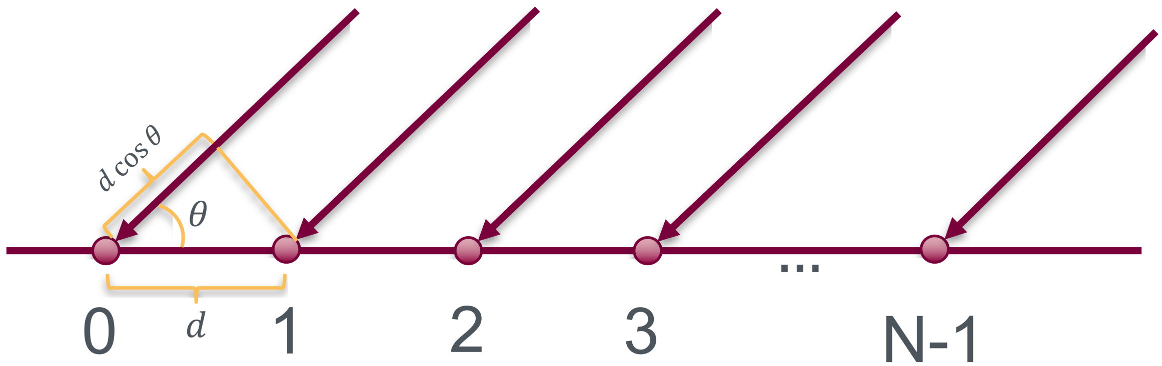


Figure 2.4: Multiple Antenna Direction Estimation

where $l = 0, 1, \ldots, L-1$ and L is the number of FFT points. Y_{1D} reaches the maximum magnitude for

$$l^{\star} = round\left(\frac{2RKt_sL}{c}\right), \tag{2.2.7}$$

and the estimated range is given by

$$\widetilde{R} = \frac{l^{\star}c}{2Kt_sL}. \tag{2.2.8}$$

For the FFT-based estimation [5], the range resolution and the maximum range are given by

$$R_{\Delta} = \frac{c}{2B}, \quad R_{max} = \frac{F_s c}{4K}. \tag{2.2.9}$$

2.2.1 Array Processing

To estimate the direction θ of a target (also called angle-of-arrival or AoA), we must deploy multiple antennas to construct an array. In Fig. 2.4, there are N antennas placed in a line with the same interval d. When the target is in the far field of a radar, we can assume that the wavefront of the reflected signals is planar and perpendicular to the propagation direction. Due to the closeness between neighboring antennas, their received signals differ by a phase (less than π). Let the distance between neighbouring antennas be d. The range difference is $d\cos\theta$. Thus, the IF signal received at the n-th antenna can be represented as

$$\begin{aligned} y_{\mathrm{IF},n}(m) &= \alpha e^{j2\pi[(\tau+nd\cos\theta/c)K(mt_s+T_{Start})+f_c(\tau+nd\cos\theta/c)]} \\ &\approx \alpha e^{j2\pi[\tau K(mt_s+T_{Start})+f_c\tau]}e^{j2\pi\cdot nd\cos\theta/\lambda}, \end{aligned} \tag{2.2.10}$$

where $\lambda = f_c/c$ is the carrier wavelength. Since $nd\cos\theta/c \ll \tau$, $nd\cos\theta/c$ has almost no impact on the IF signal frequency or the extra phase term and can be omitted. The received signals from all the antennas are given by

$$\mathbf{y}_{\mathrm{IF}} = \left[y_{\mathrm{IF},0}(m), y_{\mathrm{IF},1}(m), \ldots, y_{\mathrm{IF},N-1}(m)\right]. \tag{2.2.11}$$

By taking the FFT of $\mathbf{y}_{\mathrm{IF}}$, we have

$$\begin{aligned} Y_{1D,n}(l) &= \sum_{n=0}^{N-1} y_{\mathrm{IF},n}(m)e^{-j2\pi\frac{l}{L}n} \\ &= \alpha e^{j2\pi[\tau K(mt_s+T_{Start})+f_c\tau]}\sum_{m=0}^{M-1} e^{-j2\pi\left(\frac{d\cos\theta}{\lambda}-\frac{l}{L}\right)n}. \end{aligned} \tag{2.2.12}$$

$Y_{1D,n}(l)$ reaches its maximum magnitude when

$$l^{\star} = round\left(\frac{dL\cos\theta}{\lambda}\right), \tag{2.2.13}$$

and the estimated target direction is given by

$$\widetilde{\theta} = \arccos \frac{l^{\star}\lambda}{dL}. \tag{2.2.14}$$

For FFT-based estimation [55], the best angle resolution, which appears at the array boresight direction, is given by

$$\theta_{\Delta} = \frac{2}{N} \tag{2.2.15}$$

2.3 Radar Imaging

Radar imaging is another important application of radars. It aims to create a 2D map of an area with high resolution. The high map resolution depends on a large radar aperture size, which also enables us to obtain more target details, such as the shape and the backscatter coefficients of different parts. There are various methods to realize large apertures for different imaging purposes. For example, synthetic aperture radar (SAR) is used to image a static area by mounting a radar on a moving platform, such as a plane or a rover [48]. The movement of the platform creates a large virtual antenna array that enhances the resolution. Conversely, inverse synthetic aperture radar (ISAR) can image moving targets from a stationary radar [44]. The large aperture is constructed by collecting reflected signals from different target locations. ISAR is commonly used for detecting and identifying aircraft and satellites. Recently, several new methods have been proposed, such as mono-pulse radar 3-D imaging [70], 4D imaging radar [78] and so on.

In SAR, there are different working modes that are used to image areas. The simplest mode is the "stripmap" mode [52], where the radar is fixed on a vehicle and the antenna facing is usually perpendicular to the moving path. This mode senses a strip of the environment as the vehicle moves. The “spotlight” [8] mode adjusts the antenna bore-sight direction in real time to focus on a specific area. This mode can achieve a higher resolution for that area. The “scan” mode [47] has more complex antenna bore-sight adjustments, as it aims to cover a wider area than the “stripmap” mode, but at the cost of lower resolution.

With different working modes, many SAR imaging algorithms have also been proposed. They can be divided into two categories: (1) frequency domain-based algorithms, such as the Range-Doppler Algorithm (RDA), Chirp Scaling Algorithm [53] and Omega-K [79]; and (2) time domain-based algorithms, such as the Back-Projection Algorithm [17, 73]. The choice of algorithms depend on the working mode and the signal model of the SAR system. Frequency domain-based algorithms are popular and efficient, as they can use the fast Fourier Transform to speed up the computation. However, they assume that the radar moves linearly and transmits/receives signals at a constant rate. When these requirements are not satisfied, time domain-based algorithms are often employed, e.g., BPA.

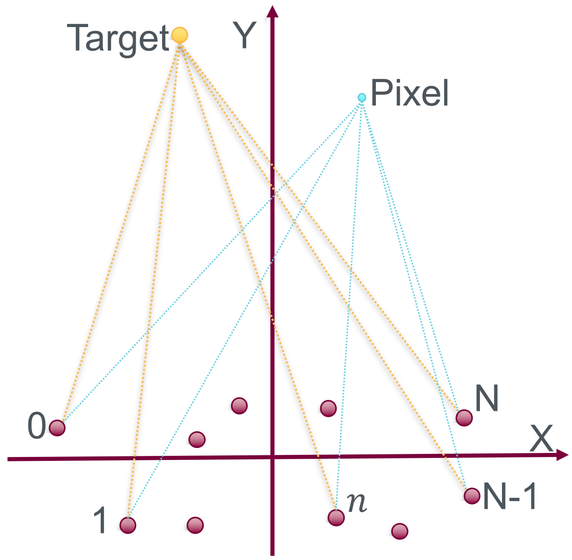


Figure 2.5: Back Projection Algorithm

The most prominent advantage of BPA is that it does not impose any restriction on the array shape (or equivalent radar movement trajectories). Consider the radar imaging system shown in Fig. 2.5. In the figure, the red dots represent N virtual phase centers, the yellow dot represents the target, and the blue dot represents a pixel in 2D space. Let (x_t, y_t), (x, y), (x_n, y_n), $n = 0, 1, \ldots, N-1$ denote the coordinates of the target, a pixel and the n-th phase center, respectively. All (x_n, y_n)'s are known a priori. The sampled IF signal received at the n-th phase center is given by

$$\mathbf{y}_{\mathrm{IF},n} = \begin{bmatrix} \alpha_n e^{j2\pi[\tau_n K(0 \cdot t_s + T_{Start}) + f_c \tau_n]} \\ \alpha_n e^{j2\pi[\tau_n K(1 \cdot t_s + T_{Start}) + f_c \tau_n]} \\ \vdots \\ \alpha_n e^{j2\pi[\tau_n K((M-1) \cdot t_s + T_{Start}) + f_c \tau_n]} \end{bmatrix} + \mathbf{v}_{\mathrm{IF}}, \tag{2.3.1}$$

where $\tau_n = 2R_{t,n}/c$; $R_{t,n} = \sqrt{(x_n - x_t)^2 + (y_n - y_t)^2}$; $\mathbf{v}_{\mathrm{IF}} = [v_{\mathrm{IF}}(0), v_{\mathrm{IF}}(1), \ldots, v_{\mathrm{IF}}(M-1)]^T$.

Collecting data from all the phase centers, we have

$$\mathbf{Y}_{\mathrm{IF}} = \left[\mathbf{y}_{\mathrm{IF},0}, \mathbf{y}_{\mathrm{IF},1}, \ldots, \mathbf{y}_{\mathrm{IF},N-1}\right]. \tag{2.3.2}$$

In BPA, to image a pixel at (x, y), we construct a matrix $\mathbf{W}$ of size $M \times N$, where the element at the intersection of the i-th row and j-th column is

$$\mathbf{W}_{(i,j)} = e^{-j2\pi[\widetilde{\tau}_n K((M-1)\cdot t_s + T_{Start}) + f_c \widetilde{\tau}_n]}, \tag{2.3.3}$$

where $\widetilde{\tau}_n = \sqrt{(x_n - x)^2 + (y_n - y)^2}$. Then, the intensity of the pixel is

$$I(x, y) = \mathbf{1}^T \cdot (\mathbf{W} \odot \mathbf{Y}_{\mathrm{IF}}) \cdot \mathbf{1}, \tag{2.3.4}$$

where $\odot$ denotes the Hadamard product. From the above formula, we can find that if (x, y) is the same as (x_t, y_t), all the phase terms in $\mathbf{Y}_{\mathrm{IF}}$ can be perfectly cancelled and the intensity reaches its maximum value.

Chapter 3

Multiple-Target Localization by Millimeter-Wave Radars with Trapezoid Virtual Antenna Arrays

3.1 Introduction

In Section 2.2.1, we have explored the use of array processing in determining the AoA of a signal. In fact, array processing can also be applied to determine target number and locations, improve SNR [26], track multiple targets [64], image the areas of interest and so on. In this chapter, we only focus on the target number and location estimations, and with very few exceptions, their estimations are performed in two separate steps in existing work. To estimate the number of targets, a variety of methods have been devised, including those using Akaike Information Criterion (AIC) [1], energy detection [21], minimum decription length [66], machine learning models [38], etc. Once the number of targets is known, targets are separated according to their ranges and angles. To estimate the angle of arrivals of known number of targets, multi-antenna systems, such as uniform linear array [40] and uniform rectangular array (URA), are used and many algorithms have been proposed, including Estimation of Signal Parameters via Rotational Invariant Techniques (ESPRIT) [23, 33], MUltiple SIgnal Classification (MUSIC) [57], Capon algorithm [14, 68], PARAllel FACtor analysis (PARAFAC) based algorithm [77], matrix pencil method [31] and its enhancement [30, 15].

In addition to the above general array processing approaches, custom algorithms have been designed to localize multiple targets using mmWave radars. In [32], the authors apply 1D-FFT, constant false alarm rate detection and peak grouping to the data matrix extracted from a radar to estimate the number of targets and their ranges. An angle-FFT is then applied to estimate the angles of remaining peaks, each corresponding to one target. Due to its usage of FFTs in range and angle domains, we call this approach “2D-FFT”. In [41], a modified 3D MUSIC algorithm was developed for 3D target localization. Instead of choosing the largest few eigenvalues as the number of targets, the authors adopt the Minimum Description Length (MDL) criterion [65, 29]. The polar coordinates of all targets are estimated from the radar 3D MUSIC pseudo-spectrum jointly.

These approaches suffer from several limitations. In the FFT-based algorithm in [32], the number of sampled data from each received antenna has to be sufficiently large making the detection and localization of moving targets difficult. Moreover, the CFAR threshold needs to be carefully tuned for different target environments [56]. 1-D angular FFT assumes linear antenna arrays. Furthermore, the time complexity of super-resolution methods for AoA estimation such the MUSIC algorithm is very high since they rely on a grid search in the pseudo-spectrum, which diminishes their applicability in real-time applications.

In this chapter, we introduce a new approach for multi-target localization using mmWave radars with trapezoid virtual antenna arrays. Trapezoid antenna arrays become increasingly common in COTS radar boards, e.g., IWR1443/1843/6843 from Taxes Instruments [54], CAL77S244 from Calterah Semiconductor Technology [63], to name a few in indoor and vehicular applications. In the proposed method, we first separate received signals from multiple targets by exploiting the received signal structure. We then cast the received signals on each antenna of FMCW radars into the form of complex moments in Barone's method [7] and extend this method to handle data from multiple antennas. Next, a Least-Square algorithm is employed to estimate the AoAs of each target. Simulation results and testbed experiments show that the proposed method not only outperforms the 2D-FFT algorithm in ranging and localization, but also achieves more accurate AoA estimations than both 2D-FFT and MUSIC algorithms in the high SNR regimes. In addition, the proposed method reaches this improved performance with as few as 100 samples when the SNR is 30dB.

The rest of the chapter is organized as follows. Section 3.2 derives the radar system model for a trapezoid virtual antenna array; Section 3.3 presents the proposed approach to extend Barone's method and apply Least Square algorithm in radar signal processing; Section 3.4 compares the MATLAB simulation results by using 2D-FFT, MUSIC and the proposed method. Section 3.5 shows the evaluation of the proposed algorithm on a testbed followed by a conclusion in Section 3.6.

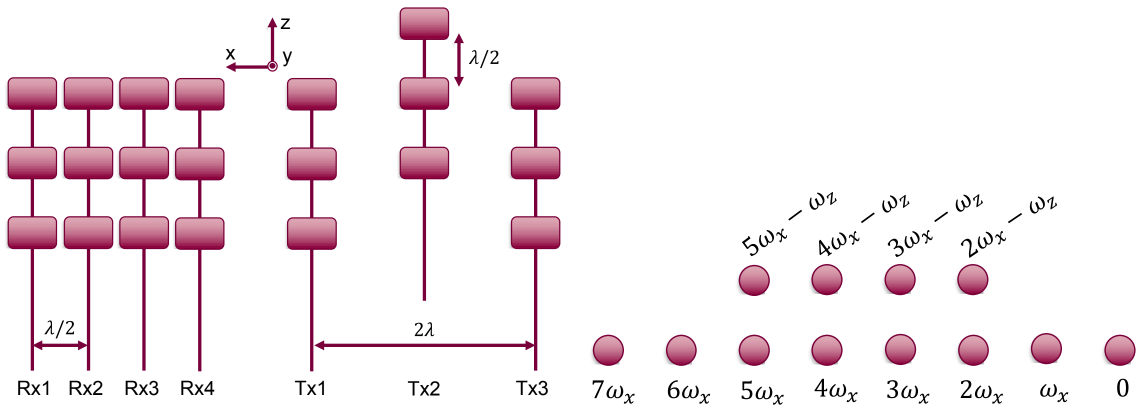
3.2 Radar Model for Multiple Targets

In this section, we present the antenna geometry of the radar we use and the mathematical representations of transmitted and received signals.

3.2.1 System Model

For concreteness of the discussion, we consider a specific radar geometry (see Figure 3.1a), where the distance between neighboring transmit antennas is λ, the carrier wavelength, and the middle transmit antenna is $\lambda/2$ higher than its two neighbors. The distance between neighboring receiver antennas is also $\lambda/2$. In order to separate transmitted signals, we let three transmit antennas take turns to transmit the same chirp signals, in the order of TX1, TX3, and TX2. Then, the reflected signals obtained on all the receiver antennas from the same target exhibit phase shifts from one another, caused by the distance differences between the target and each antenna. Equivalently, we can represent the system with $M = 1$ virtual transmit antenna located at TX1 and $N = 12$ virtual receiver antennas, among which RX1∼4 are located at their original positions, and two replications of the virtual antenna array are located at the left side and on top of original 4 antennas, respectively (see Figure 3.1b). Thus, we call it “trapezoid virtual antenna array”. The 8 virtual antennas in the bottom row are also called azimuth receiver antennas. This transformation greatly simplifies subsequent analysis.

Consider T targets located at (R_i, θ_i, ϕ_i) or (x_i, y_i, z_i), $i = 1,\ 2,\ 3, \ldots,\ T$ (shown in Figure 3.2). Based on the above trapezoid array, the two phase shifts $\omega_{x,i}$ and $\omega_{z,i}$



(a) Antenna Placement and Spacing (b) Trapezoid Virtual Antenna Array

Figure 3.1: Antenna Geometry

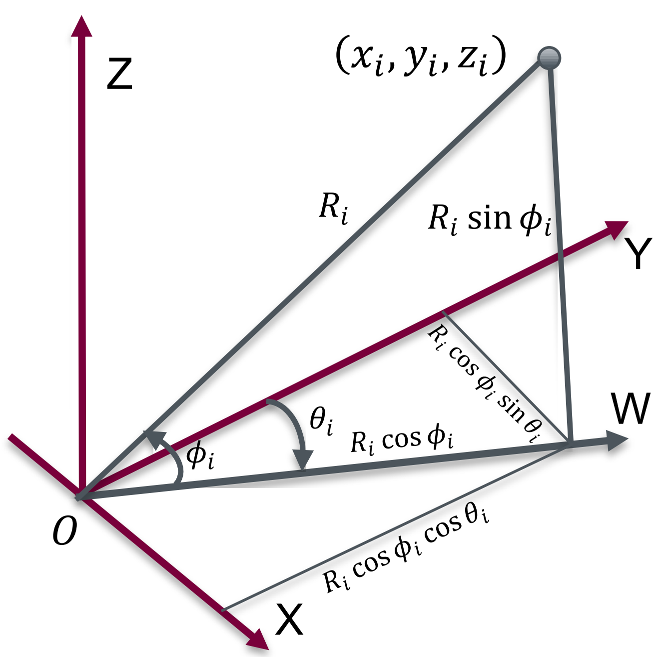


Figure 3.2: Radar and Target Geometry

can be derived by azimuth and elevation angles between the radar and the target as:

$$\omega_{x,i} = \pi \sin\theta_i \cos\phi_i, \tag{3.2.1}$$

$$\omega_{z,i} = \pi \sin\phi_i. \tag{3.2.2}$$

In fact, the phase shifts can be represented using the steering vectors for the real transmit antennas and receiver antennas, denoted by $\mathbf{a}_i\left(\omega_{x,i}, \omega_{z,i}\right) = \left[1, e^{j4\omega_{x,i}}, e^{j(2\omega_{x,i}-\omega_{z,i})}\right]^T$, $\mathbf{b}_i\left(\omega_{x,i}\right) = \left[1, e^{j\omega_{x,i}}, e^{j2\omega_{x,i}}, e^{j3\omega_{x,i}}\right]^T$. Then, the equivalent steering vector for all 12 virtual receiver antennas is given by,

$$\begin{aligned}
\mathbf{h}_i &= \mathbf{a}_i\left(\omega_{x,i}, \omega_{z,i}\right) \bigotimes \mathbf{b}_i\left(\omega_{x,i}\right) \\
&= [1, e^{j\omega_{x,i}}, e^{j2\omega_{x,i}}, e^{j3\omega_{x,i}}, e^{j4\omega_{x,i}}, e^{j5\omega_{x,i}}, e^{j6\omega_{x,i}}, e^{j7\omega_{x,i}}, \\
&\quad e^{j(2\omega_{x,i}-\omega_{z,i})}, e^{j(2\omega_{x,i}-\omega_{z,i})}e^{j\omega_{x,i}}, e^{j(2\omega_{x,i}-\omega_{z,i})}e^{j2\omega_{x,i}}, \\
&\quad e^{j(2\omega_{x,i}-\omega_{z,i})}e^{j3\omega_{x,i}}]^T,
\end{aligned} \tag{3.2.3}$$

where $\bigotimes$ denotes the Kronecker product. Now, let $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_T]$, $\boldsymbol{\omega}_{\mathbf{x}} = [\omega_{x,1}, \omega_{x,2}, \ldots, \omega_{x,T}]$, $\boldsymbol{\omega}_{\mathbf{z}} = [\omega_{z,1}, \omega_{z,2}, \ldots, \omega_{z,T}]$, $\boldsymbol{\tau} = [\tau_1, \tau_2, \ldots, \tau_T]$. We can write the received signal by all virtual antennas from all T targets as:

$$\mathbf{y}\left(t; \boldsymbol{\alpha}, \boldsymbol{\omega}_{\mathbf{x}}, \boldsymbol{\omega}_{\mathbf{z}}, \boldsymbol{\tau}\right) = \sum_{i=1}^{T} \alpha_i \mathbf{h}_i x\left(t-\tau_i\right) + \mathbf{v}\left(t\right), \tag{3.2.4}$$

where $\mathbf{v}\left(t\right)$ is a vector of Gaussian white noise at receiver side and each element in $\mathbf{v}\left(t\right)$ follows $v\left(t\right) \sim \mathcal{N}\left(0, \sigma^2\right)$. Let $\mathbf{A} = diag\left([\alpha_1, \alpha_2, \ldots, \alpha_T]\right)$, $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \ldots, \mathbf{h}_T]$, $\mathbf{x}\left(t; \boldsymbol{\tau}\right) = \left[x\left(t-\tau_1\right), x\left(t-\tau_2\right), \cdots, x\left(t-\tau_T\right)\right]^T$. The system model of the trapezoid virtual antenna array in matrix representation is given by,

$$\mathbf{y}\left(t;\boldsymbol{\alpha},\boldsymbol{\omega}_{\mathbf{x}},\boldsymbol{\omega}_{\mathbf{z}},\boldsymbol{\tau}\right)=\mathbf{H}\mathbf{A}\mathbf{x}\left(t;\boldsymbol{\tau}\right)+\mathbf{v}\left(t\right). \tag{3.2.5}$$

3.2.2 IF Signals

From the system model, we can analyze the waveform of the received signals for FMCW radars. Assume that the chirp signal emitted by a TX antenna is $x\left(t\right)=e^{j2\pi\left(f_c t+\frac{1}{2}Kt^2\right)}$. The received signals in a vector form at the 12 virtual antennas are

$$\mathbf{y}\left(t;\boldsymbol{\alpha},\boldsymbol{\omega}_{\mathbf{x}},\boldsymbol{\omega}_{\mathbf{z}},\boldsymbol{\tau}\right)=\sum_{i=1}^{T}\alpha_i\mathbf{h}_i e^{j2\pi\left[f_c(t-\tau_i)+\frac{1}{2}K(t-\tau_i)^2\right]}+\mathbf{v}\left(t\right). \tag{3.2.6}$$

After being processed by a matched filter, the resulting intermediate frequency (IF) signal is

$$\begin{aligned}\mathbf{y}_{\mathrm{IF}}\left(t;\boldsymbol{\alpha},\boldsymbol{\omega}_{\mathbf{x}},\boldsymbol{\omega}_{\mathbf{z}},\boldsymbol{\tau}\right) &= \mathbf{y}\left(t;\boldsymbol{\alpha},\boldsymbol{\omega}_{\mathbf{x}},\boldsymbol{\omega}_{\mathbf{z}},\boldsymbol{\tau}\right)^{*}\cdot x\left(t\right)\\ &= \sum_{i=1}^{T}\alpha_i\mathbf{h}_i^{*}e^{j2\pi\left[\tau_i Kt+\left(f_c\tau_i-\frac{1}{2}K\tau_i^2\right)\right]}+\mathbf{v}_{\mathrm{IF}}\left(t\right),\end{aligned}$$

where $\mathbf{v}_{\mathrm{IF}}\left(t\right)=\mathbf{v}\left(t\right)^{*}\cdot x\left(t\right)$. Let $x_{\mathrm{IF}}\left(t;\tau_i\right)=e^{j2\pi\left[\tau_i Kt+\left(f_c\tau_i-\frac{1}{2}K\tau_i^2\right)\right]}$ and $\mathbf{x}_{\mathrm{IF}}\left(t;\boldsymbol{\tau}\right)=\left[x_{\mathrm{IF}}\left(t;\tau_1\right),x_{\mathrm{IF}}\left(t;\tau_2\right),\cdots,x_{\mathrm{IF}}\left(t;\tau_T\right)\right]$. We can write the output IF signals for all 12 virtual antennas in a matrix form as

$$\mathbf{y}_{\mathrm{IF}}\left(t;\boldsymbol{\alpha},\boldsymbol{\omega}_{\mathbf{x}},\boldsymbol{\omega}_{\mathbf{z}},\boldsymbol{\tau}\right)=\mathbf{H}^{*}\mathbf{A}\mathbf{x}_{\mathrm{IF}}\left(t;\boldsymbol{\tau}\right)+\mathbf{v}_{\mathrm{IF}}\left(t\right). \tag{3.2.7}$$

3.2.3 Sampling

Define $x_{\text{IF,SAMP}}(n;\tau_i) = e^{j2\pi\left[\tau_i K(nt_s+T_{\text{Start}})+\left(f_c\tau_i-\frac{1}{2}K\tau_i^2\right)\right]}, n = 0,\ 1,\ 2,\ \ldots,\ N_{\text{SAMP}}-1$ and let

$$\mathbf{x}_{\text{IF,SAMP}}(n;\boldsymbol{\tau}) = \begin{bmatrix} x_{\text{IF,SAMP}}(n;\tau_1) \\ x_{\text{IF,SAMP}}(n;\tau_2) \\ \vdots \\ x_{\text{IF,SAMP}}(n;\tau_T) \end{bmatrix}. \tag{3.2.8}$$

The n-th sampled data for all 12 virtual antennas can be written as

$$\mathbf{y}_{\text{IF,SAMP}}(n;\boldsymbol{\alpha},\boldsymbol{\omega}_{\mathbf{x}},\boldsymbol{\omega}_{\mathbf{z}},\boldsymbol{\tau}) = \mathbf{H}^*\mathbf{A}\mathbf{x}_{\text{IF,SAMP}}(n;\boldsymbol{\tau}) + \mathbf{v}_{\text{IF}}(n). \tag{3.2.9}$$

Next, let

$$\begin{aligned}\mathbf{X}_{\text{IF}}(\boldsymbol{\tau}) &= \left[\mathbf{x}_{\text{IF,SAMP}}(0;\boldsymbol{\tau}),\ldots,\mathbf{x}_{\text{IF,SAMP}}(N_{\text{SAMP}}-1;\boldsymbol{\tau})\right] \\ &= \begin{bmatrix} x_{\text{IF,SAMP}}(0;\tau_1),\ldots,x_{\text{IF,SAMP}}(N_{\text{SAMP}}-1;\tau_1) \\ x_{\text{IF,SAMP}}(0;\tau_2),\ldots,x_{\text{IF,SAMP}}(N_{\text{SAMP}}-1;\tau_2) \\ \vdots \\ x_{\text{IF,SAMP}}(0;\tau_T),\ldots,x_{\text{IF,SAMP}}(N_{\text{SAMP}}-1;\tau_T) \end{bmatrix}.\end{aligned}$$

We can then write the sampled radar matrix $\mathbf{Y}_{\text{IF}}$ of size $12 \times N_{\text{SAMP}}$ as

$$\mathbf{Y}_{\text{IF}}(\boldsymbol{\alpha},\boldsymbol{\omega}_{\mathbf{x}},\boldsymbol{\omega}_{\mathbf{z}},\boldsymbol{\tau}) = \mathbf{H}^*\mathbf{A}\mathbf{X}_{\text{IF}}(\boldsymbol{\tau}) + \mathbf{V}_{\text{IF}}. \tag{3.2.10}$$

Now, our problem can be stated as estimating T, $\boldsymbol{\omega}_{\mathbf{x}}$, $\boldsymbol{\omega}_{\mathbf{z}}$, $\boldsymbol{\tau}$ based on the sampled radar matrix $\mathbf{Y}_{\text{IF}}$.

3.3 Approach

In this section, we present the proposed algorithm and analyze its property.

3.3.1 An overview of Barone's method

Before delving into the proposed method, we first review how Barone's method works[7] and its connection to multi-target localization. The problem that Barone's method solves is to estimate parameters from a sequence data $\mathbf{y}$ of length N_{SAMP}, where $\mathbf{y} = \{y_0, y_1, y_2, \ldots, y_{N_{\text{SAMP}}-1}\}$ and y_n satisfies

$$y_n = s_n + v_n, \tag{3.3.1}$$

for $n = 0, 1, 2, \ldots, N_{\text{SAMP}} - 1$. In (3.3.1), s_n is a complex moment satisfying the following form: $s_n = \sum_{i=0}^{T} c_i \xi_i^n$, where c_i and ξ_i are complex numbers, T is a non-negative integer. v_n is the complex addictive Gaussian white noise with zero mean and known variance σ^2. To estimate T, c_i's and ξ_i's from $\mathbf{y}$, we summarize Barone's method into the following five steps:

Step 1: Build R independent pseudo replications of the original data sequence, as

$$y_n^{(r)} = y_n + v_n^{(r)}, n = 0, 1, \ldots, N_{\text{SAMP}} - 1; r = 1, \ldots, R. \tag{3.3.2}$$

where, $\{v_n^{(r)}\} \sim \mathcal{N}(0, \sigma'^2)$. Compute Padé poles and corresponding residuals based on the replicated data by Padé approximants [27]. The result is $R \cdot N_{\text{SAMP}}/2$ pairs of $(\xi_i^{(r)}, c_i^{(r)})$.

Step 2: On a proper lattice L on the complex plane, calculate the complex measure

$\widehat{S}_{N_{\text{SAMP}},R}\,(z,\widetilde{\sigma}^2)$ based on $(\xi_i^{(r)}, c_i^{(r)})$s on those lattice points. Find all the local maxima of $|\widehat{S}_{N_{\text{SAMP}},R}\,(z,\widetilde{\sigma}^2)\,|$.

Step 3: Count the number of Padé poles in the neighborhood of each local maxima.

Step 4: Select those local maxima whose number of associated Padé poles meet the prescribed threshold. The number of those local maxima is the estimated $\widehat{T}$.

Step 5: For each of those local maxima, take the average of its corresponding $c_i^{(r)}$ and $\xi_i^{(r)}$ in its neighbor area. The results are the estimated $\widehat{\xi}_i$, $\widehat{c}_i$.

Though Barone's method was originally developed as a mathematical tool in approximation theory, we find that the received signals from multiple targets in radar systems can be expressed in a form that the method can be applied. In particular, the received signal is the sum of reflected signals from unknown number of targets. Each summand of the nth sample of the reflected signal contains a target-dependent constant and a target-dependent term that changes over time. Lastly, the received signal also contains a noise term. Thus, by representing the received radar signal in the form (3.3.1), we can apply Barone's method to estimate the target parameters.

3.3.2 Transformation for individual antenna

In order to apply Barone's method, we need to transform the radar data matrix $\mathbf{Y}_{\text{IF}}$ to a suitable form as we discussed above. Define

$$c_{k,i} = \alpha_i e^{-j(k-1)\omega_{x,i}} e^{j2\pi\left(\tau_i K T_{\text{Start}} + f_c\tau_i - \frac{1}{2}K\tau_i^2\right)}, \tag{3.3.3}$$

for $k = 1, 2, 3, \ldots, 8$, and

$$c_{k,i} = \alpha_i e^{j(\omega_{z,i} - 2\omega_{x,i})} e^{-j(k-9)\omega_{x,i}} e^{j2\pi\left(\tau_i K T_{\text{Start}} + f_c\tau_i - \frac{1}{2}K\tau_i^2\right)}, \tag{3.3.4}$$

for $k = 9, 10, 11, 12$, and $i = 1, 2, 3, \ldots, T$. Let

$$\xi_i = e^{j2\pi\tau_i K t_s}. \tag{3.3.5}$$

We construct a Vandermonde matrix:

$$\begin{aligned} \mathbf{V} &= Vander\left(\xi_1, \xi_2, \ldots, \xi_T\right) \\ &= \begin{bmatrix} \xi_1^0 & \xi_1^1 & \cdots & \xi_1^{(N_{\text{SAMP}}-1)} \\ \xi_2^0 & \xi_2^1 & \cdots & \xi_2^{(N_{\text{SAMP}}-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \xi_T^0 & \xi_T^1 & \cdots & \xi_T^{(N_{\text{SAMP}}-1)} \end{bmatrix}. \end{aligned} \tag{3.3.6}$$

Now, we can write the system model (3.2.10) as

$$\begin{aligned} \mathbf{Y}_{\text{IF}} &= \mathbf{S} + \mathbf{V}_{\text{IF}} \\ &= \mathbf{C} \cdot \mathbf{V} + \mathbf{V}_{\text{IF}}, \end{aligned} \tag{3.3.7}$$

where the element in the k-th row and i-th column of the matrix $\mathbf{C}$ is $c_{k,i}$. The size of $\mathbf{C}$ and $\mathbf{V}$ is $12 \times T$ and $T \times N_{\text{SAMP}}$, respectively. For the element in the k-th row and n-th column of $\mathbf{S}$, we have $s_{k,n} = \sum_{i=1}^{T} c_{k,i}\xi_i^n$, where $k = 1, 2, 3, \ldots, 12$. And this form is exactly the same as that mentioned by Barone.

To further analyze the sampling data from each virtual antenna, we write the sampling data from each virtual antenna as a column vector. Take the transpose to both sides of (3.3.7), we have

$$\begin{aligned} \mathbf{Y}_{\mathrm{IF}}^T &= [\mathbf{y}_{\mathrm{IF},1}, \mathbf{y}_{\mathrm{IF},2}, \mathbf{y}_{\mathrm{IF},3}, \ldots, \mathbf{y}_{\mathrm{IF},12}] \\ &= \mathbf{V}^T\mathbf{C}^T + \mathbf{V}_{\mathrm{IF}}^T. \end{aligned} \tag{3.3.8}$$

The sampled data of the k-th virtual antenna $\mathbf{y}_{\mathrm{IF},k}$ is

$$\begin{aligned} \mathbf{y}_{\mathrm{IF},k} &= \mathbf{s}_k + \mathbf{v}_k \\ &= \begin{bmatrix} s_{k,0} \\ s_{k,1} \\ \vdots \\ s_{k,N_{\mathrm{SAMP}}-1} \end{bmatrix} + \begin{bmatrix} v_{k,1} \\ v_{k,2} \\ \vdots \\ v_{k,N_{\mathrm{SAMP}}-1} \end{bmatrix}, \end{aligned} \tag{3.3.9}$$

Now if we directly apply Barone's method to $\mathbf{y}_{\mathrm{IF},k}$ and get the estimated parameters for the k-th virtual antenna. The results can be represented by

- Number of Targets: $\widehat{T}_k$;
- Residuals: $\widehat{\mathbf{u}}_k = \left\{\widehat{c}_{k,1}, \widehat{c}_{k,2}, \ldots, \widehat{c}_{k,\widehat{T}_k}\right\}$;
- Padé Poles: $\widehat{\boldsymbol{\xi}}_k = \left\{\widehat{\xi}_{k,1}, \widehat{\xi}_{k,2}, \ldots, \widehat{\xi}_{k,\widehat{T}_k}\right\}$.

Repeating the above operation to the sampled data of all 12 virtual antennas, i.e., $\mathbf{y}_{\mathrm{IF},1}, \mathbf{y}_{\mathrm{IF},2}, \ldots, \mathbf{y}_{\mathrm{IF},12}$, we have

$$\widehat{\mathbf{T}} = \left[\widehat{T}_1, \widehat{T}_2, \ldots, \widehat{T}_{12}\right], \tag{3.3.10}$$

$$\widehat{\mathbf{U}} = \left\{\widehat{\mathbf{u}}_1, \widehat{\mathbf{u}}_2, \ldots, \widehat{\mathbf{u}}_{12}\right\}, \tag{3.3.11}$$

$$\widehat{\boldsymbol{\Xi}} = \left\{\widehat{\boldsymbol{\xi}}_1, \widehat{\boldsymbol{\xi}}_2, \ldots, \widehat{\boldsymbol{\xi}}_{12}\right\}. \tag{3.3.12}$$

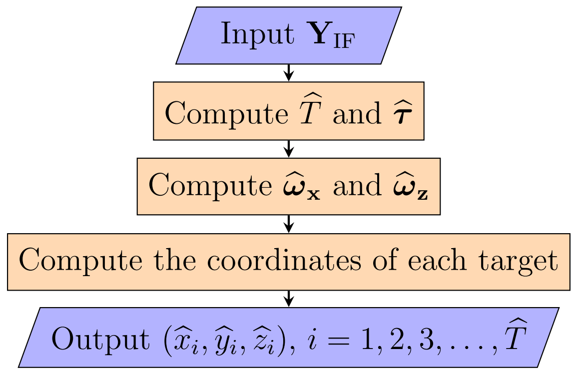


Figure 3.3: The diagram of the proposed algorithm

However, due to the existence of random noise, $\widehat{T}_1, \widehat{T}_2, \ldots, \widehat{T}_{12}$ may not be the same, and the locations of elements in $\widehat{\boldsymbol{\Xi}}$ do not coincide on the complex plane. One naive approach is to take the average of the estimated number of targets from each antenna and then perform clustering over the estimated locations. However, such a method fails to take advantage of the correlated nature of measurements at different antennas. Next, we propose a new algorithm that extends Barone's method and performs joint estimations of the number of targets and key parameters.

3.3.3 Multi-target Localization Algorithm

In the algorithm, we first extend Barone's method to multi-antenna case for estimating the number of targets and their ranges. Next, we employ a Least Square algorithm to extract all the angle information. Finally, their coordinates are calculated. Figure 3.3 shows the steps of the proposed algorithm.

Step 1: Compute $\widehat{T}$ and $\widehat{\boldsymbol{\tau}}$.

Recall the formation of the replication data in Equation (3.3.2), which can be further expressed as:

$$y_n^{(r)} = \sum_{i=0}^{T} c_i \xi_i^n + v_n + v_n^{(r)}. \tag{3.3.13}$$

Clearly, for each replicated data, adding more Gaussian white noise does not change the Padé pole ξ_i's and the magnitude of residual c_i's. From Equation (3.3.9), the n-th sampled data can be expressed as

$$y_{n,k} = \sum_{i=1}^{T} c_{k,i} \xi_i^n + v_{k,n}. \tag{3.3.14}$$

The key insight is, for different antennas (indexed by k), the only difference lies in the phase of $c_{k,i}$, while the ξ_i and the magnitude of the $c_{k,i}$ are the same among all antennas. Therefore, we can view these 12 sequences of sampled radar data (one from each antenna) as 12 replications of a data sequence whose element corresponds to

$$\overline{y}_n = \sum_{i=1}^{T} \overline{c}_i \xi_i^n. \tag{3.3.15}$$

There is no noise component in this sequence. Instead, receiver side noise $v_{k,n}$ can be seen as manually added noise (i.e., $v_k^{(r)}$ in Equation (3.3.2) for each replication. From [7], when the replication data only differs in the manually added noise, the Padé poles and the residuals remain the same. Therefore, the positions of all the ξ_i's do not change using the sampled radar data. Furthermore, the number of targets is determined by the number of local maxima that has enough estimated Padé poles in its neighbor area on the complex plane, which depends on the magnitude of corresponding residuals $|\overline{c}_i|$. Though $c_{k,i}$'s differ in phase for different antennas, their magnitude are the same. Thus, the distribution of estimated Padé poles $\xi_i^{(r)}$ remains the same. In other words, the estimated number of targets is not affected by the phase shift among antennas. Here, $\bar{c}_i$ is a placeholder and we will not compute its precise value.

Usually, people make tens of replicated data. If we only use 12 sequences of sampled radar data, the detection results may not be ideal. Thus, we can reuse some results we have already obtained to avoid making new pseudo data. We have got the $\widehat{\mathbf{T}}$, estimated Padé poles set $\widehat{\mathbf{U}}$ and corresponding residuals set $\widehat{\boldsymbol{\Xi}}$ from 12 sequences of sampled radar data. By using these parameters, we can still continue to use Step 2 to Step 5 in Section 3.3.1 to get the final estimated $\widehat{T}$ and $\widehat{\boldsymbol{\xi}}$.

For each element in $\widehat{\boldsymbol{\xi}}$, we can compute an estimated $\widehat{\tau}_i$ from Equation (3.3.5) as

$$\widehat{\tau}_i = \frac{1}{2\pi K t_s} \angle \widehat{\xi}_i. \tag{3.3.16}$$

The distance from radar to target i is computed by $\widehat{R}_i = \frac{c\widehat{\tau}_i}{2}$.

Now, we summarize the detailed Step 1 as:

Algorithm 1 Compute $\widehat{T}$ and $\widehat{\boldsymbol{\tau}}$

Input $\mathbf{Y}_{\mathrm{IF}}$;
$\widehat{\mathbf{T}} = \emptyset$, $\widehat{\mathbf{U}} = \emptyset$, $\widehat{\boldsymbol{\Xi}} = \emptyset$, $\widehat{T} = 0$, $\widehat{\boldsymbol{\tau}} = \emptyset$;
for $k \leftarrow 1$ **to** 12 **do**
  Apply Barone's method to $\mathbf{y}_{\mathrm{IF},k}$, get $\widehat{T}_k$, $\widehat{\mathbf{u}}_k$, $\widehat{\boldsymbol{\xi}}_k$;
  $\widehat{\mathbf{T}} = [\widehat{\mathbf{T}}, \widehat{T}_k]$, $\widehat{\mathbf{U}} = \widehat{\mathbf{U}} \cup \widehat{\mathbf{u}}_k$, $\widehat{\boldsymbol{\Xi}} = \widehat{\boldsymbol{\Xi}} \cup \widehat{\boldsymbol{\xi}}_k$;
end for
Continue Step 2∼5 in Section 3.3.1 based on $\widehat{\mathbf{T}}$, $\widehat{\mathbf{U}}$ and $\widehat{\boldsymbol{\Xi}}$, get $\widehat{T}$ and $\widehat{\boldsymbol{\xi}}$;
for $i \leftarrow 1$ **to** $\widehat{T}$ **do**
  Compute $\widehat{\tau}_i$ by (3.3.16);
  $\widehat{\boldsymbol{\tau}} = \widehat{\boldsymbol{\tau}} \cup \widehat{\tau}_i$;
end for

Remark 1 *Since $\angle\widehat{\xi}_i \in (0, 2\pi]$, the maximum $\widehat{\tau}_i$ is $\frac{1}{Kt_s}$, and the maximum estimated* *range of the target that can be uniquely determined satisfies* $\widehat{R}^{\max} = \frac{c}{2Kt_s}$.

Step 2: Compute $\widehat{\boldsymbol{\omega}}_{\mathbf{x}}$ and $\widehat{\boldsymbol{\omega}}_{\mathbf{z}}$.

From the estimated Padé poles $\widehat{\boldsymbol{\xi}}$, we construct a Vandermonde matrix

$$\begin{aligned}\widehat{\mathbf{V}} &= Vander\left(\widehat{\xi}_1, \widehat{\xi}_2, \ldots, \widehat{\xi}_{\widehat{T}}\right)\\ &= \begin{bmatrix} \widehat{\xi}_1^0 & \widehat{\xi}_1^1 & \cdots & \widehat{\xi}_1^{(N_{\mathrm{SAMP}}-1)} \\ \widehat{\xi}_2^0 & \widehat{\xi}_2^1 & \cdots & \widehat{\xi}_2^{(N_{\mathrm{SAMP}}-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{\xi}_{\widehat{T}}^0 & \widehat{\xi}_{\widehat{T}}^1 & \cdots & \widehat{\xi}_{\widehat{T}}^{(N_{\mathrm{SAMP}}-1)} \end{bmatrix}.\end{aligned} \tag{3.3.17}$$

Let $\widehat{\mathbf{c}}^{[k]}$ be the k-th column of $\widehat{\mathbf{C}}^T$, $k = 1, 2, 3, ..., 12$. We have $\mathbf{y}_{\mathrm{IF},k} = \widehat{\mathbf{V}}^T \widehat{\mathbf{c}}^{[k]} + \mathbf{v}_{\mathrm{IF}}$. Due to the separability of the parameters of each target and simple computation, we usually employ the Least Square algorithm to resolve the remaining parameters. The closed-form solution to the Least Square optimization problem $\min_{\widehat{\mathbf{c}}^{[k]}} \left\| \mathbf{y}_{\mathrm{IF},k} - \widehat{\mathbf{V}}^T \widehat{\mathbf{c}}^{[k]} \right\|$ is given by $\widehat{\mathbf{c}}^{[k]} = \left(\widehat{\mathbf{V}}\widehat{\mathbf{V}}^T\right)^{-1} \widehat{\mathbf{V}} \mathbf{y}_{\mathrm{IF},k}$. Repeating using the above solution for all $k = 1, 2, 3, ..., 12$, we can get the estimated $\widehat{\mathbf{C}}^T$. Let $\widehat{\mathbf{c}}_i$ and $\mathbf{c}_i$ be the i-th column of $\widehat{\mathbf{C}}$ (or the i-th row of $\widehat{\mathbf{C}}^T$) and $\mathbf{C}$, respectively. The vector $\mathbf{c}_i$ is determined by the remaining parameters $\alpha_i, \omega_{x,i}, \omega_{z,i}$ of target i, which can be estimated by solving another Least Square optimization problem:

$$\min_{\alpha_i, \omega_{x,i}, \omega_{z,i}} f\left(\alpha_i, \omega_{x,i}, \omega_{z,i}\right), \tag{3.3.18}$$

where $f\left(\alpha_i, \omega_{x,i}, \omega_{z,i}\right) = \left\|\widehat{\mathbf{c}}_i - \mathbf{c}_i\right\|^2$. The solution to the optimization problem can be transformed to the formula that we can get the value of estimated parameters directly:

$$\widehat{l} = \arg_l \max \left|\left|\widetilde{B}^{1:8}\left[l\right]\right| + \left|\widetilde{B}^{9:12}\left[l\right]\right|\right|, \tag{3.3.19}$$

where

$$\widetilde{B}^{1:8}\left[l\right] = \sum_{n=0}^{N_{\mathrm{DFT}}-1} \widehat{\mathbf{d}}_i^{1:8}\left[n\right] e^{-j\omega_{x,i}[l]n}, \tag{3.3.20}$$

$$\widetilde{B}^{9:12}\left[l\right] = \sum_{n=0}^{N_{\mathrm{DFT}}-1} \widehat{\mathbf{d}}_i^{9:12}\left[n\right] e^{-j\omega_{x,i}[l]n}, \tag{3.3.21}$$

in which, $\widehat{\mathbf{d}}_i^{1:8} = \begin{bmatrix} \widehat{\mathbf{c}}_i^{1:8H} \\ \mathbf{0} \end{bmatrix}$ and $\widehat{\mathbf{d}}_i^{9:12} = \begin{bmatrix} \widehat{\mathbf{c}}_i^{9:12H} \\ \mathbf{0} \end{bmatrix}$. The detailed transformation steps have been included in the Appendix A.

Let $\omega_{x,i}\left[l\right] = \frac{2\pi l}{N_{\mathrm{DFT}}}$, where N_{DFT} is the length of $\widehat{\mathbf{d}}_i^{1:8}$ and $\widehat{\mathbf{d}}_i^{9:12}$, $0 \leq l \leq N_{\mathrm{DFT}} - 1$. $\widehat{l}$ can be determined by finding the maximum value within $[0, N_{\mathrm{DFT}} - 1]$. Then, the phase shift between azimuth receiver antennas for target i can be obtained as

$$\widehat{\omega}_{x,i} = \frac{2\pi\widehat{l}}{N_{\mathrm{DFT}}}. \tag{3.3.22}$$

Replacing $\widehat{\omega}_{x,i}$ in the definitions of $B^{1:8}$ and $B^{9:12}$ (in Equations (A.0.3), (A.0.4) in Appendix A enables us to obtain the phases of $\widehat{B}^{1:8}$ and $\widehat{B}^{9:12}$:

$$\widehat{\phi}_{B^{1:8}} = \angle\widehat{B}^{1:8}, \widehat{\phi}_{B^{9:12}} = \angle\widehat{B}^{9:12}. \tag{3.3.23}$$

Finally, if we substitute $\widehat{\omega}_{x,i}$, $\widehat{\phi}_{B^{1:8}}$, $\widehat{\phi}_{B^{9:12}}$ into Equation (A.0.5), we get

$$\widehat{\omega}_{z,i} = 2\widehat{\omega}_{x,i} + \widehat{\phi}_{B^{1:8}} - \widehat{\phi}_{B^{9:12}}. \tag{3.3.24}$$

Now, we summarize the details of Step 2 as:

Algorithm 2 Compute $\widehat{\boldsymbol{\omega}}_{\mathbf{x}}$ and $\widehat{\boldsymbol{\omega}}_{\mathbf{z}}$

Input $\widehat{T}$ and $\widehat{\boldsymbol{\xi}}$;
$\widehat{\boldsymbol{\omega}}_{\mathbf{x}} = \emptyset$, $\widehat{\boldsymbol{\omega}}_{\mathbf{z}} = \emptyset$;
Construct Vandermonde matrix $\widehat{\mathbf{V}}$ based on $\widehat{\boldsymbol{\xi}}$ by (3.3.17);
Calculate $\widehat{\mathbf{C}}$ based on $\mathbf{Y}_{\mathrm{IF}}$ and $\widehat{\mathbf{V}}$;
for $i \leftarrow 1$ **to** $\widehat{T}$ **do**
  Take DFT to $\widehat{\mathbf{c}}_i^{1:8H}$ and $\widehat{\mathbf{c}}_i^{9:12H}$, find the index $\widehat{l}$ by (3.3.19);
  Calculate $\widehat{\omega}_{x,i}$ by (3.3.22);
  Calculate $\widehat{\phi}_{B^{1:8}}$ and $\widehat{\phi}_{B^{9:12}}$ by (3.3.23);
  Calculate $\widehat{\omega}_{z,i}$ by (3.3.24);
  $\widehat{\boldsymbol{\omega}}_{\mathbf{x}} = \widehat{\boldsymbol{\omega}}_{\mathbf{x}} \cup \widehat{\omega}_{x,i}$, $\widehat{\boldsymbol{\omega}}_{\mathbf{z}} = \widehat{\boldsymbol{\omega}}_{\mathbf{z}} \cup \widehat{\omega}_{z,i}$;
end for

Step 3: Compute the coordinates of each target

For target i, we can determine its location $\widehat{\mathbf{P}}_i = [\widehat{x}_i, \widehat{y}_i, \widehat{z}_i]$ from $\widehat{R}_i$, $\widehat{\omega}_{x,i}$ and $\widehat{\omega}_{z,i}$ as

$$\widehat{x}_i = \widehat{R}_i \cos\widehat{\phi}_i \sin\widehat{\theta}_i = \widehat{R}_i \frac{\widehat{\omega}_{x,i}}{\pi}, \tag{3.3.25}$$

$$\widehat{z}_i = \widehat{R}_i \sin\widehat{\phi}_i = \widehat{R}_i \frac{\widehat{\omega}_{z,i}}{\pi}, \tag{3.3.26}$$

$$\widehat{y}_i = \sqrt{\widehat{R}_i^2 - \widehat{x}_i^2 - \widehat{z}_i^2}. \tag{3.3.27}$$

Besides that, according to Equation (3.2.1) and (3.2.2), the azimuth angle and elevation angle of target i can be computed as

$$\widehat{\phi}_i = \arcsin\frac{\widehat{\omega}_{z,i}}{\pi}, \widehat{\theta}_i = \arcsin\frac{\widehat{\omega}_{x,i}}{\pi\cos\widehat{\phi}_i}. \tag{3.3.28}$$

Remark 2 *The complexity of the proposed algorithm can be analyzed as follows. Step 1 of Barone's method is $O(RN_{SAMP}^3)$, followed by $O(l^2\,(RN_{SAMP}/2+1))$ in Step 2 where l is the number of points along the lattice edge. Let the number of local* *maxima be* T'. *The complexity of Step 3 & 4 is* $O(T'(RN_{SAMP}/2+1))$ *and that of Step 5 is* $O(\widehat{T})$. *Thus, the overall complexity of Barone's method is* $O(RN_{SAMP}^3 + l^2(RN_{SAMP}/2+1)+T'(RN_{SAMP}/2+1)+\widehat{T})$. *We apply his method for the data from 12 antennas and apply Step 2∼5 for an extra time. We can now deduce that the complexity of Algorithm 1 is* $O(12RN_{SAMP}^3+13l^2(RN_{SAMP}/2+1)+13T'(RN_{SAMP}/2+1)+13\widehat{T})$. *The complexity of Algorithm 2 is dominated by calculating* $\widehat{\mathbf{C}}$ *and* $\widehat{l}$, *which is* $O(\widehat{T}^2 N_{SAMP}+\widehat{T}^3)$ *and* $O(N_{DFT}\log N_{DFT})$, *respectively. Thus, the overall complexity of Algorithm 2 is* $O(\widehat{T}^2 N_{SAMP}+\widehat{T}^3+\widehat{T}N_{DFT}\log N_{DFT})$.

3.4 Simulation Study

In this section, we conduct simulations to evaluate the performance of the proposed algorithm.

3.4.1 Simulation Settings

We have implemented the proposed algorithm in MATLAB on a PC equipped with an Intel Core 8700 CPU and 16GB RAM. The simulation settings for the radar are as follows: (1) The start frequency and end frequency are 77GHz and 81GHz, respectively; (2) The three transmit antennas take turns in sending chirp signals of 58us length in the order of TX1, TX3 and TX2; (3) The IF signals at the 12 virtual receiver antennas are sampled between 7us and 57us after a chirp has been sent. The total sampling number is 225.

For the targets and the environment, we vary the Signal-to-Noise Ratio (SNR) from 0dB to 30dB in 5dB increments with complex addictive Gaussian white noise.

Under each SNR situation, 10000 Monte-Carlo experiments are repeated with randomly generated T targets with ranges between 0.05m and 9m, azimuth angles between -28° and 28°, and elevation angles between -14° and 14°.

Two metrics are used to evaluate the performance of the different algorithms that we consider: the average number of detected targets under different SNRs, and target location estimation errors. To calculate the location errors, we need to first associate each estimated target with an actual target. Let $\mathbf{P}_i = [x_i, y_i, z_i]$, $i = 1,\ 2,\ 3, \dots,\ T$, be the actual locations of T target, and $\widehat{\mathbf{P}}_j = [\widehat{x}_j, \widehat{y}_j, \widehat{z}_j]$, $i = 1,\ 2,\ 3, \dots,\ \widehat{T}$, be the estimated locations, where $\widehat{T}$ is the estimated number of targets. We perform target association using the Kuhn-Munkres (KM) algorithm by solving the following bipartite matching problem:

$$\min_{i,j} \frac{1}{\widetilde{T}} \sum_{i=1}^{T} \sum_{j=1}^{\widehat{T}} S_{ij} x_{ij}$$

$$\text{s.t.} \begin{cases} S_{ij} = \|\mathbf{P}_i - \widehat{\mathbf{P}}_j\| \\ x_{ij} \in \{0,1\} \\ \sum_{i=1}^{T} x_{ij} \leq 1 \\ \sum_{j=1}^{\widehat{T}} x_{ij} \leq 1 \\ \sum_{i=1}^{T} \sum_{j=1}^{\widehat{T}} x_{ij} = \widetilde{T} \end{cases}, \tag{3.4.1}$$

where $\widetilde{T} = \min(T, \widehat{T})$. We denote the resulting pairs by $\langle \mathbf{P}^t, \widehat{\mathbf{P}}^t \rangle$, $t = 1, 2, 3, \dots, \widetilde{T}$. Finally, the location estimation error is then computed as,

$$D = \frac{1}{\widetilde{T}} \sum_{t=1}^{\widetilde{T}} \frac{\|\mathbf{P}^t - \widehat{\mathbf{P}}^t\|}{\|\mathbf{P}^t\|}. \tag{3.4.2}$$

With $\widetilde{T}$ pairs of associated targets, the MAE of all polar coordinate parameters (i.e., R, θ, ϕ) are given by,

$$\mathrm{MAE}_R = \frac{1}{\widetilde{T}} \sum_{t=1}^{\widetilde{T}} \left| R_t - \widehat{R}_t \right|, \tag{3.4.3}$$

$$\mathrm{MAE}_\theta = \frac{1}{\widetilde{T}} \sum_{t=1}^{\widetilde{T}} \left| \theta_t - \widehat{\theta}_t \right|, \tag{3.4.4}$$

$$\mathrm{MAE}_\phi = \frac{1}{\widetilde{T}} \sum_{t=1}^{\widetilde{T}} \left| \phi_t - \widehat{\phi}_t \right|. \tag{3.4.5}$$

3.4.2 Baseline Algorithms

For comparison purposes, we have also implemented three baseline algorithms: 2D-FFT, 2D-MUSIC and 3D-MUSIC. In the 2D-FFT algorithm, we apply range-FFT to the sampled radar data matrix and find peaks in the spectrum. CFAR detection and group peaking are used to estimate the number of targets and their ranges. Next, we apply angle-FFT to the data at each detected peak from all antennas to obtain the angle information. In 2D-MUSIC, target detections and range estimation are the same as in 2D-FFT, but angle-FFT is replaced by the MUSIC algorithm to estimate the AoAs of potential targets.

The 3D-MUSIC algorithm is based on the approach in [41]. It pre-calculates range steering vectors, azimuth & elevation angle steering vectors. The number of targeted is estimated by MDL, a mode selection approach based on information theory criteria. To accelerate the computation time, after calculating pseudo-spectrum, we restrict peak searches to neighborhoods of ground truth target locations. Note that doing so errs on the optimistic side for 3D-MUSIC. Exhaustive search in 3D space will incur

Table 3.1: Key Parameters of the Three Algorithms

2D-FFT algorithm	
Range-FFT bins	256
Angle-FFT bins	1024
CFAR averaging mode	CFAR-Cell Averaging Smallest Of (CASO)
CFAR noise averaging window length	8
CFAR guard length	4
CFAR cyclic mode or wrapped	Not used
CFAR threshold scale	15dB
MUSIC algorithm	
Angle bins	1024
Proposed algorithm	
N_{SAMP}	80
R	50
Threshold	50% of R
Range of the Lattice L	-1 to 1 and -j to j
Side length of each grid of L	0.01
Variance of added pseudo noise	${\sigma'}^2 = 0.64\sigma^2$

Table 3.2: Time Complexity of the Three Algorithms

2D-FFT	$O(12N_{\text{SAMP}}\log N_{\text{SAMP}} + L_a N_{\text{SAMP}} + N_{\text{SAMP}} + 2\widehat{T}N_{\text{DFT}}\log N_{\text{DFT}})$
2D-MUSIC	$O(\widehat{T}(N^3 + N_A^2(N(N-\widehat{T})2 + N^2) + N_{\text{DFT}}^2))$
3D-MUSIC	$O((N_{\text{SAMP}}N)^3 + N_{\text{SAMP}}(\sqrt{N_{\text{DFT}}})^2(N_{\text{SAMP}}N(N_{\text{SAMP}}N - \widehat{T})^2 + (N_{\text{SAMP}}N)^2) + N_{\text{SAMP}}(\sqrt{N_{\text{DFT}}})^2 + \widehat{T}((\sqrt{N_{\text{DFT}}})^2(N_{\text{SAMP}}N(N_{\text{SAMP}}N - \widehat{T})^2 + (N_{\text{SAMP}}N)^2) + (\sqrt{N_{\text{DFT}}})^2))$

excessive computation overhead and likely worse performance.

The key parameters of the three algorithms are summarized in Table 3.1 and the time complexity analysis has been put in Table 3.2. Note that L_a is the CFAR noise averaging window length and N_A is the number of angle bins. In our case, N_A equals to N_{DFT}. All the baseline algorithms except 3D-MUSIC have been evaluated in 10000 Monte-Carlo experiments for each SNR situation. We only run 100 experiments per SNR scenario for 3D-MUSIC due to its extremely high time complexity.

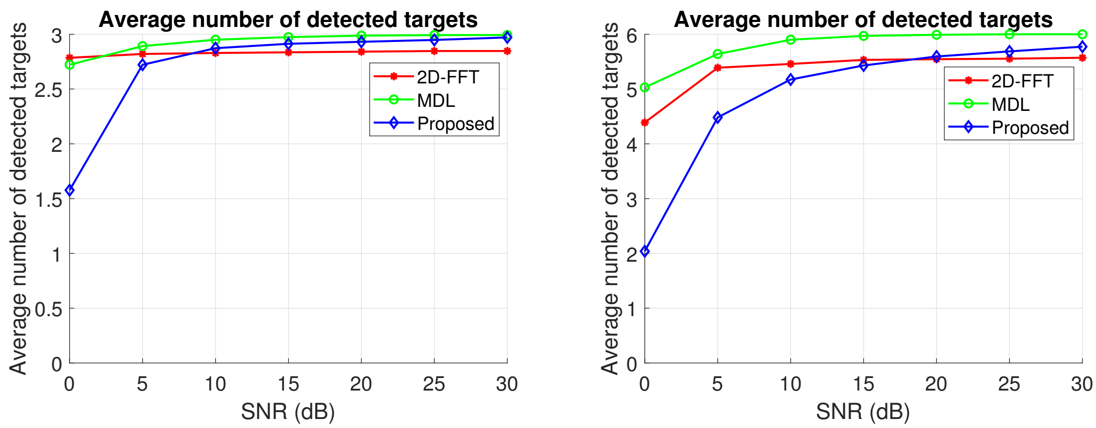
3.4.3 Results and Analysis

Average number of detected targets

Figure 3.4 shows that with 3 or 6 targets, the average number of detected targets under different SNRs. It can be observed the proposed method under-estimate the number of targets in the low SNR regime but outperform the 2D-FFT method when the SNR is above 8dB for 3 targets and 18dB for 6 targets. In addition, MDL, a method based on information theoretic criteria, has the best performance in nearly all SNR regimes.

Recall that from Padé approximant theory, the candidate Padé poles are among the local maxima of complex measure $\widehat{S}_{N_{\mathrm{SAMP}},R}(z, \widetilde{\sigma}^2)$. However, in the low SNR situation, the energy of the noise and the energy of the signal are comparable. Thus, the local maxima due to signal and those from noise are of similar values. From [7], the higher local maxima, the higher the probability for more Padé poles in its neighboring area. However, due to noise, the number of large local maxima increases on the lattice. As a result, the number of Padé poles around the local maxima corresponding to a real target may drop below the predefined percentage (e.g., >50% of R). In contrast, in the higher SNR regime, large local maxima typically corresponds to the signals reflected from real targets. They can attract sufficient number of Padé poles to exceed the threshold and thus can be detected.

The 2D-FFT method performs well when targets are well separated. However, its range resolution is determined by the sweeping bandwidth B (e.g., 4GHz) and its best angle resolution is given by $2/N$ at the boresight direction. If two targets fall in the same range and angle bins, they cannot be separated in the 2D-FFT method. The choice of the CFAR threshold also affects the number of detected targets in low



(a) 3 Targets (b) 6 Targets

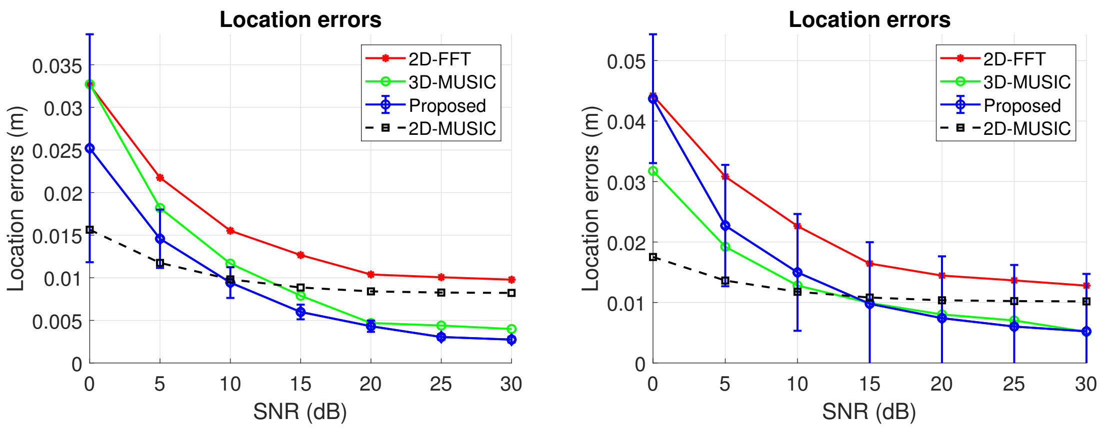
Figure 3.4: Average number of detected targets under different SNRs

SNRs by trading-off false positive rates with detection probabilities.

Location errors

Figure 3.5 shows the distance between the estimated and true locations for 3 and 6 targets under different SNRs. As expected, as the SNR increases, the average location error decreases for all algorithms. In comparison, the proposed algorithm always achieves lower location errors than 2D-FFT and has better performance than 2D-MUSIC in high SNR regime. We further plot the MAE of range, azimuth and elevation angle estimation errors for 3 and 6 targets in Figure 3.6. It can be observed from Figure 3.6a that the proposed algorithm has lower errors in estimating range parameters. However, in estimating azimuth and elevation angles, 2D-FFT and 2D-MUSIC algorithms have higher accuracy in low SNRs, but the proposed algorithm outperforms both of them under high SNRs.

While the AoA and localization accuracy of 2D-MUSIC consistently outperforms 2D-FFT in all scenarios, the AoA estimations of 3D-MUSIC are worse than those



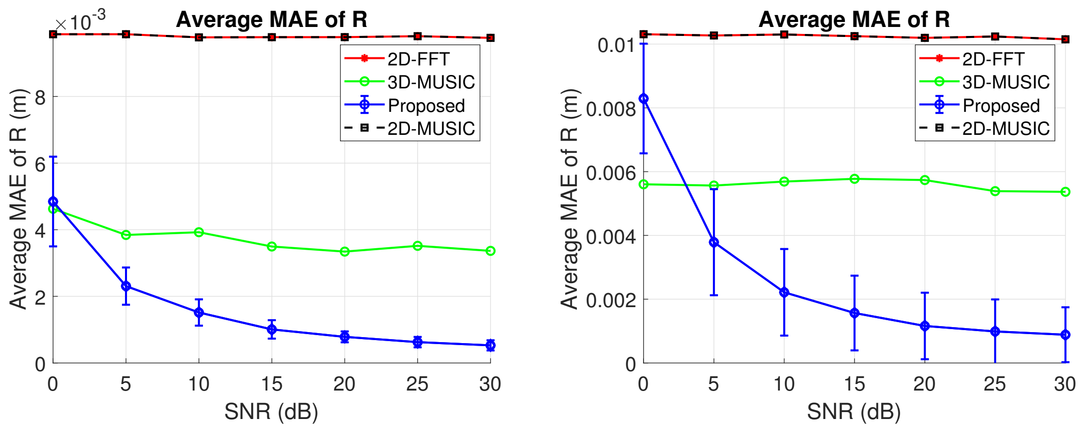
(a) 3 Targets (b) 6 Targets

Figure 3.5: Average location errors under different SNRs

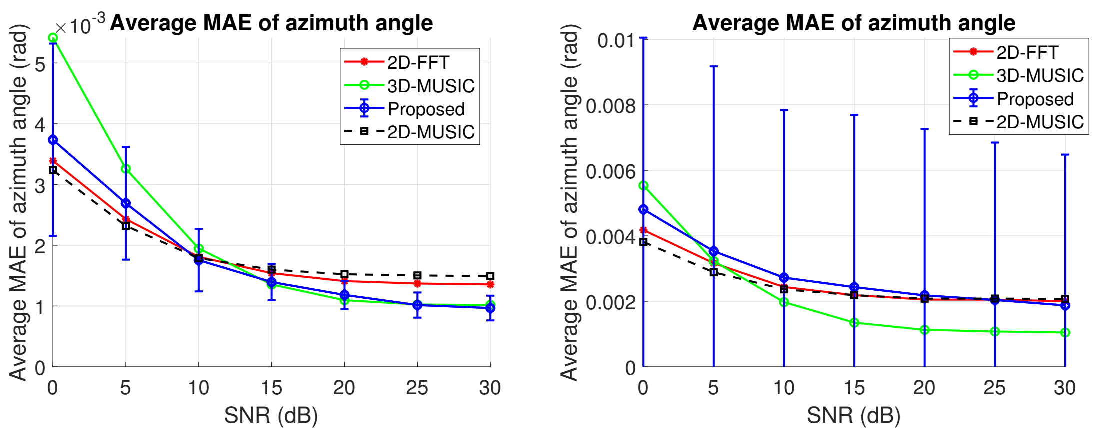
of 2D-FFT in the low SNR regime for both 3-target and 6-target cases. This can be attributed to two factors. First, the 3D-MUSIC algorithm in [41] only utilizes stacked-up auto-correlation matrices for each antenna element over time (instead of exploring the correlation in both space and time. Second, to approximate auto-correlation matrices in time, a large number of frames are needed. Otherwise, the resulting approximation is biased. This is especially true in the low SNR regime. It should be noted that due to the peak search approach adopted, the accuracy of 3D-MUSIC reported errs on the optimistic side. Nevertheless, the proposed method has comparable or lower localization errors as 3D-MUSIC in mid- to high- SNR regimes at significantly lower computation costs. This demonstrates the advantages of the proposed method.

Impact of the number of data samples

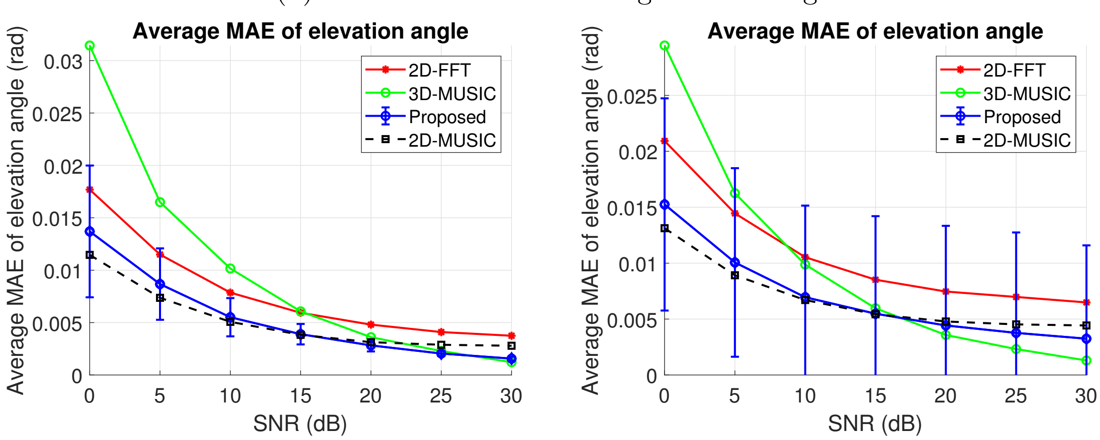
The purpose of this set of experiments is to understand how the number of samples affects the performance of the proposed method. In the experiments, the SNRs are



(a) The MAE of the range of the targets R

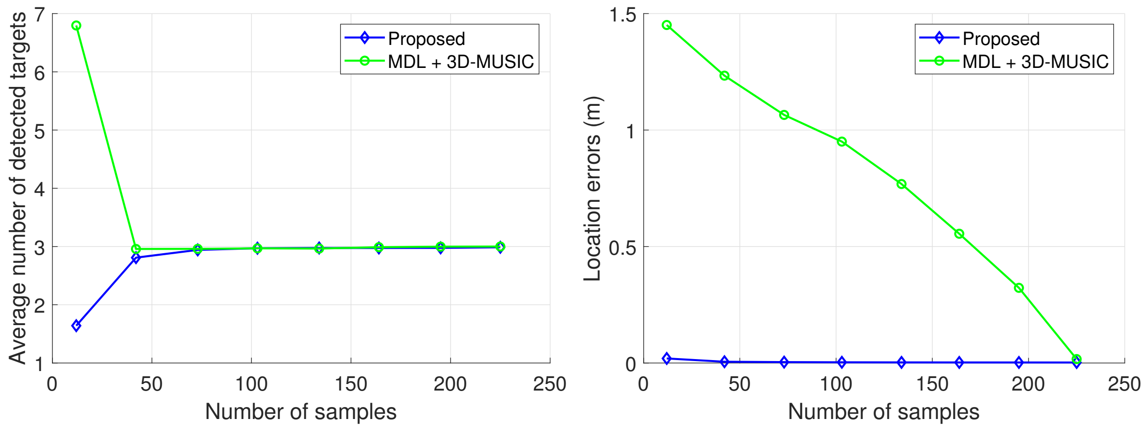


(b) The MAE of azimuth angle of the targets θ



(c) The MAE of elevation angle of the targets ϕ

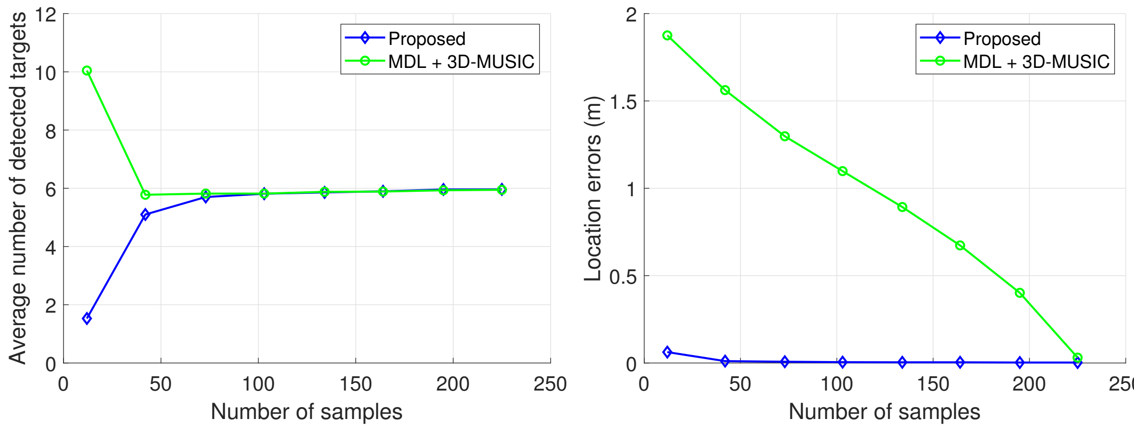
Figure 3.6: The MAE of the polar coordinate parameters of the targets in 3- (left column) & 6- (right column) target situation



(a) Average number of detected targets (b) Location errors

Figure 3.7: Effects of the number of data samples. (3 Targets)

set to 30dB and the number of sampled data is chosen from 12, 42, 73, 103, 134, 164, 195 and 225. A total of 10000 experiments are performed for each value. Figs. 3.7 and 3.8 show the results for 3-target and 6-target cases, respectively. It can be observed from the figures that as the number of data samples becomes larger, the number of detected targets approaches to the actual value while the estimated target locations get closer to the ground truth locations. In fact, about 100 data samples are sufficient to detect all targets accurately with the proposed algorithm, and using less data samples is advantageous in detecting and locating mobile targets. In the previous experiments using the proposed algorithm, we only utilize the first 80 sampled data from each virtual receiver antenna to achieve the results and any 80 contiguous data samples can be used to get the similar results. In contrast, the 3D-MUSIC algorithm cannot attain comparable results with such a small number of data samples.



(a) Average number of detected targets (b) Location errors

Figure 3.8: Effects of the number of data samples. (6 Targets)

3.5 Testbed Evaluation

We have implemented the proposed and baseline algorithms using a COTS mmWave radar, i.e., Taxes Instruments IWR6843ISK. The radar has 3 transmit antennas and 4 receiver antennas, where the antenna arrangement is the same as that in Fig. 3.1a. Three corner reflectors, each of which is constructed by three metal plates perpendicular to one another, are used to represent three point targets. We use Optitrack, an optical motion capture system, to determine the ground truth locations of the radar and targets. The detailed parameters are summarized in Table 3.3 and the experiment environment is shown in Fig. 3.9.

Table 3.4 and 3.5 summarizes the average range and directional errors for the 1-target case and the 3-target case, respectively. From the experiment results, we see that the proposed algorithm has lower average errors than baseline algorithms in most of the criteria. Nonetheless, we notice that the measured errors are far larger than those from simulations. The discrepancy between simulation and experimental results can be attributed to several sources. First and foremost, interference from

Table 3.3: Testbed Setup

Radar	
Frequency	60∼64GHz
Pulse width	50us
Number of ADC samples	225
Corner Reflector	
Edge length	8 inch
Optitrack	
Measurement error	0.2mm

Figure 3.9: Experiment Environment

Table 3.4: 1-Target Experiment Results

Metrics (averaged)	Proposed algorithm	2D-FFT	2D-MUSIC	3D-MUSIC
# of detected targets	**1**	**1**	**1**	**1**
Location errors (m)	**0.4501**	0.6198	0.4663	0.6088
MAE_R (m)	0.0671	0.0714	0.0714	**0.0669**
MAE_θ (rad)	**0.0212**	0.0454	0.0248	0.0442
MAE_ϕ (rad)	**0.4405**	0.6113	0.4554	0.6100
Time cost (s)	7.7902	**0.0024**	17.6388	21.0659

Table 3.5: 3-Target Experiment Results

Metrics (averaged)	Proposed algorithm	2D-FFT	2D-MUSIC	3D-MUSIC
# of detected targets	**3**	**3**	**3**	**3**
Location errors (m)	**0.4140**	0.5619	0.4736	0.5638
MAE_R (m)	**0.2811**	0.5191	0.4614	0.5192
MAE_θ (rad)	**0.1861**	0.2676	0.2129	0.2692
MAE_ϕ (rad)	0.2322	0.2239	**0.1994**	0.2239
Time cost (s)	7.8355	**0.0038**	31.0145	34.0348

surrounding objects such as walls, pillars and furniture, can significantly impact the performance. Second, the edge length of corner reflectors is 8 inches, making them multi-scatter point targets. Third, corner reflectors can interfere with each other. For instance, the peak of one corner reflector may be obscured by the sidelobes of another in the spectrum, making the algorithms capture wrong targets elsewhere with relatively high peaks. This may explain the performance gap between the single target and 3-target cases. Other reasons may include the limitations of the radar board, e.g., the accuracy of its mixer, ADC and the cosine pattern of its antennas. In terms of the MUSIC algorithms, we only utilize one received chirp in the analysis, which may not be sufficient to estimate covariance matrices accurately in such an indoor environment with high interference. In addition, the MUSIC algorithms need more computation time than the proposed algorithm.

3.6 Conclusion

In this chapter, we proposed a new algorithm to estimate the locations of multi-targets by using the radar with irregular antenna placement. From the simulation results, it can be concluded that the proposed algorithm can detect nearly all the targets especially in the high SNR regime, and the probability of losing targets is very low compared with 2D-FFT method. Besides, among those detected targets, the location error is significantly lower than that of 2D-FFT and MUSIC algorithm. We also found that the resolution and performance of the proposed algorithm does not heavily depends on the number of sampled data and hardware design. Compared with 2D-FFT algorithm which needs hundreds of data samples to cover a large indoor area, the proposed algorithm only needs less than half of them to get an even better result.

Chapter 4

A Robotic ROSAR System for Indoor Imaging

4.1 Introduction

In this chapter, we present the design of a real-world robotic ROSAR system. We begin by analyzing the array pattern of ROSAR. The analysis informs the choice of key design parameters, including the rotation speed, diameter of the rotation platform, radar frame rate, and maximum rover speed. We then introduce the details of the ROSAR system, followed by a measurement study of possible error sources in controlled radar trajectories.

4.2 Related Work

The use of ROSAR was first proposed by Klausing [36]. The author attached a radar to a helicopter blade tip. The radar boresight direction pointed outward with a downward angle. Through the analysis of the data collected during the fast blade rotation, a basic idea of the ground situation can be characterized. Later, ROSAR theory was applied to precise wireless positioning [62]. By accurately locating a backscatter transponder, the position of a vehicle with the transponder in a parking lot could be determined.

ROSAR was first applied to indoor mapping in [3]. The authors developed a system consisting of an omnidirectional radar mounted on a rotating platform operating in K band. By applying BPA to the data collected by the radar, the system can produce clear images of objects with different backscatter coefficients in a 360° view [4]. Windowing functions are also used to enhance the image quality. In a follow-up work by the same author, the ROSAR system is mounted on to a rover that moved in a straight line to image a larger area [2]. In this case, the radar path is a spiral line, but BPA is still applicable. To correct the measurement errors of the radar positions, modifications were made to the original BPA.

4.3 Array Pattern Analysis

To determine the key design parameters of ROSAR, we need to characterize its array pattern. In this section, we compare the array patterns of two types of arrays: Uniform Linear Array and Uniform Circular Array in both near and far field cases. UCA is a special case of ROSAR when there is only rotation motion and the rotation speed is constant. In practice, because of the unforeseen time-variant disturbances in the platform, the phase centers are not always evenly spaced. Moreover, compared to UCA, the ROSAR system can be mounted on a moving vehicle, which makes the array manifold a spiral line. Nevertheless, under slow translational motions and constant rotation speed, one can approximate the array pattern of a ROSAR by that of a UCA.

Consider N antennas arranged in a line or a circle with identical spatial intervals, i.e., ULA and UCA. The antenna radiation pattern is cosine-shaped and the field-of-view of each antenna is between 0° to 180°. The region of the electromagnetic field around the antennas can be divided into near field and far field, where the wavefront of the signals reflected from targets in far field is assumed to be planar and perpendicular to the direction of propagation. The limit between the near and far field is usually determined by Fraunhofer distance [58], i.e., $d_F = 2L^2/\lambda$, where L is the array physical length. Let d, r, n, θ denote the distance between the neighbouring antennas in the ULA, its radius, the antenna index and AoA of the signal, respectively. The array patterns are given by

$$AP(\theta) = \mathbf{w}^H \mathbf{a}(\theta), \tag{4.3.1}$$

where θ is within 0° to 180°; $\mathbf{a}(\theta)$ is the steering vector and $\mathbf{w}$ is a complex matched filter, which has the same expression of $\mathbf{a}(\theta)$. In far field scenarios, $\mathbf{w}$ steers the beam to the 90° direction (boresight direction). In near field scenarios, $\mathbf{w}$ is chosen to steer the resulting beam to a pixel in polar coordinate $(R, 90^\circ)$. Next, we analyze four array patterns of ULA and UCA in near and far fields.

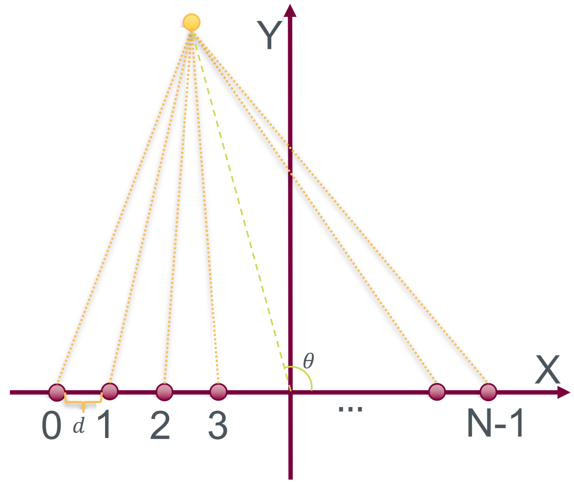


Figure 4.1: Radar Model of the ULA (Near Field)

(1) ULA (Near Field)

$$AP(\theta) = \mathbf{w}^H \begin{bmatrix} \cos(\theta_0)\, e^{j2k\sqrt{(R\cos\theta - x_0)^2 + (R\sin\theta)^2}} \\ \cos(\theta_1)\, e^{j2k\sqrt{(R\cos\theta - x_1)^2 + (R\sin\theta)^2}} \\ \vdots \\ \cos(\theta_{N-1})\, e^{j2k\sqrt{(R\cos\theta - x_{N-1})^2 + (R\sin\theta)^2}} \end{bmatrix}, \tag{4.3.2}$$

where $k = 2\pi/\lambda$ is the wavenumber; x_n is the x-coordinate of the n-th antenna; (R, θ) is the target coordinate and θ_n is given by

$$\theta_n = \arctan \frac{R\sin\theta}{R\cos\theta - x_n} - 90. \tag{4.3.3}$$

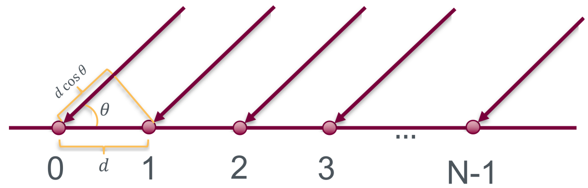


Figure 4.2: Radar Model of the ULA (Far Field)

(2) ULA (Far Field)

$$AP(\theta) = \mathbf{w}^H \begin{bmatrix} \cos(\theta - 90)e^{j2k\cdot 0\cdot d\cos\theta} \\ \cos(\theta - 90)e^{j2k\cdot 1\cdot d\cos\theta} \\ \vdots \\ \cos(\theta - 90)e^{j2k\cdot(N-1)\cdot d\cos\theta} \end{bmatrix} \tag{4.3.4}$$

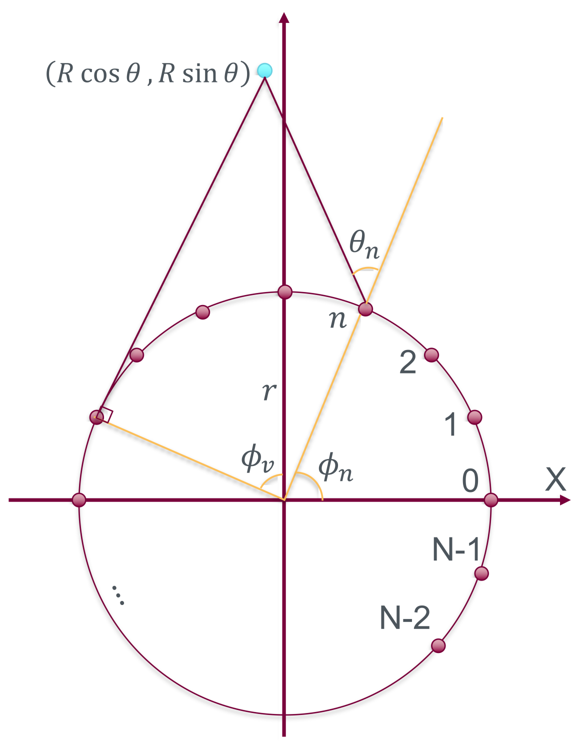


Figure 4.3: Radar Model of the UCA (Near Field)

(3) UCA (Near Field)

$$AP(\theta) = \mathbf{w}^H \begin{bmatrix} \cos\left(\theta_{N_{\min}}\right) e^{j2k\sqrt{R^2+r^2+2Rr\cos\left(\theta-\phi_{N_{\min}}\right)}} \\ \cos\left(\theta_{N_{\min}+1}\right) e^{j2k\sqrt{R^2+r^2+2Rr\cos\left(\theta-\phi_{N_{\min}+1}\right)}} \\ \vdots \\ \cos\left(\theta_{N_{\max}}\right) e^{j2k\sqrt{R^2+r^2+2Rr\cos(\theta-\phi_{N_{\max}})}} \end{bmatrix}, \qquad (4.3.5)$$

where θ_n is given by

$$\theta_n = \arctan \frac{R \cdot \sin\left(|\theta - \phi_n|\right)}{R \cdot \cos\left(|\theta - \phi_n|\right) - r},$$

$n = N_{\min}, N_{\min}+1, \ldots, N_{\max}$, $N_{\min} = \left\lceil \frac{(\theta-\phi_v)}{\phi_\Delta} \right\rceil$, $N_{\max} = \left\lfloor \frac{(\theta+\phi_v)}{\phi_\Delta} \right\rfloor$, $\phi_v = \arccos r/R_t$, $\phi_\Delta = 2\pi/N$.

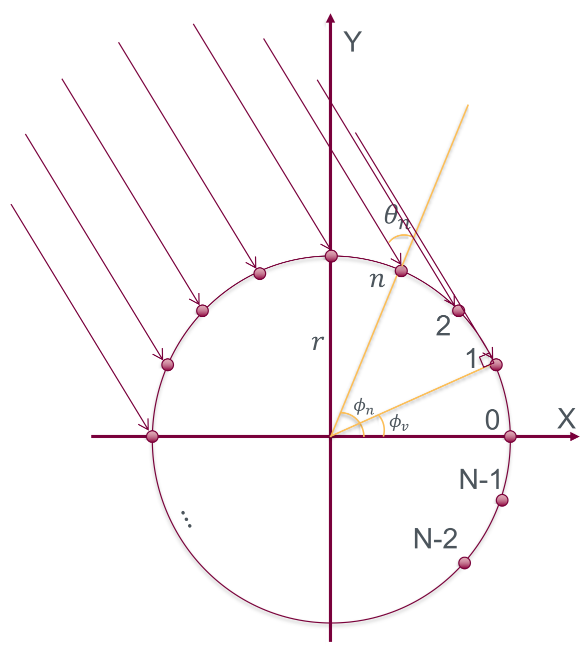


Figure 4.4: Radar Model of the UCA (Far Field)

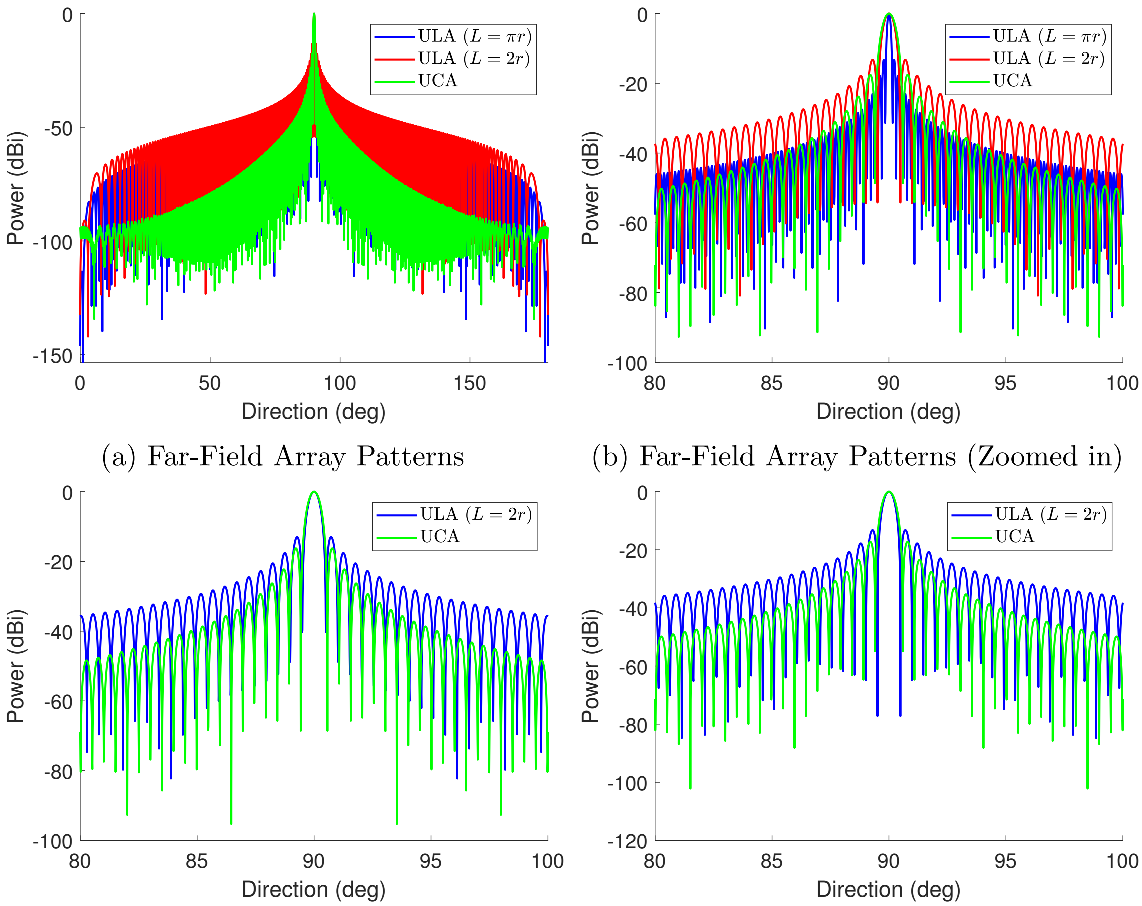
(4) UCA (Far Field)

$$AP(\theta) = \mathbf{w}^H \begin{bmatrix} \cos\left(\theta - \theta_{N_{\min}}\right) e^{j2kr\cos\left(\theta - \theta_{N_{\min}}\right)} \\ \cos\left(\theta - \theta_{N_{\min}+1}\right) e^{j2kr\cos\left(\theta - \theta_{N_{\min}+1}\right)} \\ \vdots \\ \cos\left(\theta - \theta_{N_{\max}}\right) e^{j2kr\cos\left(\theta - \theta_{N_{\max}}\right)} \end{bmatrix} \tag{4.3.6}$$

Table 4.1: Parameter Settings

r	0.145 m
λ	0.005 m
Antenna Pattern	Omni-directional
Radar Signal Type	Chirp Signal
Bandwidth	4 GHz
Chirp duration	50 us
N	800

Fig. 4.5 shows the array patterns of the four cases with the parameter settings in Table 4.1. Fig. 4.5a shows the far-field array patterns, where the blue line and red line represent the array patterns of the ULA with L being πr and $2r$, respectively. Green line represents the UCA's with radius r. Fig. 4.5b zooms in on the direction between $80°$ to $100°$. From the figures, we can see that the ULA has narrower main beamwidth than that of a UCA, which means it has better angle resolution. However, the UCA has lower sidelobe levels. Figs. 4.5c and 4.5d show the near-field array patterns, where $L = 2r$ and the target location is at $(0.5m, 90°)$ and $(2m, 90°)$, respectively. Based on the parameter settings, both targets are within the Fraunhofer distance, i.e., $d_F = 75.69m$. The array patterns in Figs. 4.5c and 4.5d are nearly comparable to those in far-field case.



(a) Far-Field Array Patterns

(b) Far-Field Array Patterns (Zoomed in)

(c) Near-Field Array Patterns (Zoomed in) Target Location: $(0.5m, 90°)$

(d) Near-Field Array Patterns (Zoomed in) Target Location: $(2m, 90°)$

Figure 4.5: Array Patterns

In this thesis, we focus on ROSAR, which has a similar array pattern to a near-field UCA. In Chapter 6, we investigate the case where a ROSAR system is mounted on a rover that moves linearly. This results in a spiral line for the radar movement path. Although ROSAR has low sidelobe levels, the spacing of phase centers can have an significant impact on the sharpness of the resulting images. In [72], it is demonstrated that when the spatial Nyquist–Shannon sampling criterion is violated, grating lobes emerge in the beam pattern. These grating lobes cause the same target to appear at multiple locations. To mitigate this effect, the maximum distance between adjacent phase centers in a ULA should be $\lambda/4$. Since ROSAR lacks a closed-form array pattern, we approximate its behavior by using the requirement that the aperture size of the ULA matches the diameter of the ROSAR, given their similar mainlobe width. For ROSAR, this implies constraints on the rotation speed ω (rad/s), rotation platform radius r (m) and radar chirp rate f Hz. Specifically,

$$\frac{\omega r}{f} \leq \frac{\lambda}{4}. \tag{4.3.7}$$

Similarly, the angular resolution can be approximated as $\theta_\Delta = \frac{\lambda}{2r}$ at its boresight direction (i.e., 90°).

Considering the case that the ROSAR system is mounted on a moving rover. Let the maximum and actual rotation speed be $\omega_{\max}$ and ω_0 (rad/s), respectively. The maximum rover speed without violating the spatial Nyquist–Shannon sampling criterion is given by

$$v_{max} = (\omega_{\max} - \omega_0)\, r. \tag{4.3.8}$$

For example, based on the parameter settings in Table 4.1, $\omega_{\max}$ is 6.8918 rad/s. Assuming $\omega_0 = 2\pi$ rad/s, the maximum rover speed is 0.0882 m/s.

4.4 ROSAR System Design

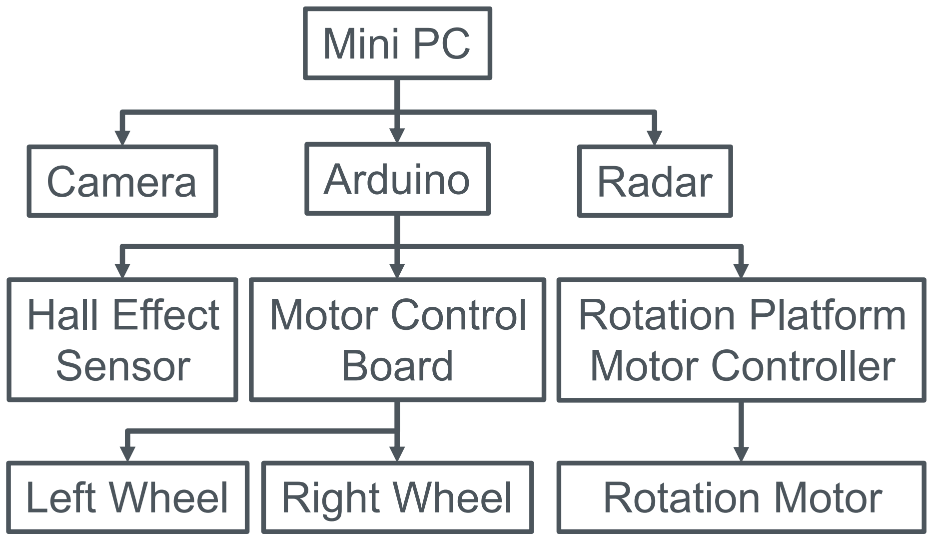


Figure 4.6: System Architecture

Figure 4.6 illustrates the overall system architecture. We use an Arlo robot to carry all the components, and the entire system is controlled by a mini-PC running Linux and Robot Operating System (ROS). The Mini-PC is directly connected to a webcam, an Arduino Mega 2560 board, and a Texas Instruments (TI) IWR6843ISK+DCA1000EVA radar. The antenna pattern of IWR6843ISK can be approximated as a cosine shape based on its real-world measurement[1], which matches the simulated antenna elements in Sec. 4.3. The FoV of each antenna along the azimuth direction is wider than that along the elevation direction. The Arduino board controls all the embedded devices, including a hall effect sensor (US5881) that marks the zero-degree direction of the rotation plate, the Arlo motor control board (DHB-10) that controls the rotation of the two wheels, and the rotation platform motor controller (DRV8771) that controls the speed of the rotation motor. Figure 4.7 shows a prototype of the built ROSAR

[1] https://www.ti.com/lit/ug/swru546e/swru546e.pdf

system, and the detailed circuit is attached in Appendix C. As shown in these figures, we placed the rotation platform on top of the robot. The rotation platform contains the radar, a 3D-printed plate, counterweight, and a hidden magnetic chip. The rotation motor connects the 3D-printed plate directly and drives the whole platform to rotate. The radar antenna's rotation radius is 0.145m.

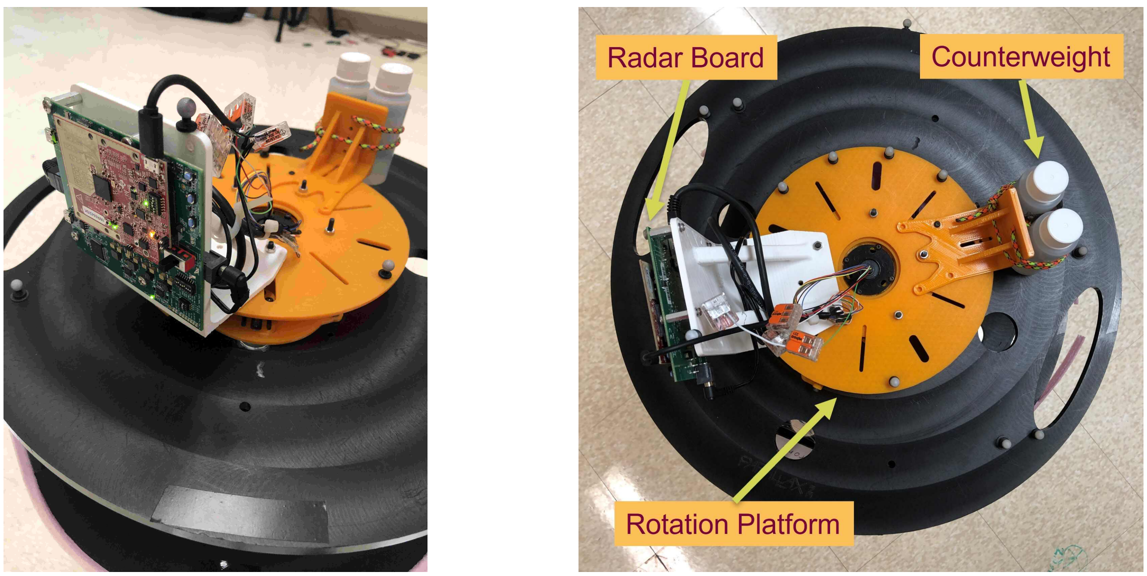


Figure 4.7: ROSAR system

The entire system is controlled by ROS, including the radar signal transmission and reception, the rotation speed, the recording of hall effect triggers, and the rover moving speed. Specifically, the radar data is collected from the antenna board, and the angle readings of the rotation platform are obtained from the rotation motor encoder. Combined with hall effect triggers and rotation radius, the real-time directions and locations of the radar can be derived. The wheel encoder can provide the tick readings of both wheels. When the system starts running, the rotation speed is adjusted by a PID controller running on the Arduino board based on angle readings. The magnetic chip attached to the plate passes the hall effect sensor in each round, generating a trigger to signal the zero-degree direction. Encoders on both wheels are used to control the robot's real-time moving speed by using Arlo APIs. All the data is also streamed to the mini-PC for further processing, such as ROSAR imaging.

4.5 Movement Profiling

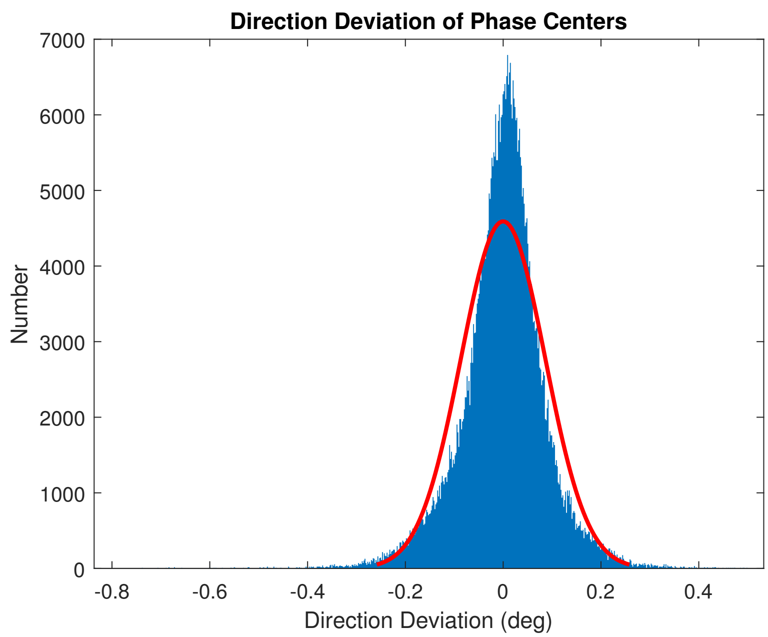


Figure 4.8: Direction deviation of phase centers

Ideally, the locations of the radar as well as its antennas should be precisely known and controlled. But unfortunately, unforeseen disturbances and resistances can lead to fluctuations in the radar's rotation speed that cannot be corrected by the PID controller. Consequently, the radar may not consistently locate at its target position at each timestamp. In other words, there is a direction deviation of each virtual phase center. To measure rotation stability, we mount several optical markers (grey balls in Fig. 4.7) on top of the 3D-printed plate. Markers can be tracked by OptiTrack, an optical motion capture system to obtain ground truth locations (position accuracy: 0.2 mm). To measure the direction deviation of phase centers, we let the radar rotate for many rounds at 2π rad/s and compare the directions derived from sensors on the ROSAR system in Sec. 4.4 and from the OptiTrack system. Fig. 4.8 shows the mean value of 3.9×10^{-4} degree. If the data is fit by a Gaussian distribution, the standard deviation is $\sigma = 0.086$. Thus, we can approximate the direction of the n-th phase center as $\widehat{\phi}_n \sim \mathcal{N}(2\pi n/N, 0.086)$.

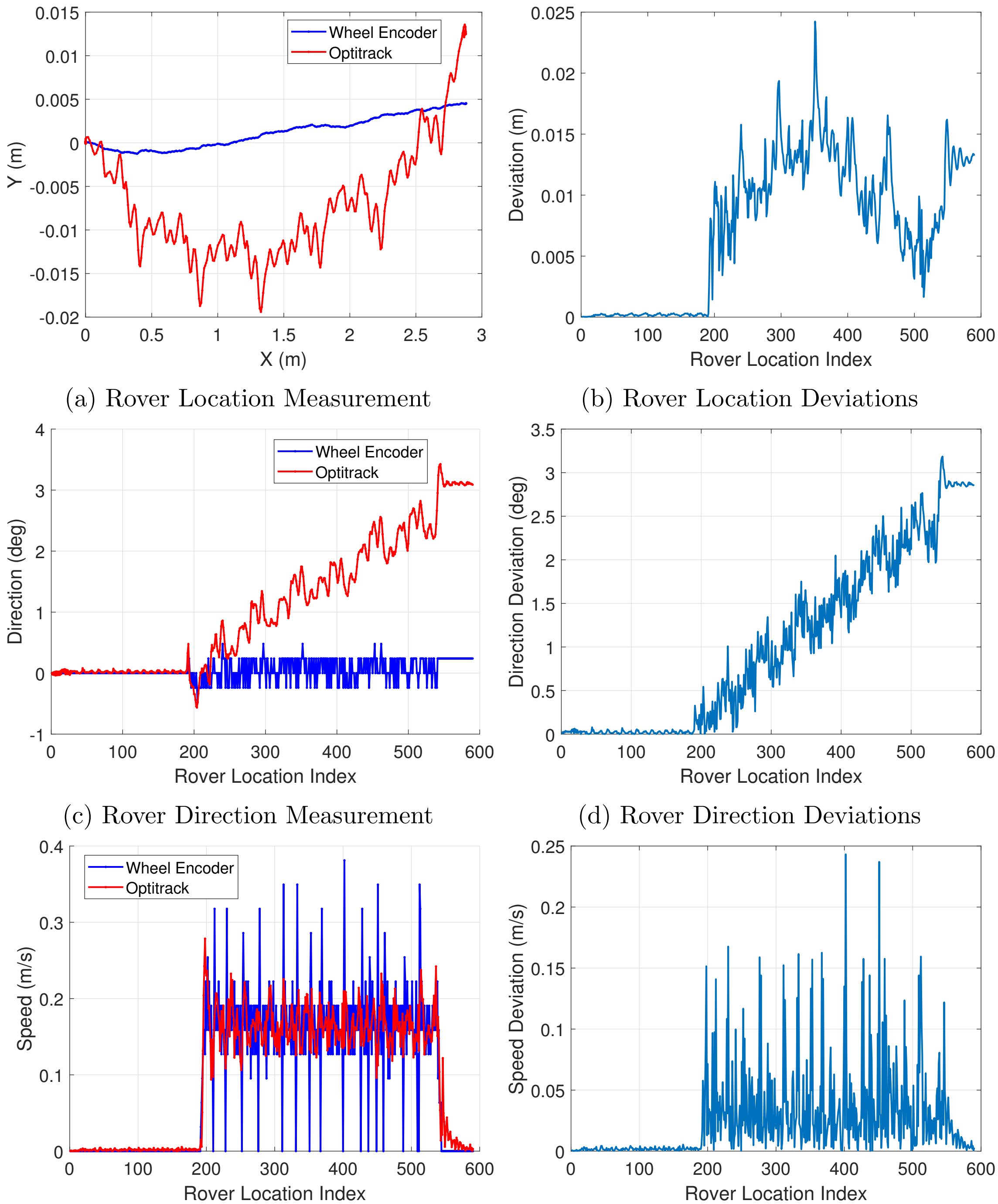
Next, we quantify the accuracy of rover movements as estimated from its wheel encoders. To determine rover's location, direction and speed at each timestamp, we input the ticks from two encoders into a MATLAB module called "wheelEncoderOdometryDifferentialDrive"[1]. The resulting poses are then processed by another module called "poseGraph"[2], which provides rover's locations and directions. Additionally, we continuously rely on Optitrack to obtain the rover's ground truth locations and directions. The rover's real-time speed can be estimated based on the displacement between adjacent location points and a time interval of 0.05 seconds.

The rover movement is subject to more disturbances than the radar rotation, due to the ground surface roughness and levelness. In practice, these factors could vary a lot within a short distance. Moreover, by using the two wheel encoders, one can only obtain the rotation angles of the wheels while wheel slippage cannot be obtained from the encoders. Therefore, it is difficult to find any patterns for the deviations of the rover. We present two rover movement examples in Figs. 4.9 and 4.10. Fig. 4.9 shows that the rover is set to move in a straight line for about 3m. Its real-time location, direction and speed are measured by both the wheel encoders and the OptiTrack. The

[1] https://www.mathworks.com/help/nav/ref/wheelencoderodometrydifferentialdrive-system-object.html

[2] https://www.mathworks.com/help/nav/ref/posegraph.html

location measurement result is shown in Fig. 4.9a. Fig. 4.9b shows the rover location deviations, which is the distance between the ground truth location (OptiTrack) and the measured location (Wheel Encoder). From these figures, we can see that the deviation may reach up to 0.025m. As the rover moves further, the deviation could accumulate to a larger value. In fact, based on the hardware parameters, when the radar rotates at 2π rad/s, a 0.025m difference on the rover can make the virtual phase center deviate by more than 20 phase centers. Such a deviation may cause significant blurring of the radar image, which will be shown in the following two chapters. We can also observe the rover direction deviation was accumulating while moving from Fig. 4.9c and the final deviation is approximate 3° as shown in Fig. 4.9d. Figs. 4.9e and 4.9f present the rover speed measurement by both modalities and the speed deviations, respectively. It is clearly shown that the wheels were suffering from uneven surface roughness, since they frequently stopped rotating followed by a suddenly fast rotating. Fig. 4.10 shows that the rover is set to move in a loop. When involving turning, there is a big difference between the ground truth data and the measured data.



(a) Rover Location Measurement

(b) Rover Location Deviations

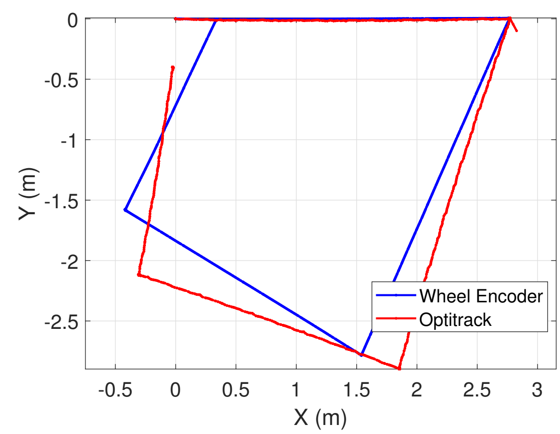
(c) Rover Direction Measurement

(d) Rover Direction Deviations

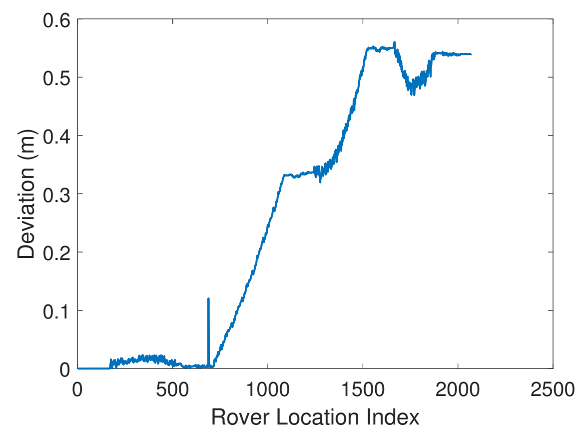
(e) Rover Speed Measurement

(f) Rover Speed Deviations

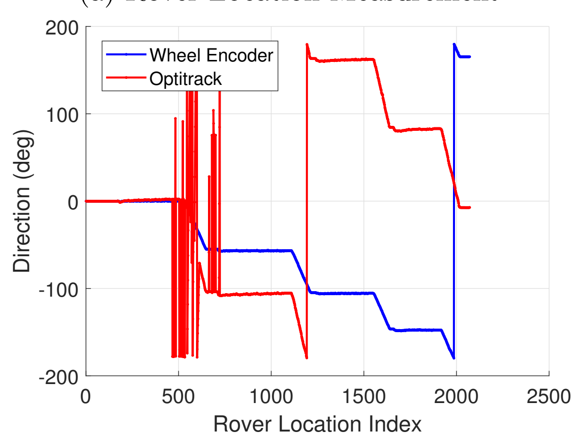
Figure 4.9: An Example of a Rover Moving in a Straight Line



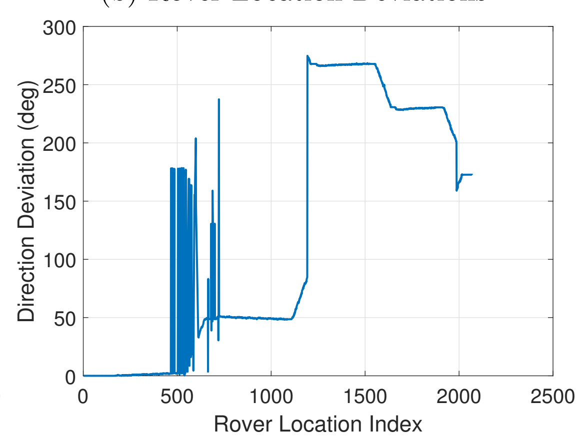
(a) Rover Location Measurement



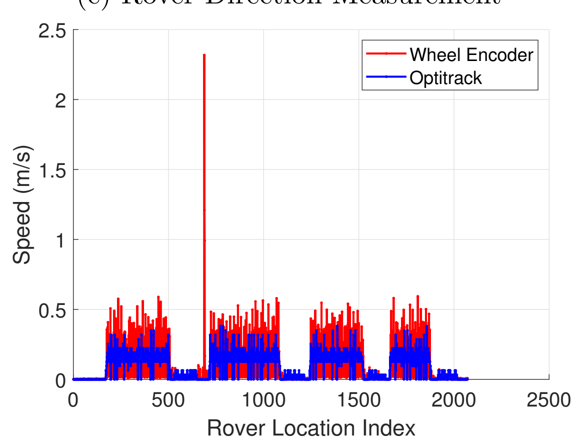
(b) Rover Location Deviations



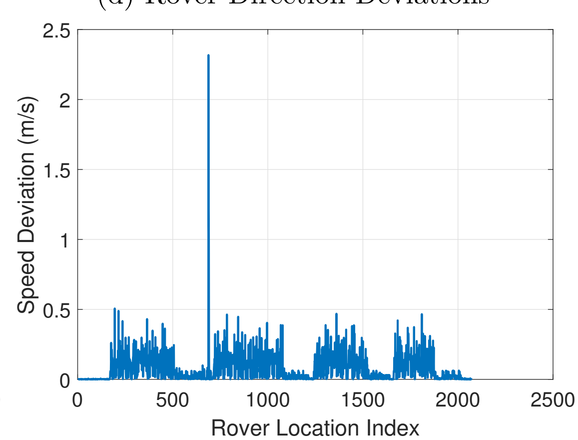
(c) Rover Direction Measurement



(d) Rover Direction Deviations



(e) Rover Speed Measurement



(f) Rover Speed Deviations

Figure 4.10: An Example of a Rover Moving in a Loop

Chapter 5

Efficient Rotating Synthetic Aperture Radar Imaging via Robust Sparse Array Synthesis

5.1 Introduction

SAR has been widely used in military reconnaissance and remote sensing, because of its all-weather all-day acquisition capabilities [67]. Conventional SAR working modes include “stripmap”, “spotlight” and “scan” [47]. In these modes, high range resolutions are achieved by transmitting large bandwidth signals, while the high resolution in the cross-range dimension is achieved by utilizing the Doppler effect induced by the relative motion between the radar platform and the target. However, the imaging swaths of these SAR modes are relatively small due to the limited beam footprint and the restricted moving track. Different from the aforementioned imaging schemes, ROSAR systems mount antennas on the edge of rotation platforms with a certain radius [36]. Through platform rotation, ROSAR systems are able to scan the surrounding environment continuously and generate a 360° image using the collected data from a single moving track [4]. ROSAR can overcome limited (angular) field-of-view of radar boards and allow imaging without translational movements of the platform making it a promising low-cost solution in helicopter-borne SAR imaging [39, 51, 75], indoor imaging [2] and so on. In indoor environments, ROSAR can be used for mapping and localization in case of fire emergencies or situations where other sensors fail due to high heat and low visibility.

Due to the highly non-linear moving track of ROSAR, BPA is typically employed, where its basic idea is to perform range-azimuth matched filtering with the prior knowledge of the distance between the target and each phase center. Although the conventional BPA can produce high-quality images without any limitation on the array geometry, it suffers extremely high computation complexity making it inadequate for real-time high-resolution imaging systems. The computational complexity of BPA is determined by the number of pixels, the number of fast-time samples per pulse and the number of pulses needed to generate one image. In a practical system, all three parameters can be very large: the number of pixels depends on the image resolution; a high pulse repetition frequency and consequently dense virtual array elements are required to avoid aliasing [72]; and a large signal bandwidth, which results in a large number of fast-time samples, is needed to ensure the high range-resolution. However, due to the unique array geometry of ROSAR, frequency domain processing algorithms such as Chirp Scaling Algorithm [53] and Omega-K [79] that assume linear motions of the radar platform relative to the scene are not applicable. In the past decades, much effort has been made in improving the efficiency of BPA and many algorithms have been proposed. For example, fast factorized Back-Projection (FFBP) [60, 76], Cartesian factorized BPA [19] and its variant [45]. The core idea of these algorithms is sub-aperture fusion. In sub-aperture fusion, the entire aperture is split into many small apertures and BPA is applied to each sub-aperture to obtain coarse images. A high-quality image can then be obtained by fusing these coarse-grained images together. However, all of them assume a linear aperture too and cannot be applied to circular aperture directly. In addition, sparse array synthesis is also a technique with low complexity. Conventional ways to select sparse elements, e.g., randomly or uniformly, are not optimal under every condition. Compressive sensing-based algorithms [11, 74] still suffer from high complexity, the requirement of sparse environment and the sensitivity to array manifold error.

In this chapter, we propose a new sparse array synthesis technique to reduce the computation complexity of BPA. The proposed algorithms are implemented in MATLAB. Extensive numerical simulations are conducted to evaluate the impact of the parameter settings on the sparsity of the design and array patterns. Additionally, we collect real-world data from indoor environments from the rotational hardware platform detailed in Chapter 4. The evaluation study shows that in both simulations and real experiments the proposed algorithm can reduce the total computation time by more than 90% while generating SAR images with comparable quality as BPA.

The rest of the chapter is organized as follows. Section 5.2 gives the system model of ROSAR and formulates the sparse array synthesis problem for ROSAR. Problem transformation and the solution approach are proposed in Section 5.3. Section 5.3.4 introduces range-dimension filtering using range-FFT to further reduce computation complexity. We validate our approach in Section 5.4 by numerical evaluation and simulation study as well as experiments in real environment in Section 5.4.5. Section 5.5 concludes the chapter.

5.2 System Model and Problem Formulation

In this section, we introduce the ROSAR system geometry, signal model, preprocessing steps and give the formal problem formulation of spare array design at the end.

5.2.1 Radar Geometry

Consider a stationary ROSAR system in Fig. 5.1. The radar is moving along the edge of a circle centered at the origin with radius r. The bore-sight of the antenna always faces outwards along the radial direction. The antenna radiation pattern in azimuth is assumed to be cosine-shape and non-zero within $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. The radar transmits chirp signals at a constant rate, e.g., N times per circle. Due to the symmetry, without loss of generality, we define a 2D coordinate frame such that a point target R_t distance away from the circle center locates at $(0, R_t)$ and the first (indexed by 0) phase center (with respect to the X-axis counter-clock wise) is at $(r, 0)$. Then, the bore-sight direction of the antenna at the n-th radar position (phase center) is

$$\phi_n = \frac{2\pi n}{N}, \tag{5.2.1}$$

where $n = 0, 1, .., N-1$. Let R_n and θ_n be the distance and the direction from the n-th phase center to the target with respect to its bore-sight direction, respectively. Due to the cosine antenna beam pattern, the target is in the field of view (FoV)

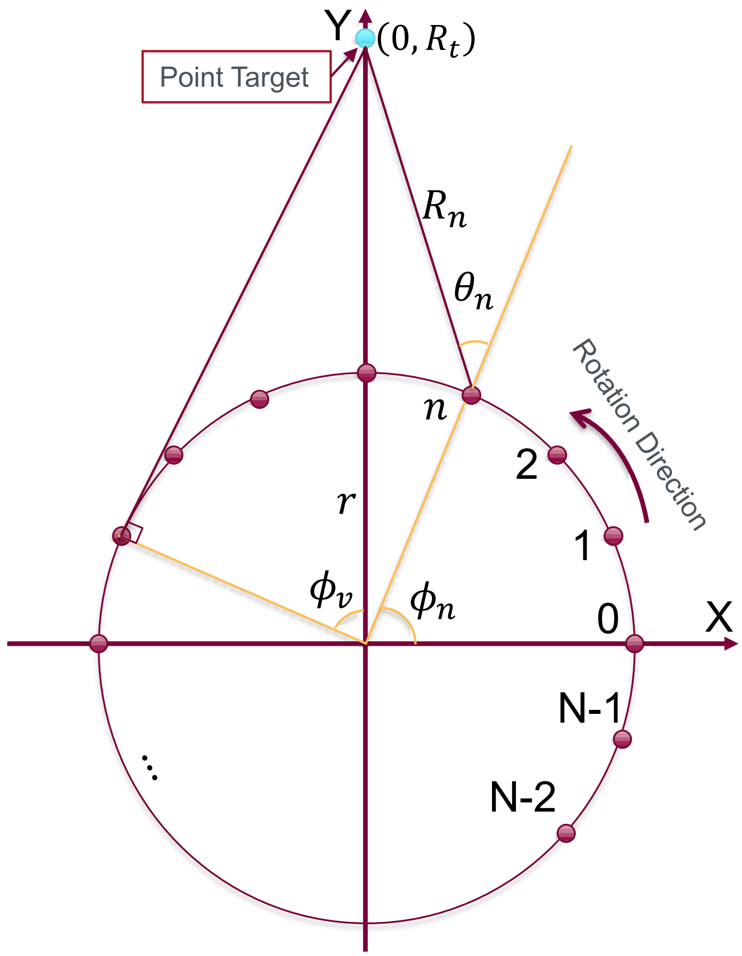


Figure 5.1: Imaging geometry of a ROSAR system

of only a subset of antenna positions. Denote by ϕ_v the angle such that if $\phi_n \in (\pi/2 - \phi_v, \pi/2 + \phi_v)$, the target is visible to the n-th phase center. R_n, θ_n and ϕ_v can be derived from trigonometry relationships, i.e.,

$$R_n = \sqrt{R_t^2 + r^2 - 2rR_t \cos(\phi_n - \phi_t)}, \tag{5.2.2}$$

$$\theta_n = \arctan \frac{R_t \cdot \sin(|\phi_n - \phi_t|)}{R_t \cdot \cos(|\phi_n - \phi_t|) - r}, \tag{5.2.3}$$

$$\phi_v = \arccos \frac{r}{R_t}. \tag{5.2.4}$$

The indices of the phase centers where the target is in their FOV are given by

$$n = N_{\min}, N_{\min} + 1, N_{\min} + 2, \ldots, N_{\max}, \tag{5.2.5}$$

where $N_{\min} = \left\lceil \frac{\frac{\pi}{2} - \phi_v}{\phi_\Delta} \right\rceil$, $N_{\max} = \left\lfloor \frac{\frac{\pi}{2} + \phi_v}{\phi_\Delta} \right\rfloor$ and $\phi_\Delta = \frac{2\pi}{N}$. To generate the image of the target, only signals received by those phase centers are used.

5.2.2 Signal Model and Preprocessing

Let the chirp signal transmitted by the radar be

$$x(t) = e^{j2\pi\left(f_c t + \frac{1}{2}Kt^2\right)}, \tag{5.2.6}$$

The received signal at the n-th phase center is

$$y_n(t) = \alpha_n e^{j2\pi\left[f_c(t-\tau_n) + \frac{1}{2}K(t-\tau_n)^2\right]} + v(t), \tag{5.2.7}$$

where α_n combines the complex reflection coefficient of the target, the antenna radiation pattern and channel fading, $\tau_n = 2R_n/c$ is the round-trip time delay. Specifically, the antenna radiation pattern is represented as

$$p\left(\theta\right) = \begin{cases} \cos\left(\theta\right) & \theta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right), \\ 0 & \text{otherwise.} \end{cases} \tag{5.2.8}$$

After down-converting and deramping, the resulting intermediate frequency (IF) signal is approximated to be

$$y_{\mathrm{IF},n}(t) \approx \alpha_n e^{j2\pi[\tau_n K t + f_c \tau_n]} + v_{\mathrm{IF}}(t), \tag{5.2.9}$$

and the sampled IF signal is

$$y_{\mathrm{IF},n}(m) = \alpha_n e^{j2\pi[\tau_n K(mt_s + T_{Start}) + f_c \tau_n]} + v_{\mathrm{IF}}(m), \tag{5.2.10}$$

where m is the sampling index, $0 \leq m \leq M-1$. Combining all the samples, we get the following vector representation

$$\mathbf{y}_{\mathrm{IF},n} = \begin{bmatrix} \alpha_n e^{j2\pi[\tau_n K(0\cdot t_s + T_{Start}) + f_c \tau_n]} \\ \alpha_n e^{j2\pi[\tau_n K(1\cdot t_s + T_{Start}) + f_c \tau_n]} \\ \vdots \\ \alpha_n e^{j2\pi[\tau_n K((M-1)\cdot t_s + T_{Start}) + f_c \tau_n]} \end{bmatrix} + \mathbf{v}_{\mathrm{IF}}, \tag{5.2.11}$$

where $\mathbf{v}_{\mathrm{IF}} = [v_{\mathrm{IF}}(0), v_{\mathrm{IF}}(1), \ldots, v_{\mathrm{IF}}(M-1)]^T$. Let $k = 2\pi\,(KT_{start} + f_c)/c$. Substituting the τ_n's in each entry by $2R_n/c$ and rearranging items in (5.2.11), we have

$$\mathbf{y}_{\mathrm{IF},n} = \begin{bmatrix} \alpha_n e^{j2\pi\tau_n K\cdot 0\cdot t_s} e^{j2kR_n} \\ \alpha_n e^{j2\pi\tau_n K\cdot 1\cdot t_s} e^{j2kR_n} \\ \vdots \\ \alpha_n e^{j2\pi\tau_n K\cdot (M-1)\cdot t_s} e^{j2kR_n} \end{bmatrix} + \mathbf{v}_{\mathrm{IF}}. \tag{5.2.12}$$

Now, the data matrix from the effective phase centers used for SAR is

$$\mathbf{Y}_{\text{IF}} = \left[\mathbf{y}_{\text{IF},N_{\min}}, \mathbf{y}_{\text{IF},N_{\min}+1}, \ldots, \mathbf{y}_{\text{IF},N_{\max}}\right]. \tag{5.2.13}$$

The conventional BPA images a point with parameter (ϕ_t, R_t) by computing the Hadamard product of $\mathbf{Y}_{\text{IF}}$ and a matrix $\mathbf{W}_{\text{BP}}$ of size $M \times (N_{\max} - N_{\min} + 1)$, where the element locating at the i-th row and j-th column is

$$\mathbf{W}_{\text{BP},(i,j)} = \alpha_{N_{\min}+j-1} e^{-j2\pi\tau_{N_{\min}+j-1}K(i-1)t_s} e^{-j2kR_{N_{\min}+j-1}}.$$

Then, the intensity of the point is

$$I\left(\phi_t, R_t\right) = \mathbf{1}^T \cdot \left(\mathbf{W}_{\text{BP}} \odot \mathbf{Y}_{\text{IF}}\right) \cdot \mathbf{1}, \tag{5.2.14}$$

If the target indeed locates at (ϕ_t, R_t) in polar coordinates, all the phases of the sampled data are perfectly compensated, and (5.2.14) achieves its maximum. However, the computation complexity of imaging a rectangular area using conventional BPA is $O\left(L_x \times L_y \times M \times N\right)$, where L_x and L_y is the number of grids along X and Y direction of the area. Clearly, the complexity grows linearly with M, N and area size. From the complexity analysis, it can be deduced that two possible ways to lower the complexity of BPA are, (1) reducing the number of phase centers to be used, i.e., reducing N, and (2) apply range-dimension matched filtering and select the appropriate range bin instead of using all M data samples in each pulse.

In the subsequent sections, we first develop a Sparse Array Synthesis method that selects a subset of the phase centers and assigns appropriate complex weights. Then, we investigate the use of range-dimension matched filter (or more commonly known as range FFT) to further reduce computation complexity. The two approached are abbreviated to "SAS" and "FFT+SAS" respectively for simplicity.

5.2.3 Problem Formulation for Robust Sparse Array Synthesis

For simplicity, we assume that the reflection coefficient is always 1. Due to complex multipath reflection, wall penetration in indoor environments and the small diameter of the rotation platform relative to the dimension of the environment, the channel fading factor can be approximated to be a constant for the same range bin in all directions and thus we omit it in the formulation. Thus,

$$\alpha_n = p\left(\theta_n\right). \tag{5.2.15}$$

BPA can be viewed as a form of range-azimuth two-dimension filtering. To generalize it to sparsely-selected phase centers, we first apply a compensation matrix to $\mathbf{Y}_{\mathrm{IF}}$ to remove the phase items related to fast-time sampling, i.e.,

$$\begin{aligned}
\mathbf{Y}'_{\mathrm{IF}} &= \mathbf{W}_{\mathrm{SA}} \odot \mathbf{Y}_{\mathrm{IF}}, \\
&= \begin{bmatrix}
\alpha_{N_{\min}} e^{j2kR_{N_{\min}}} & \cdots & \alpha_{N_{\max}} e^{j2kR_{N_{\max}}} \\
\alpha_{N_{\min}} e^{j2kR_{N_{\min}}} & \cdots & \alpha_{N_{\max}} e^{j2kR_{N_{\max}}} \\
\vdots & \ddots & \vdots \\
\alpha_{N_{\min}} e^{j2kR_{N_{\min}}} & \cdots & \alpha_{N_{\max}} e^{j2kR_{N_{\max}}}
\end{bmatrix},
\end{aligned}$$

where the i-th row and j-th column element of $\mathbf{W}_{\mathrm{SA}}$ is $\mathbf{W}_{\mathrm{SA},(i,j)} = e^{-j2\pi\tau_{N_{\min}+j-1}K\cdot(i-1)\cdot t_s}$. The steering vector of the ROSAR array to a near-field target located at range R can be represented as

$$\mathbf{a}(\phi;R) = \begin{bmatrix} \cos(\theta'_{N_{\min}})e^{j2k\sqrt{R^2+r^2-2Rr\cos(\phi-\phi_{N_{\min}})}} \\ \cos(\theta'_{N_{\min}+1})e^{j2k\sqrt{R^2+r^2-2Rr\cos(\phi-\phi_{N_{\min}+1})}} \\ \vdots \\ \cos(\theta'_{N_{\max}})e^{j2k\sqrt{R^2+r^2-2Rr\cos(\phi-\phi_{N_{\max}})}} \end{bmatrix}, \tag{5.2.16}$$

where $\theta'_n = \arctan\frac{R\sin(|\phi-\phi_n|)}{R\cos(|\phi-\phi_n|)-r}$, and the array pattern can also be calculated as

$$F(\phi;R) = \mathbf{w}^H\mathbf{a}(\phi;R), \tag{5.2.17}$$

where $\mathbf{w}$ is a sparse complex weight vector to be designed and some of its elements can be equal or be close to zero. Note that, $\mathbf{w}$ is also a function of R, but we omit the subscript R for simplicity. To focus a point target locating at (ϕ_t, R_t), we need to compute

$$\begin{aligned} I\left(\phi_t, R_t\right) &= \mathbf{1}^T \cdot \left(\mathbf{w}^H \circ \mathbf{Y}'_{\mathrm{IF}}\right) \cdot \mathbf{1} \\ &= \mathbf{1}^T \cdot \left(\mathbf{w}^H \circ \mathbf{W}_{\mathrm{SA}} \odot \mathbf{Y}_{\mathrm{IF}}\right) \cdot \mathbf{1}, \end{aligned}$$

where $\circ$ is the Khatri-Rao product.

Due to the vibration of the rotation platform, odometry errors and antenna pattern mismatch (e.g., cosine pattern), there exist array manifold errors, which may lead to blurred images. To obtain an SAR image with good quality in this situation, the sparse weight vector $\mathbf{w}$ must be carefully designed with consideration of robustness to array errors.

We formulate the way to obtain the desirable $\mathbf{w}$ as to solve an optimization problem. In the optimization, the first constraint is that the power of the main-lobe peak, locating at $\phi_m = \frac{\pi}{2}$, should be larger than or equal to a threshold U, i.e.,

$$\left|\mathbf{w}^H \left(\mathbf{a}\left(\phi_m; R\right) + \mathbf{e}_m\right)\right|^2 \geq U, \quad \|\mathbf{e}_m\| \leq \Delta_R, \tag{5.2.18}$$

where $\mathbf{e}_m$ is the array error vector caused by measurement and imperfect radar rotations, and we assume that $\mathbf{e}_m$ is bounded by a ball with radius Δ_R ($0 \leq \Delta_R \leq \|\mathbf{a}(\phi; R)\|$). Second, to restrict the main-lobe width and the sidelobe level, we put limitations on the received power at some uniformly spaced discrete directions except for the desired main lobe area, i.e.,

$$\left|\mathbf{w}^H \left(\mathbf{a}\left(\phi_s; R\right) + \mathbf{e}_s\right)\right|^2 \leq \eta U, \quad s = 1, 2, \ldots, S; \|\mathbf{e}_s\| \leq \Delta_R, \tag{5.2.19}$$

where $\mathbf{e}_s$ is the error vector for sidelobe area with the same properties as $\mathbf{e}_m$, S is the number of uniformly spaced discrete directions, $\phi_s \in \left[\phi_{N_{\min}}, \frac{\pi}{2} - \phi_{\mathrm{MW}}\right] \cup \left[\frac{\pi}{2} + \phi_{\mathrm{MW}}, \phi_{N_{\max}}\right]$, ϕ_{MW} is the half of the desirable main-lobe width, η is the preset power ratio of the main-lobe to the sidelobe. Third, to avoid amplifying the noise level, we impose a constraint on the gain of noise power:

$$\|\mathbf{w}\|_2^2 = 1. \tag{5.2.20}$$

Lastly, to guarantee a sufficient gain on the target, we set another constraint $U \geq U_{\min}$.

The objective is to minimize the number of virtual phase centers given by $\|\mathbf{w}\|_0$. To this end, we formulate the sparse array synthesis problem as

$$\min_{\mathbf{w},U,\mathbf{e}_m,\mathbf{e}_s} \|\mathbf{w}\|_0$$

$$s.t. \begin{cases} C1 : \left|\mathbf{w}^H\left(\mathbf{a}\left(\phi_m;R\right)+\mathbf{e}_m\right)\right|^2 \geq U, \|\mathbf{e}_m\| \leq \Delta_R, \\ C2 : \left|\mathbf{w}^H(\mathbf{a}(\phi_s;R)+\mathbf{e}_s)\right|^2 \leq \eta U, s=1,2,\ldots,S; \|\mathbf{e}_s\| \leq \Delta_R, \\ C3 : \|\mathbf{w}\|_2^2 = 1, \\ C4 : U \geq U_{\min}. \end{cases} \tag{5.2.21}$$

Problem (5.2.21) is a non-convex optimization problem since both the objective function and constraints C1 and C3 are non-convex. Because of the consideration of robustness to array errors, the problem formulation is markedly different from those in conventional sparse array synthesis[50, 25, 43, 42, 35, 22], rendering existing techniques inapplicable. In the next section, we develop a customized algorithm based on FPP and SCA to solve (5.2.21).

5.3 Solution Approach for Robust Sparse Array Synthesis

5.3.1 Problem Transformation

Directly solving (5.2.21) is hard, since l^0-norm minimization problem requires intractable combinatorial search. To reduce the complexity, we replace the l^0-norm objective function with l^1-norm, i.e., $\|\mathbf{w}\|_1$ as suggested by [13, 20].

C1 and C2 contain additional control variables $\mathbf{e}_m$ and $\mathbf{e}_s$ to express robustness constraints. They can be simplified by considering the worst case scenarios. Specifically, by using the Cauchy-Schwarz inequality and the triangle inequality, we can find the minimum of main-lobe response and the maximum of sidelobe response respectively as follows

$$\begin{aligned}\left|\mathbf{w}^H(\mathbf{a}(\phi_m;R)+\mathbf{e}_m)\right|^2 =&\left|\mathbf{w}^H\mathbf{a}(\phi_m;R)+\mathbf{w}^H\mathbf{e}_m\right|^2\\ \geq&(|\mathbf{w}^H\mathbf{a}(\phi_m;R)|-\left|\mathbf{w}^H\mathbf{e}_m\right|)^2\\ \geq&(|\mathbf{w}^H\mathbf{a}(\phi_m;R)|-\|\mathbf{w}\|_2\|\mathbf{e}_m\|_2)^2\\ \geq&(|\mathbf{w}^H\mathbf{a}(\phi_m;R)|-1\cdot\Delta_R)^2,\end{aligned} \tag{5.3.1}$$

$$\begin{aligned}\left|\mathbf{w}^H(\mathbf{a}(\phi_s;R)+\mathbf{e}_s)\right|^2 =&\left|\mathbf{w}^H\mathbf{a}(\phi_s;R)+\mathbf{w}^H\mathbf{e}_s\right|^2\\ \leq&(|\mathbf{w}^H\mathbf{a}(\phi_s;R)|+\left|\mathbf{w}^H\mathbf{e}_s\right|)^2\\ \leq&(|\mathbf{w}^H\mathbf{a}(\phi_s;R)|+\|\mathbf{w}\|_2\|\mathbf{e}_s\|_2)^2\\ \leq&(|\mathbf{w}^H\mathbf{a}(\phi_s;R)|+1\cdot\Delta_R)^2.\end{aligned} \tag{5.3.2}$$

The equalities hold when $\mathbf{e} = a \cdot \mathbf{a}(\phi; R)$ $(a \in \mathbb{R})$. Substituting (5.3.1) and (5.3.2) into C1 and C2, we obtain the worst-case constraints as

$$(|\mathbf{w}^H\mathbf{a}(\phi_m;R)|-\Delta_R)^2 \;\geq\; U \tag{5.3.3}$$

$$(|\mathbf{w}^H\mathbf{a}(\phi_s;R)|+\Delta_R)^2 \;\leq\; \eta U, s=1,2,\ldots,S \tag{5.3.4}$$

Taking the square root of both sides of (5.3.3) and (5.3.4), and re-arranging items in the inequalities, we have

$$C1:(U^{'}+\Delta_R)^2-\mathbf{w}^H\mathbf{a}(\phi_m;R)\mathbf{a}^H(\phi_m;R)\mathbf{w}\leq 0, \tag{5.3.5}$$

$$C2 : \mathbf{w}^H \mathbf{a}(\phi_s; R)\mathbf{a}^H(\phi_s; R)\mathbf{w} - (\sqrt{\eta}U^{'} - \Delta_R)^2 \leq 0, s = 1, 2, \ldots, S, \tag{5.3.6}$$

where $U^{'} = \sqrt{U}$. Constraint $C3$ in (5.2.21) can be replaced by two inequality constraints as

$$C3 : \|\mathbf{w}\|_2^2 - 1 \leq 0, \tag{5.3.7}$$

$$C4 : 1 - \|\mathbf{w}\|_2^2 \leq 0, \tag{5.3.8}$$

and $C4$ is replaced by

$$C5 : U^{'} \geq U^{'}_{\min}, \tag{5.3.9}$$

where $U^{'}_{\min} = \sqrt{U_{\min}}$. The optimization problem is thus transformed to

$$\min_{\mathbf{w}, U^{'}} \|\mathbf{w}\|_1$$

$$s.t. \begin{cases} C1 : (U^{'} + \Delta_R)^2 - \mathbf{w}^H \mathbf{a}(\phi_m; R)\mathbf{a}^H(\phi_m; R)\mathbf{w} \leq 0, \\ C2 : \mathbf{w}^H \mathbf{a}(\phi_s; R)\mathbf{a}^H(\phi_s; R)\mathbf{w} - (\sqrt{\eta}U^{'} - \Delta_R)^2 \leq 0, s = 1, 2, \ldots, S, \\ C3 : \|\mathbf{w}\|_2^2 - 1 \leq 0, \\ C4 : 1 - \|\mathbf{w}\|_2^2 \leq 0, \\ C5 : U^{'} \geq U^{'}_{\min}. \end{cases} \tag{5.3.10}$$

5.3.2 FPP-SCA-based Algorithm

The reformulated problem (5.3.10) is still non-convex which is hard to be solved directly, and it is also tricky to find feasible initial solutions, but it is now amenable to the convex approximation technique. Inspired by the idea of FPP [46], we introduce three slack variables b, b_1, b_2 ($b, b_1, b_2 > 0$) for C1–C4, and we construct the following slacked surrogate problem of (5.3.10)

$$\min_{\mathbf{w},U',b,b_1,b_2} \|\mathbf{w}\|_1 + \lambda_b\,(b + b_1 + b_2)$$

$$s.t. \begin{cases} C1: (U' + \Delta_R)^2 - \mathbf{w}^H\mathbf{a}(\phi_m;R)\mathbf{a}^H(\phi_m;R)\mathbf{w} - b_1 \leq 0, \\ C2: \mathbf{w}^H\mathbf{a}(\phi_s;R)\mathbf{a}^H(\phi_s;R)\mathbf{w} - (\sqrt{\eta}U' - \Delta_R)^2 - b_2 \leq 0, s = 1,2,\ldots,S, \\ C3: \|\mathbf{w}\|_2^2 - 1 - b \leq 0, \\ C4: 1 - b - \|\mathbf{w}\|_2^2 \leq 0, \\ C5: U' \geq U'_{\min}, \end{cases} \tag{5.3.11}$$

where λ_b ($\lambda_b > 0$) is a pre-set coefficient to penalize constraint violations.

To handle the non-convex terms in above constraints, i.e., $-\mathbf{w}^H\mathbf{a}\left(\phi_m;R\right)\mathbf{a}^H\left(\phi_m;R\right)\mathbf{w}$ in $C1$, $-(\sqrt{\eta}U' - \Delta_R)^2$ in $C2$ and $-\|\mathbf{w}\|_2^2$ in $C4$, we next apply SCA [9]. SCA is an iterative method and its core idea is to replace non-convex terms with convex approximations (usually upper bounds of these non-convex terms) in each iteration. In our case, given $\{\mathbf{w}_{(i)}, U'_{(i)}\}$ after the i-th iteration, the convex upper bounds of the three non-convex terms are derived by applying their respective first-order Taylor expansions as

$$\begin{aligned} & -\mathbf{w}^H\mathbf{a}\left(\phi_m;R\right)\mathbf{a}^H\left(\phi_m;R\right)\mathbf{w} \\ \leq & -\mathbf{w}_{(i)}^H\mathbf{a}\left(\phi_m;R\right)\mathbf{a}^H\left(\phi_m;R\right)\mathbf{w}_{(i)} \\ & -2\mathrm{Re}\{\mathbf{w}_{(i)}^H\mathbf{a}\left(\phi_m;R\right)\mathbf{a}^H\left(\phi_m;R\right)\left(\mathbf{w} - \mathbf{w}_{(i)}\right)\} \\ = & \mathbf{w}_{(i)}^H\mathbf{a}\left(\phi_m;R\right)\mathbf{a}^H\left(\phi_m;R\right)\mathbf{w}_{(i)} \\ & -2\mathrm{Re}\{\mathbf{w}_{(i)}^H\mathbf{a}\left(\phi_m;R\right)\mathbf{a}^H\left(\phi_m;R\right)\mathbf{w}\}, \end{aligned} \tag{5.3.12}$$

$$
\begin{aligned}
&-(\sqrt{\eta}U^{'}-\Delta_R)^2\\
\leq&-(\sqrt{\eta}U^{'}_{(i)}-\Delta_R)^2-2(\eta U^{'}_{(i)}-\sqrt{\eta}\Delta_R)(U^{'}-U^{'}_{(i)})\\
=&2(\sqrt{\eta}\Delta_R-\eta U^{'}_{(i)})U^{'}+\eta U^{'2}_{(i)}-\Delta_R^2,
\end{aligned}
\tag{5.3.13}
$$

$$
\begin{aligned}
-\left\|\mathbf{w}\right\|_2^2\leq&-\left\|\mathbf{w}_{(i)}\right\|_2^2-2Re\{\mathbf{w}_{(i)}^H\left(\mathbf{w}-\mathbf{w}_{(i)}\right)\}\\
=&\left\|\mathbf{w}_{(i)}\right\|_2^2-2Re\{\mathbf{w}_{(i)}^H\mathbf{w}\}.
\end{aligned}
\tag{5.3.14}
$$

Replacing the non-convex terms in C1, C2 and C4 with the convex upper bounds in (5.3.12)–(5.3.14), we finally transform (5.2.21) to a convex sub-problem in (5.3.15).

$$
\begin{aligned}
&\min_{\mathbf{w},U^{'},b,b_1,b_2} \left\|\mathbf{w}\right\|_1+\lambda_b\left(b+b_1+b_2\right)\\
s.t.&\begin{cases}
C1{:}(U^{'}+\Delta_R)^2+\mathbf{w}_{(i)}^H\mathbf{a}\left(\phi_m;R\right)\mathbf{a}^H\left(\phi_m;R\right)\mathbf{w}_{(i)}-2\mathrm{Re}\{\mathbf{w}_{(i)}^H\mathbf{a}\left(\phi_m;R\right)\mathbf{a}^H\left(\phi_m;R\right)\mathbf{w}\}-b_1\leq 0,\\
C2:\mathbf{w}^H\mathbf{a}\left(\phi_s;R\right)\mathbf{a}^H\left(\phi_s;R\right)\mathbf{w}+2(\sqrt{\eta}\Delta_R-\eta U^{'}_{(i)})U^{'}+\eta {U^{'}_{(i)}}^2-\Delta_R^2-b_2\leq 0,\\
s=1,2,\ldots,S,\\
C3:\left\|\mathbf{w}\right\|_2^2-1-b\leq 0,\\
C4:1-b+\left\|\mathbf{w}_{(i)}\right\|_2^2-2\mathrm{Re}\{\mathbf{w}_{(i)}^H\mathbf{w}\}\leq 0,\\
C5:U^{'}\geq U^{'}_{\min}.
\end{cases}
\end{aligned}
\tag{5.3.15}
$$

Repeatedly solving (5.3.15) with the values from the previous iteration until the number of iterations reaches a pre-set value $ITER$. Let $J^{(i)}=\|(\mathbf{w})_i^{\star}\|_1+\lambda_b(b^{(i)}+b_1^{(i)}+b_2^{(i)})$ be the value of the objective function after the i-th iteration, $\mathbf{R}$ be a vector of target range bins of concern, $\Delta_{\mathbf{R}}$ be a vector of the maximum l^2-norm of $\mathbf{e}$ for all range bins. Algorithm 3 and 4 summarize the proposed approaches to determine the sparse weight vector for each range bin and to apply the resulting weight vector for

SAR imaging, respectively.

Algorithm 3 Computing sparse weight vectors

Input $\mathbf{R}$, $\Delta_{\mathbf{R}}$, ϕ_{MW}, η, λ_b, N, J, $\mathbf{w}_{(0)}$, $U^{'}_{(0)}$;
$W^{\star} \leftarrow \varnothing$; $i \leftarrow 0$;
for each R, Δ_R **do**$\mathbf{R}$, $\Delta_{\mathbf{R}}$
  Calculate ϕ_v, ϕ_Δ, $N_{\min}$, $N_{\max}$, $a\left(\phi_m; R\right)$, $a\left(\phi_s; R\right)$
  repeat
    $i \leftarrow i + 1$;
    Compute $\mathbf{w}_{(i)}$, $U^{'}_{(i)}$ by solving (5.3.15) with $\mathbf{w}_{(i-1)}$, $U^{'}_{(i-1)}$;
  until $i > ITER$
  Save $\mathbf{w}_{(i)}$ to the set $W^{\star}$;
end for
Output $W^{\star}$.

Algorithm 4 The proposed approach for ROSAR imaging

Input $\mathbf{Y}_{\mathrm{IF}}$, R_t, ϕ_t;
Compute the index of range bin, i.e., $\mathcal{I}_{R_t}$, for R_t;
Compute the central index of the phase center $\mathcal{I}_{\phi_t} \leftarrow round(\phi_t/\phi_\Delta)$;
$\mathbf{w}^{\star} \leftarrow W^{\star}\left(\mathcal{I}_{R_t}\right)^1$;
$\mathbf{Y}^{'}_{\mathrm{IF}} \leftarrow \mathbf{Y}_{\mathrm{IF}}(:, \mathcal{I}_{\phi_t} - \left(|\mathbf{w}^{\star}| - 1\right)/2 : \mathcal{I}_{\phi_t} + \left(|\mathbf{w}^{\star}| - 1\right)/2)$;
Output $I(\phi_t, R_t) \leftarrow \mathbf{1}^T \cdot (\mathbf{w}^{\star H} \circ (\mathbf{W}_{\mathrm{SA}} \odot \mathbf{Y}^{'}_{\mathrm{IF}})) \cdot \mathbf{1}$.

5.3.3 Initial values and parameter settings for Algorithms 3 and 4

Initial Value Settings

The proposed approach is able to work with any initial values of the control variables since the constraints are always feasible due to the introduced slack variables. In our implementation, the initial value of $\mathbf{w}$ is chosen as $\mathbf{w}_{(0)} = \frac{1}{\sqrt{N_{\max} - N_{\min} + 1}}\mathbf{1}$ and $U^{'}_{(0)} = |\mathbf{w}^H_{(0)}\mathbf{a}(\phi; R)|^2$.

[1] $W^{\star}(i)$ represents the i-th vector in the ordered set $W^{\star}$.

Parameter Settings

To ensure convergence, $ITER$ and Th can be set to around $50 \sim 100$ and $10^{-4} \sim 10^{-3}$, respectively. ϕ_{MW} and η are design parameters of the sparse array. We found that any value lower than both $\phi_{\mathrm{MW}} = 1^{\circ}$ and $\eta = -33dB$ makes the SCA diverge. Δ_R should be chosen by considering practical limitations of target platforms. For example, in Section 5.4.1, the simulation setup and experiments take the unstable rotation speed of a ROSAR platform into account. The direction of each phase center under unstable rotations is modeled as a Gaussian distribution

$$\widehat{\phi}_n \sim \mathcal{N}\left(\frac{2\pi n}{N}, \sigma\right), \tag{5.3.16}$$

where the σ is the standard deviation of the direction. The error vector is computed from $\mathbf{e} = \widehat{\mathbf{a}}(\phi;\ R) - \mathbf{a}(\phi;\ R)$, where $\widehat{\mathbf{a}}(\phi; R)$ is determined by substituting ϕ_n with $\widehat{\phi}_n$. Let $\widehat{\Delta}_R = \|\mathbf{e}\|$. By repeatedly sampling from (5.3.16), we can obtain the cumulative distribution function of $\widehat{\Delta}_R$, and choose the $\widehat{\Delta}_R$ corresponding to 99% of the cumulative probability as Δ_R. The testbed evaluations in Section 5.4.5 show that this model is reasonable in realistic settings. Note since the phase error $\mathbf{e}$ differs among range bins, one specific Δ_R must be pre-computed for each range bin.

To ensure accurate results, the angle interval should be less than or equal to the angular resolution of ROSAR. However, reducing the angle interval increases the number of grid points and leads to higher computation costs. Since there is no closed-form solution to the angular resolution of a circular array, we use the results of a linear array as a reference. The determination of $U_{\min}$ is based on the expected image quality in target applications. It should be large enough to guarantee a sufficient gain for all range bins. Otherwise, there could be light and dark strips on the generated SAR image.

In (5.3.15), the purpose of the slack variables b, b_1, b_2 is to penalize constraint violation. The penalty coefficient λ_b plays a crucial role in balancing the trade-off between sparsity and the magnitudes of slackness in constraints. A large λ_b encourages stricter adherence to the constraints, but it is challenging to find the sparse $\mathbf{w}^\star$ within a small feasible region. Conversely, a small λ_b can lead to the significant violations of the constraints, e.g., sidelobe levels exceeding the preset limits, which may further reduce the quality of the generated SAR images. As a general rule of thumb, λ_b should be several times larger than the maximum of $\|\mathbf{w}\|_1$ to ensure that the slack variables are close to zero. Since the l^1-norm differs l^0-norm and cannot enforce sparsity in itself, we must manually set any term in $\mathbf{w}^\star$ lower than a pre-defined threshold to 0. Thus, in Algorithm 4, only the nonzero entries in $\mathbf{w}^\star$ are included in computing $\mathbf{w}^{\star H} \circ \mathbf{W}_{\text{SA}} \odot \mathbf{Y}'_{\text{IF}}$. Moreover, a final step must be taken to verify the solution. This can be accomplished by checking if the slack variables are sufficiently small, i.e., $b + b_1 + b_2 < b_{\min}$ and $b_{\min}$ is set to 10^{-5} in the experiments.

5.3.4 Further Complexity Reduction

Since applying range compression to the fast time samples [16] (denoted as "FFT+BPA") can reduce the processing time of conventional BPA, we borrow its idea to further reduce the complexity of the proposed SAS approach (denoted as "FFT+SAS"). Ignoring the noise term in (5.2.10) and applying range-FFT to $\mathbf{y}_{\mathrm{IF},n}$, we have

$$\begin{aligned} Y_{\mathrm{1D},n}(l) &= \sum_{m=0}^{M-1} y_{\mathrm{IF},n}(m) e^{-j2\pi\frac{l}{L}m} \\ &= \alpha_n e^{j2\pi(\tau K T_{Start}+f_c\tau)} \sum_{m=0}^{M-1} e^{j2\pi\left(\tau K t_s-\frac{l}{L}\right)m} \\ &= \alpha_n e^{j4\pi\frac{KT_{Start}+f_c}{c}R_n} \sum_{m=0}^{M-1} e^{j2\pi\left(\frac{2R_nKt_s}{c}-\frac{l}{L}\right)m}, \end{aligned} \tag{5.3.17}$$

where $l = 0, 1, \ldots, L-1$ and L is the number of range bins. Let $k = \frac{2\pi(KT_{\mathrm{Start}}+f_c)}{c}$. We have

$$Y_{\mathrm{1D},n}(l) = \alpha_n e^{j2kR_n} \sum_{m=0}^{M-1} e^{j2\pi\left(\frac{2R_nKt_s}{c}-\frac{l}{L}\right)m}, \tag{5.3.18}$$

and $Y_{\mathrm{1D},n}(l)$ reaches the maximum for

$$l_n^\star = round\left(\frac{2R_nKt_sL}{c}\right), \tag{5.3.19}$$

which is the range bin where the target is located. The data vector at the n-th phase center now becomes

$$\mathbf{y}_{\mathrm{1D},n} = \left[Y_{\mathrm{1D},n}(0), Y_{\mathrm{1D},n}(1) \ldots, Y_{\mathrm{1D},n}(L-1)\right]^T. \tag{5.3.20}$$

The data matrix from effective phase centers used for SAR is given by

$$\mathbf{Y}_{\mathrm{1D}} = \left[\mathbf{y}_{\mathrm{1D},N_{\min}}, \mathbf{y}_{\mathrm{1D},N_{\min}+1}, \ldots, \mathbf{y}_{\mathrm{1D},N_{\max}}\right]. \tag{5.3.21}$$

To focus a point locating at (ϕ_t, R_t) in polar coordinates, we need to compute

$$I(\phi_t, R_t) = \mathbf{w}^H \cdot \mathbf{Y}_{1\mathrm{D}}(l^\star_{N_{\min}}, l^\star_{N_{\min}+1}, \ldots, l^\star_{N_{\max}}), \tag{5.3.22}$$

where $\mathbf{Y}_{1\mathrm{D}}(l^\star_{N_{\min}}, l^\star_{N_{\min}+1}, l^\star_{N_{\min}+2}, \ldots, l^\star_{N_{\max}}) = [\mathbf{y}_{1\mathrm{D},N_{\min}}(l^\star_{N_{\min}}), \mathbf{y}_{1\mathrm{D},N_{\min}+1}(l^\star_{N_{\min}+1}), \ldots, \mathbf{y}_{1\mathrm{D},N_{\max}}(l^\star_{N_{\max}})]^T$ and $\mathbf{y}_{1\mathrm{D},n}(l)$ represents the l-th entry of the vector $\mathbf{y}_{1\mathrm{D},n}$. Vector $\mathbf{w}$ in (5.3.22) can be the sparse weight vector of the corresponding range bin determined by Algorithm 2, or the weight vector of BPA with the j-th entry being $\mathbf{w}_{\mathrm{BP},j} = \alpha_{N_{\min}+j-1} \cdot e^{-j2kR_{N_{\min}+j-1}}$.

Remark 3 *The steering vector may differ from the original one in* (5.2.16) *after applying range-FFT, since substituting* (5.3.19) *into* (5.3.18) *cannot fully cancel the phase summation term in some cases (e.g., when R_n is not multiple of the length of range bin). In our implementation, $\mathbf{w}^\star$ is still obtained from the original steering vector. Thus, the array pattern could deviate from the desired one. However, doing so can lead to computation reduction.*

Remark 4 *The computation complexity of SAS is $O(L_x \times L_y \times M \times N')$, where N' is number of phase centers corresponding to non-zero weights. N' is typically less than a half of N. As for FFT+SAS, the computation complexity is given by $O(N' \times M \log_2 M + L_x \times L_y \times N')$. When $L_x \times L_y \gg M$, the second term dominates. Thus, the overall reduction in complexity by combining FFT and SAS is substantial compared with that of the conventional BPA algorithm. Since the proposed algorithms conduct filtering pixel-by-pixel independently, they can be further accelerated by separating these pixels into multiple groups and processing them in a parallel manner.*

5.4 Performance Evaluation

In this section, we conduct experimental study to evaluate the effectiveness of the proposed ROSAR imaging algorithms.

5.4.1 Implementation and Parameter Settings

We implement the proposed algorithms in MATLAB using the Phased Array Toolbox on a PC equipped with an Intel Core 8700 CPU and 16GB RAM. Our ROSAR system is used for real-world data collection (shown in Fig. 4.7). The detailed parameter settings of the ROSAR system for this work are summarized in Table 5.1. Based on the settings, the range resolution is calculated as $R_\Delta = \frac{c}{2B} = 0.0435m$, where B is the bandwidth of the sampled chirp signal. The maximum unambiguous range is $R_{\max} = \frac{cF_s}{4K} \approx 4.8686m$. If we choose Δ_R to represent 99% of the probability of the CDF, Fig. 5.2a shows an example CDF for $R = 2$ and $\Delta_R = 0.035$. Fig. 5.2b shows Δ_R as a function of range, from which we can see Δ_R decreases as the range becomes larger. It is because a small displacement from the desirable positions of phase centers has less impact when the radar is further away from the target. The detailed calculation steps have been given in Section 5.3.3.

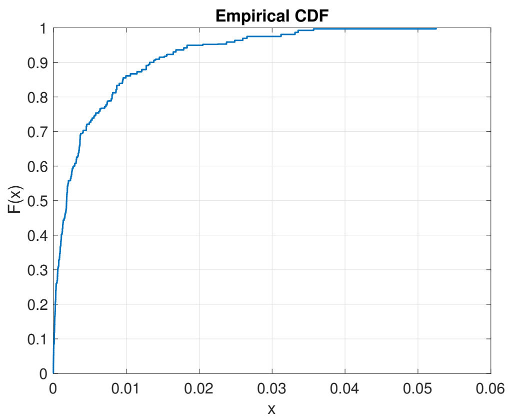
All the sparse weight vector $\mathbf{w}$ for each range bin are computed in advance using Algorithm 3 with parameters listed in Table 5.2. We use CVX and Mosek solver [28] to find the optimal values in each iteration of SCA.

Table 5.1: Parameters of the ROSAR System

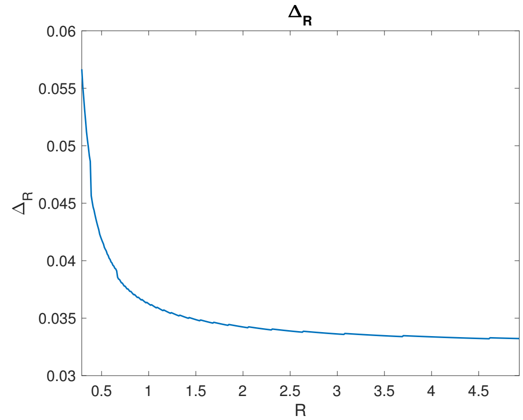
Radar Settings	
r	0.145 m
Rotation speed	60 RPM
Rotation time	1 s
# Of TX	1
# Of RX	1
Antenna Pattern	Cosine
Antenna FOV	[-90°, 90°]
Start rotating direction	0°
Rotation direction	Counter-clockwise
TX power	12 dBm
Antenna gain	7 dBi
RX gain	48 dB
Chirp Signal Settings	
Start Frequency	60 GHz
End Frequency	64 GHz
Ramp start time	0 us
Ramp end time	58 us
T_{Start}	7 us
Sampling end time	57 us
F_s	4.5 MHz
M	225
N	800
K	6.8×10^{13} Hz/s
L	225

Table 5.2: SAS Parameters

$ITER$	50
Th	0.001
ϕ_{MW}	1°
Sampled angle interval	0.5°
λ_b	50
η	0.0005 (-33 dB)
$U_{\min}$	5



(a) The CDF of $\widehat{\Delta}_R$ ($R = 2$)



(b) Δ_R for all range bins

Figure 5.2: Δ_R calculation

5.4.2 Baseline Algorithm and Metrics

We implement four algorithms: "BPA", "FFT+BPA", "SAS", "FFT+SAS", "RBPA" (BPA with randomly selected phase centers) and "FFT+RBPA" for comparison. The following metrics are used in quantitative evaluations:

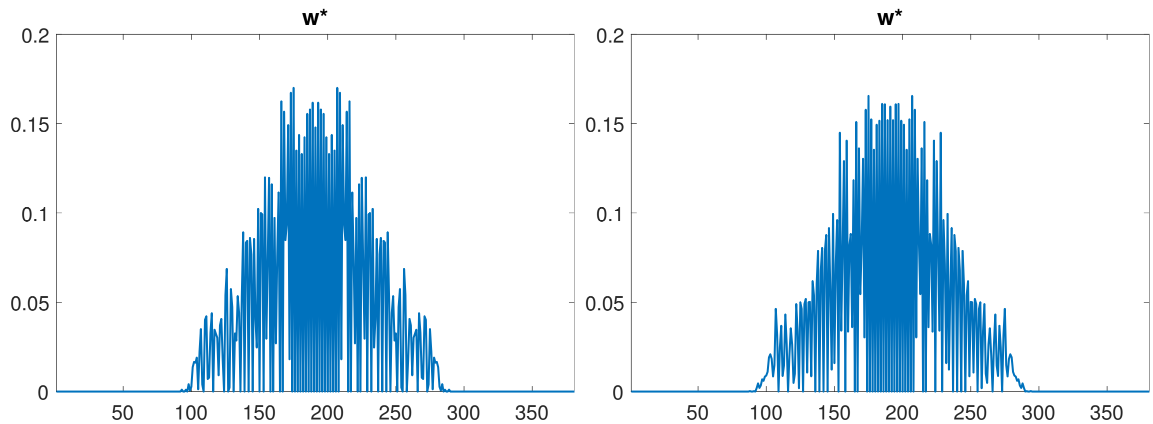
- *Half main lobe width* ϕ_{MW}: The half main lobe width is defined as the angle interval between the peak and the closest local minima on either side of the main lobe.

- *Peak-to-integral sidelobe ratio (PISR)*: The PISR $\mathcal{R}$ for a specific range bin R is calculated as

$$\mathcal{R} = \frac{|I(\phi_m, R)|^2}{\sum_{s=1}^{S} |I(\phi_s, R)|^2}. \tag{5.4.1}$$

- *SAR computation cost*: The elapsed time of generating a SAR image.

- *Image entropy* [69]: Let $E = \sum_\phi \sum_R |I(\phi, R)|^2$ be the total energy of the image, and $d_{(\phi,R)} = \frac{|I(\phi,R)|^2}{E=\sum_\phi \sum_R |I(\phi,R)|^2}$ be the energy density of a pixel. The



(a) Non-robust design (b) Robust design

Figure 5.3: The magnitude of each element in $\mathbf{w}^*$ under different scenarios

image entropy is defined as

$$E_I = -\sum_{\phi,R} d_{(\phi,R)} \ln d_{(\phi,R)}. \tag{5.4.2}$$

The targets are well focused on the SAR image if $\mathcal{R}$ is large and E_I is small.

5.4.3 Numerical Results and Analysis

Solution to the Optimization Problem

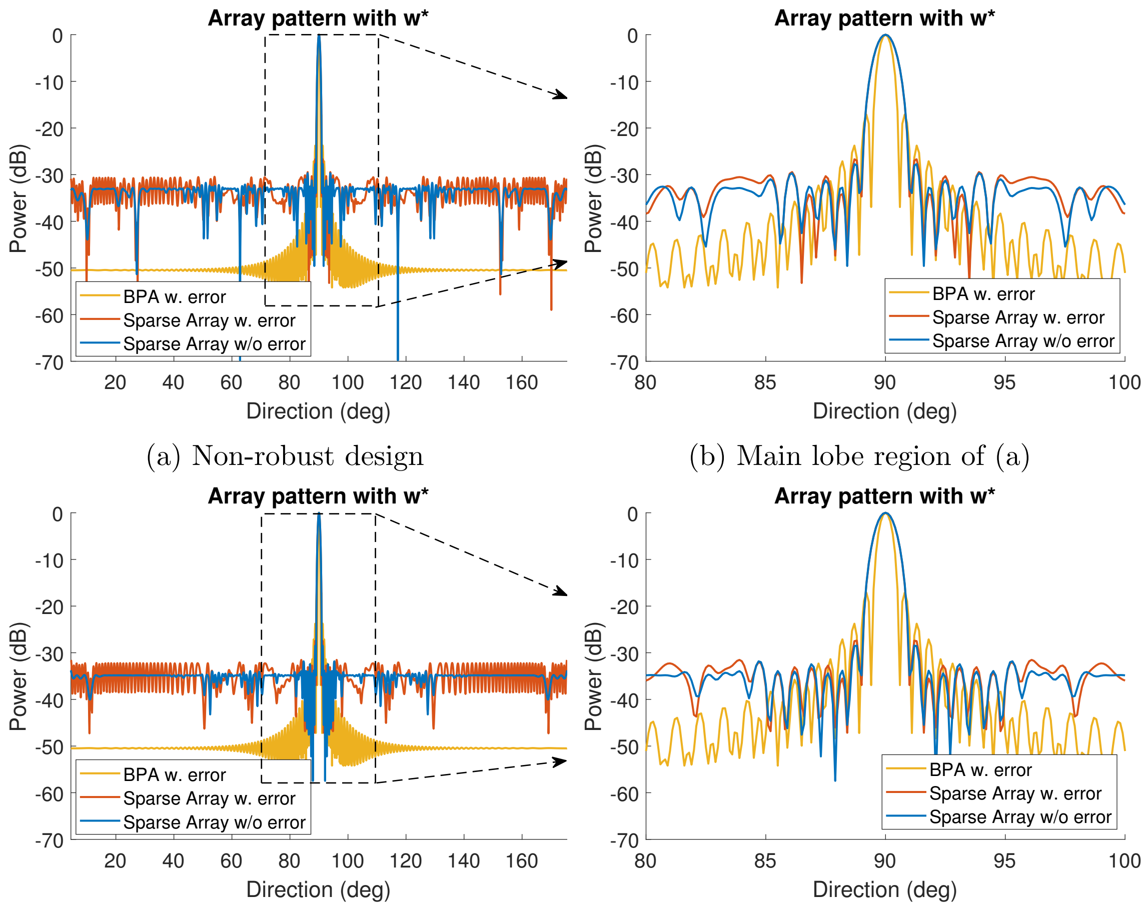
We first give the numerical result through imaging a point target located at $\left(\frac{\pi}{2}, 2m\right)$. In this case, the number of effective phase centers, i.e., the length of synthesized aperture, is calculated to be 381 by (5.2.5). Fig. 5.3 shows the magnitude of each element in $\mathbf{w}^\star$ out and robust design, respectively. In both cases, we can see $\|\mathbf{w}\|_0$ is less than a half of the total number of phase centers from Table 5.3. However, in the robust design, the sparsity is slightly reduced.

Table 5.3: Solution to the Optimization Problem

	$\|\mathbf{w}\|_0$	U'
Non-Robust ($\Delta_R = 0$)	141	8.21
Robust ($\Delta_R = 0.035$)	157	8.38

Array Pattern

Fig. 5.4 shows the array patterns of the SAS and considering robust design for $R = 2$. Figs. 5.4b and 5.4d are the main lobe area of Figs. 5.4a and 5.4c, respectively. The array pattern is calculated by $F(\phi; 2) = |\mathbf{w}^{\star H}\mathbf{a}(\phi; 2)|$ for $\phi = \left[\frac{\pi}{2} - \phi_v, \frac{\pi}{2} + \phi_v\right]$. The blue and red lines show the array pattern and adding phase errors $\mathbf{e}$, respectively. The yellow line is the array pattern of BPA phase errors. As we can see, the main lobe is in $[-1°, 1°]$, which meets the design parameters. As shown in Fig. 5.4a, the power of the sidelobes when there is no error $\mathbf{e}$ in the steering vector is mostly -33dB lower than that of the main lobe peak. The few exceptions fall in between angle grid points of interval 0.5° and thus their power levels are not enforced by the constraints. Although denser grid points (and consequently more constraints) can reduce the chance of requirement violation, the computation cost of SCA grows drastically. When there are errors in the steering vector, most of the sidelobes do not meet the -33dB criteria in non-robust design. In contrast, the robust design (the red line in Fig. 5.4c), this is no longer the case. Furthermore, the average power of the sidelobes is lower than -33dB robust design even in absence of steering vector errors. This is due to the worst case assumption of robustness design as evident in (5.3.5) and (5.3.6). The PISRs are given in Table 5.4. High values are better. The conventional BPA gives the best PISR due to its low sidelobes but needs much more computation time.



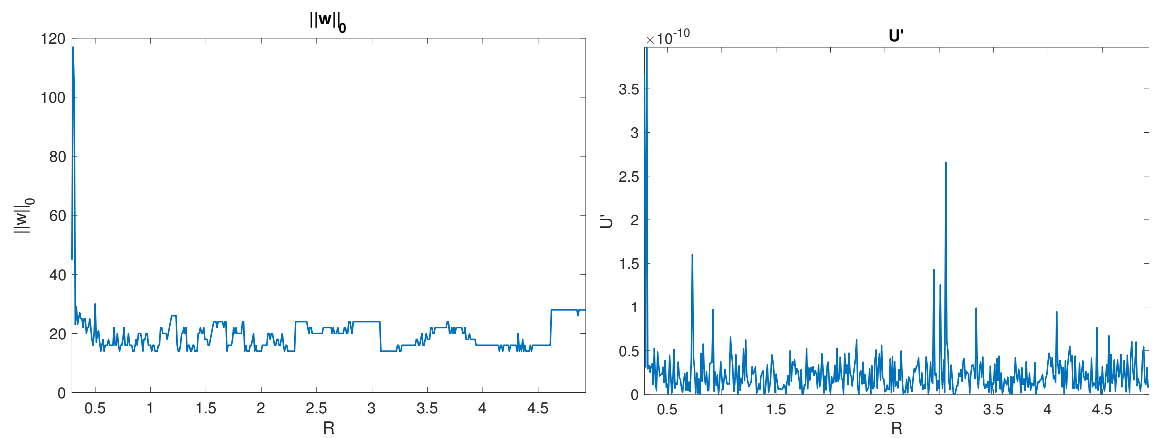
(a) Non-robust design (b) Main lobe region of (a)

(c) Robust design (d) Main lobe region of (c)

Figure 5.4: The Array Pattern in Different Scenarios

Table 5.4: Peak-to-Integral Sidelobe Ratio

Algorithms & Settings	w/o error	w. error
SAS w. Robust	0.1256	0.1253
SAS w/o robust	0.1308	0.1303
BPA w. error	N/A	0.2339



(a) $\|\mathbf{w}\|_0$ (b) U'

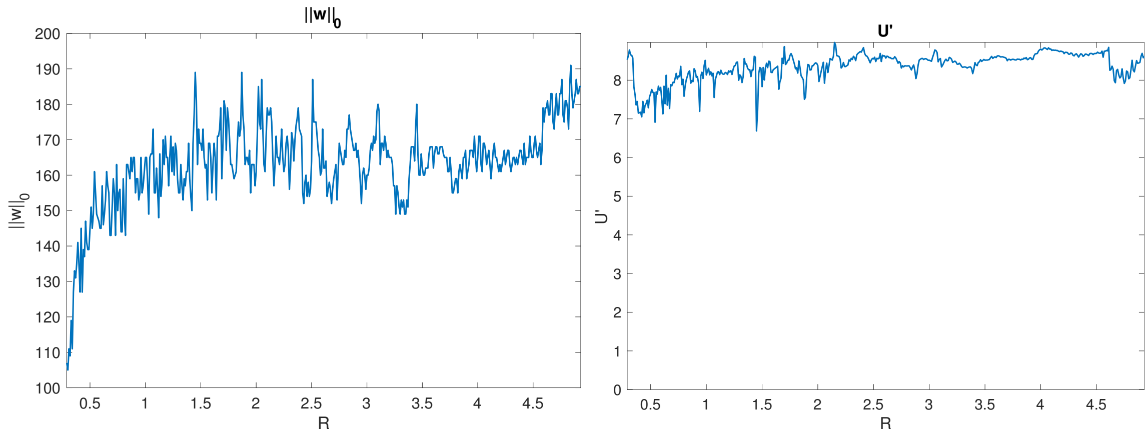
Figure 5.5: The solutions of the SCA $U_{\min} = 0$ for different range bins

Results for All Range Bins

Fig. 5.5 shows the values of $\|\mathbf{w}\|_0$ and U' for all range bins when $U_{\min} = 0$. Clearly, the sparsity of the array holds in all range bins. We observe that all U's are small. In this case, although the array is very sparse, the SNR is low (recall that U' is the magnitude of main-lobe peak and the noise power is constant from C3). Setting $U_{\min} = 5$ can bound the SNR at the cost of reduced sparsity as shown in Fig. 5.6.

5.4.4 SAR Imaging Simulation for a Point Target

By using the MATLAB Phased Array Toolbox, we simulated a point target locating at $\left(\frac{\pi}{2}, 2m\right)$, a rotating radar and the sending/receiving signals of radar antennas. Figs. 5.7a, 5.7c and 5.7e show the imaging results of the target area by conventional BPA, SAS and RBPA, while Figs. 5.7b, 5.7d and 5.7f show the SAR images by employing FFT acceleration. The SAR image quality and computation cost is summarized in Table 5.5. Although the entropy of the SAR image generated by SAS



(a) $\|\mathbf{w}\|_0$ (b) U'

Figure 5.6: The solutions of the SCA $U_{\min} = 5$ for different range bins

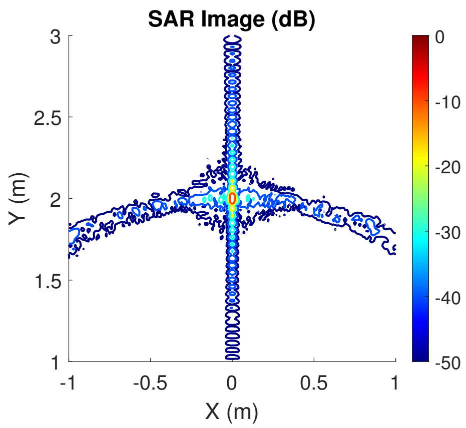
Table 5.5: SAR image quality and time cost

Algorithm	E_I	Time Cost (s)
BPA	**3.7489**	43.66
FFT + BPA	3.9812	29.43
SAS	4.4073	18.03
FFT + SAS	4.4703	**3.61**
RBPA	5.7850	18.03
FFT + RBPA	5.8728	5.86

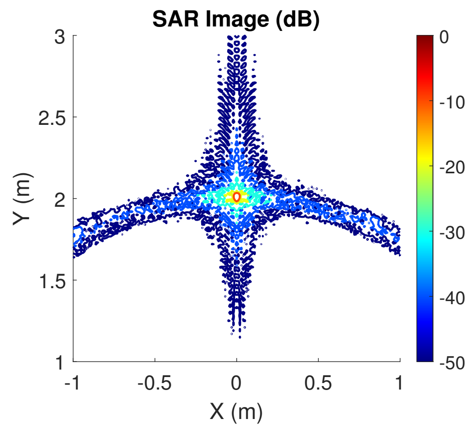
is slightly worse than that by BPA, the computational time is significantly reduced. Furthermore, although BPA with randomly selected phase centers takes less time as well, the resulting image quality is much worse than others.

5.4.5 Testbed Evaluation

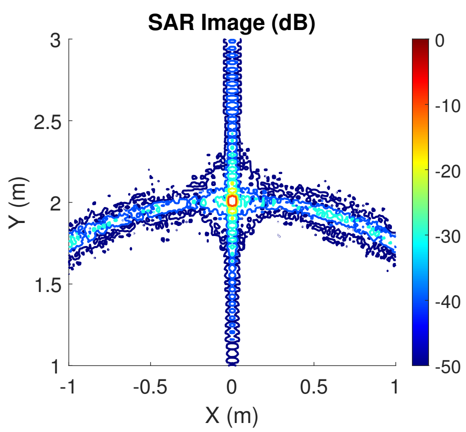
Although simulations can provide insights on the impacts of configuration parameters and the performance of the proposed approach in simulated environments, existing packages in MATLAB cannot model the reflection, diffusion and deflection properties of mmWave signals in indoor environments well. In this section, the performance and



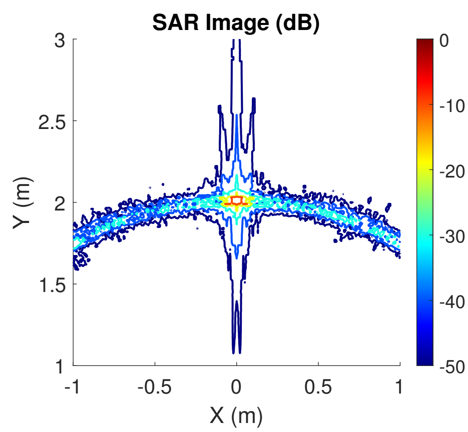
(a) BPA



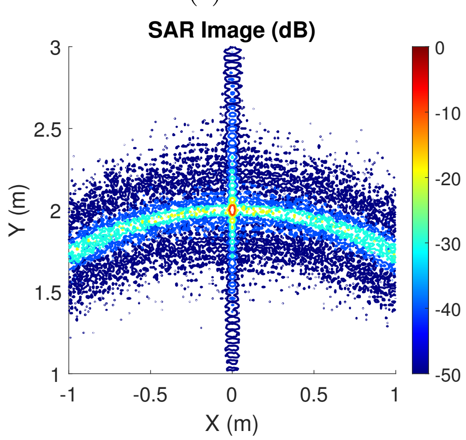
(b) FFT + BPA



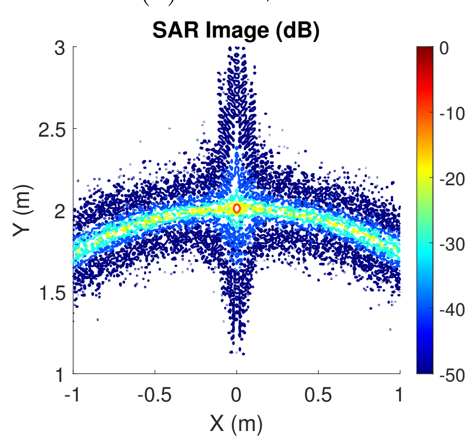
(c) SAS



(d) FFT + SAS



(e) RBPA



(f) FFT + RBPA

Figure 5.7: The SAR Image (Contour) of a point at $\left(\frac{\pi}{2}, 2m\right)$

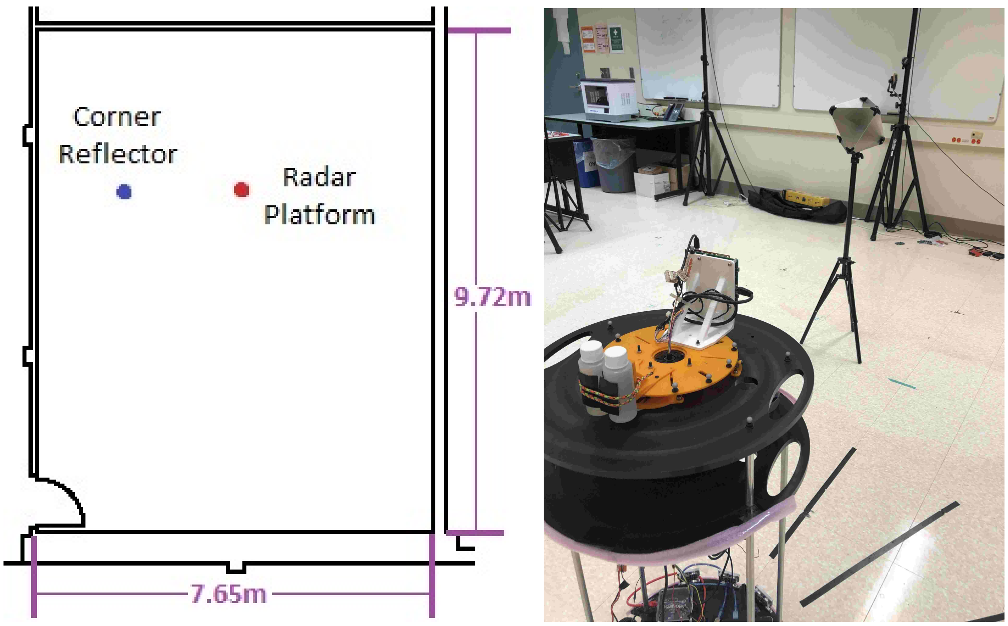
efficiency of the proposed approach is validated through two real-world experiments. The size of SAR area is set to be 9.8756m$\times$9.8756m with grid size being 0.04m$\times$0.04m.

Scenario 1: Corner Reflector

We put a radar platform (red dot) and a corner reflector (blue dot) in a lab (see Fig. 5.8). In addition to a corner reflector which can be treated as the point target with strong reflection in practice, there are also computer desks, wood cabinet, metal cases and other equipment in the environment. Figs. 5.9, 5.10 and 5.11 illustrate the SAR images under different algorithms and settings. The numerical results are summarized in Table 5.6. It can be observed from the figures that BPA gives the clearest image among all approaches. The inclusion of robust design can improve sharpness of the image. At $\phi_{\text{MW}} = 1°$ and $\eta = -33$dB, the image entropy from SAS is comparable to that of the BPA but takes only one fifth of the total computation time. Range-FFT can work in conjunction with both BPA and SAS. When comparing all these figures, we find that range-dimension match filtering degrades image sharpness slightly. This is also corroborated by the image entropy results in Table 5.6. Among all approaches, "FFT+SAS" incurs the least amount of compute time – close to 13 times faster than BPA while achieving acceptable image quality.

Scenario 2: Corridor Corner

Next we collect data from the corner of a corridor. The floor map is given in Fig. 5.12, where the red dot indicates the location of the system, the blue area corresponds to the corridor and the white space with labels represents different rooms. Figs. 5.13, 5.14 and 5.15 show the SAR images from different algorithms and parameter settings.

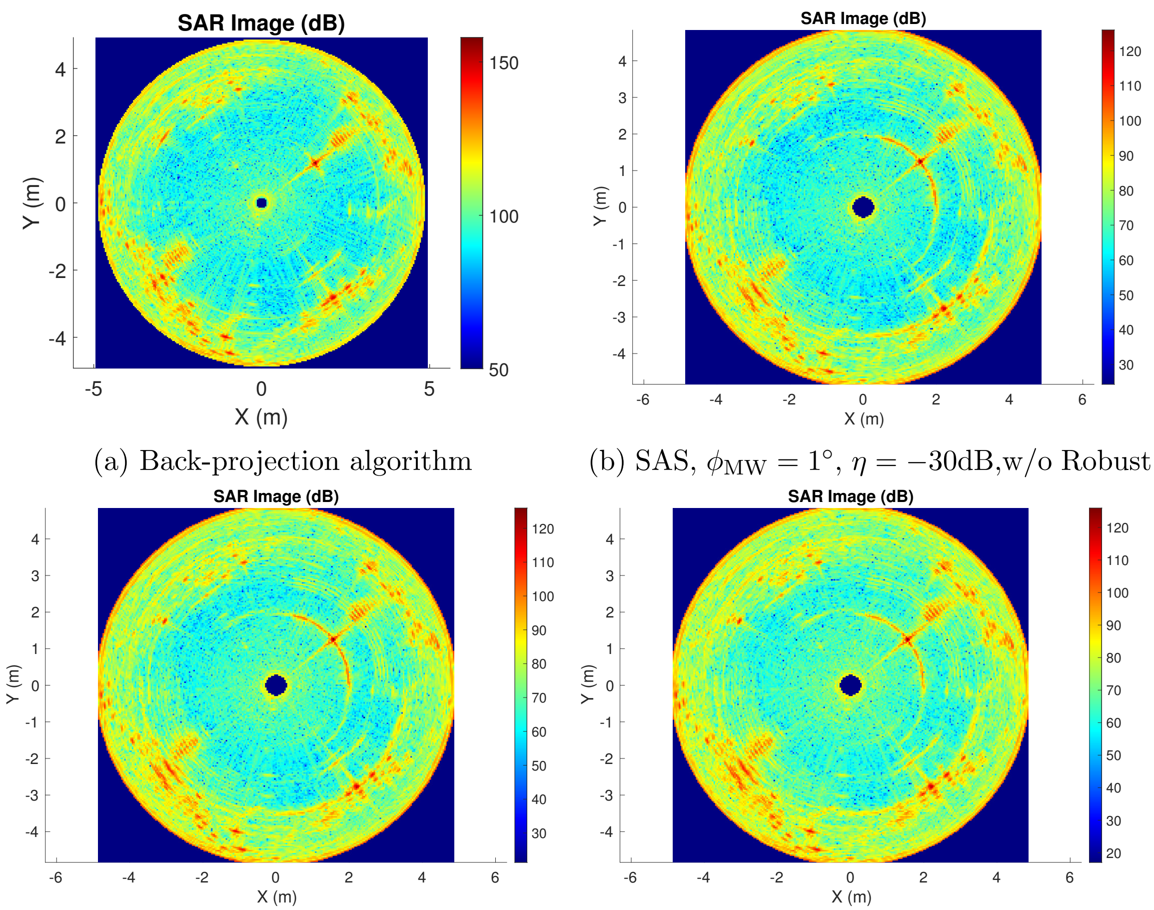


(a) Geometry Relation (b) Experimental Scenario

Figure 5.8: Experiment setup for imaging a corner reflector

Table 5.6: Imaging Performance with Different Approaches in Corner Reflector Case

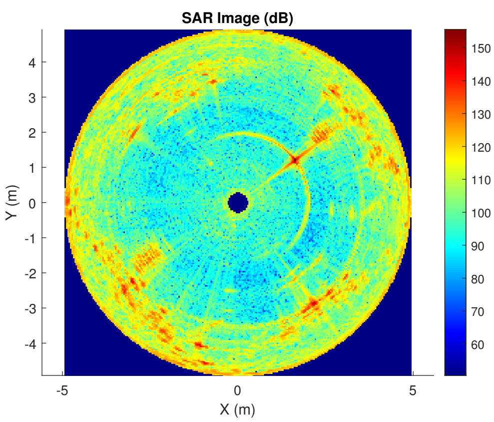
Algorithm	Settings	E_I	Time Cost (s)
BPA	N/A	**6.0472**	151.84
FFT+BPA	N/A	6.3400	21.69
SAS	ϕ_{MW}=1°,η=−30dB,w/o Robust	7.2607	28.04
	$\phi_{\text{MW}} = 1°, \eta = -30$dB,Robust	7.1717	28.94
	$\phi_{\text{MW}} = 1°, \eta = -33$dB,Robust	6.8913	30.56
FFT+SAS	ϕ_{MW}=1°,η=−30dB,w/o Robust	7.3989	**11.45**
	$\phi_{\text{MW}} = 1°, \eta = -30$dB,Robust	7.3227	**12.14**
	$\phi_{\text{MW}} = 1°, \eta = -33$dB,Robust	7.1170	**12.31**
RBPA	N/A	8.3060	25.02
FFT+RBPA	N/A	8.3179	13.07



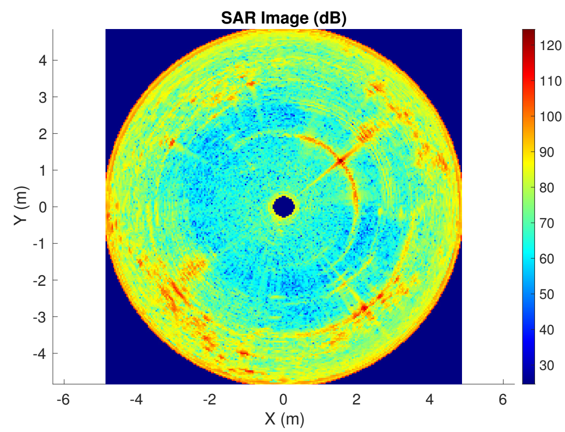
(a) Back-projection algorithm (b) SAS, $\phi_{\mathrm{MW}} = 1^{\circ}$, $\eta = -30$dB,w/o Robust

(c) SAS, $\phi_{\mathrm{MW}} = 1^{\circ}$, $\eta = -30$dB, Robust (d) SAS, $\phi_{\mathrm{MW}} = 1^{\circ}$, $\eta = -30$dB, Robust

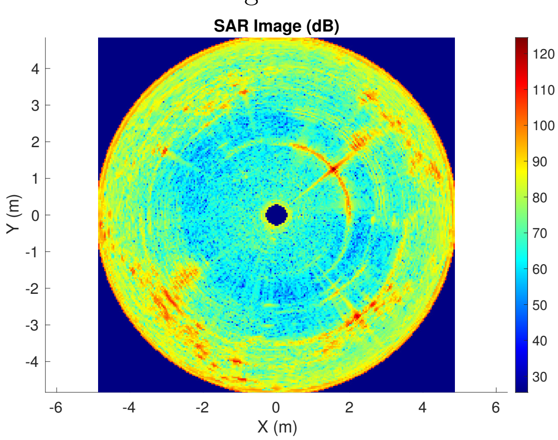
Figure 5.9: SAR Images of a corner reflector (BPA and SAS)



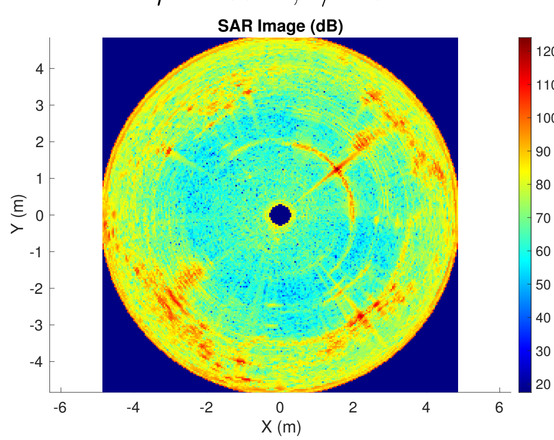
(a) FFT + Back-projection algorithm



(b) FFT + SAS, $\phi_{\mathrm{MW}} = 1°$, $\eta = -30$dB,w/o Robust

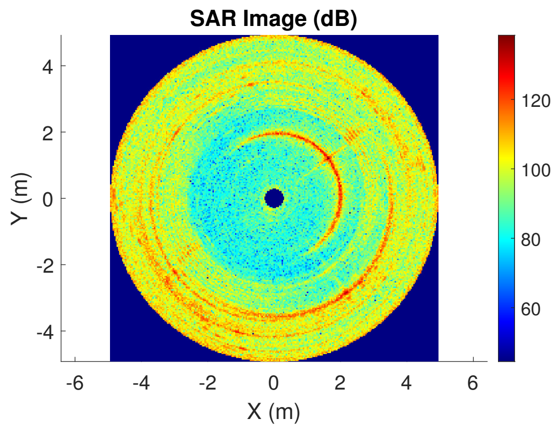


(c) FFT + SAS, $\phi_{\mathrm{MW}} = 1°$, $\eta = -30$dB, Robust

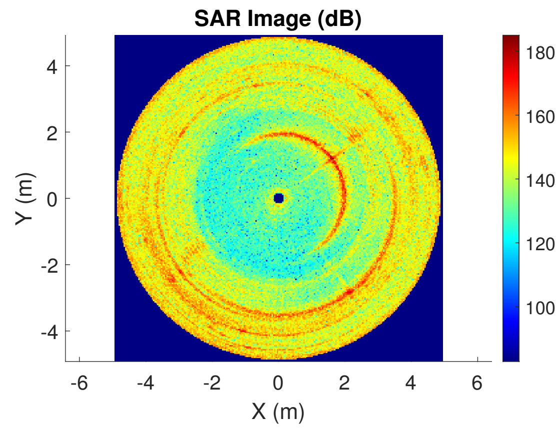


(d) FFT + SAS, $\phi_{\mathrm{MW}} = 1°$, $\eta = -30$dB, Robust

Figure 5.10: SAR images of a corner reflector (FFT + BPA and FFT + SAS)



(a) RBPA



(b) FFT + RBPA

Figure 5.11: SAR images of a corner reflector (RBPA and FFT + RBPA)

The numerical results are summarized in Table 5.7. From the figures, one can discern the outline of the corridor corner. Due to the penetration of mmWave signals through drywalls, the steel bars inside the walls are visible in the figures. Additionally, an object in the Room 128 and the contour of Room 132 are also visible. Similar to the case with corner reflector, the consideration of robust design can indeed improve image quality. When η is set to -33dB and robust design, the sidelobes are less visible. Moreover, inclusion of range-domain FFT can indeed greatly reduce the compute time. BPA with randomly selected phase centers takes less time, but the generated image is blurred. The proposed robust design, with $\phi_{\mathrm{MW}} = 1°$ and $\eta = -33$dB, gives comparable image quality as that of BPA and consume much less computation time.

5.5 Conclusion

In this chapter, we proposed a new fast imaging algorithm based on robust sparse array synthesis for ROSAR. Since radar path is circular, such an algorithm only needs

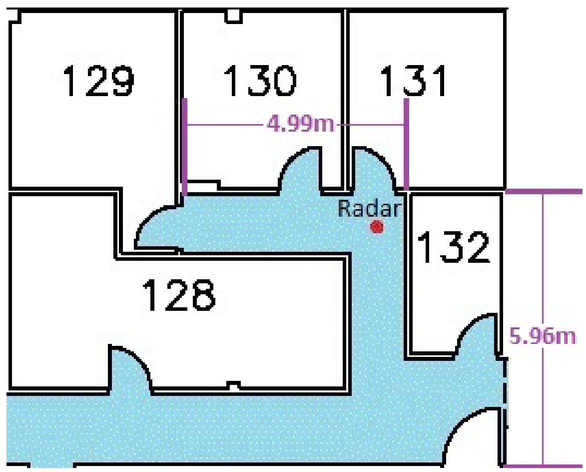
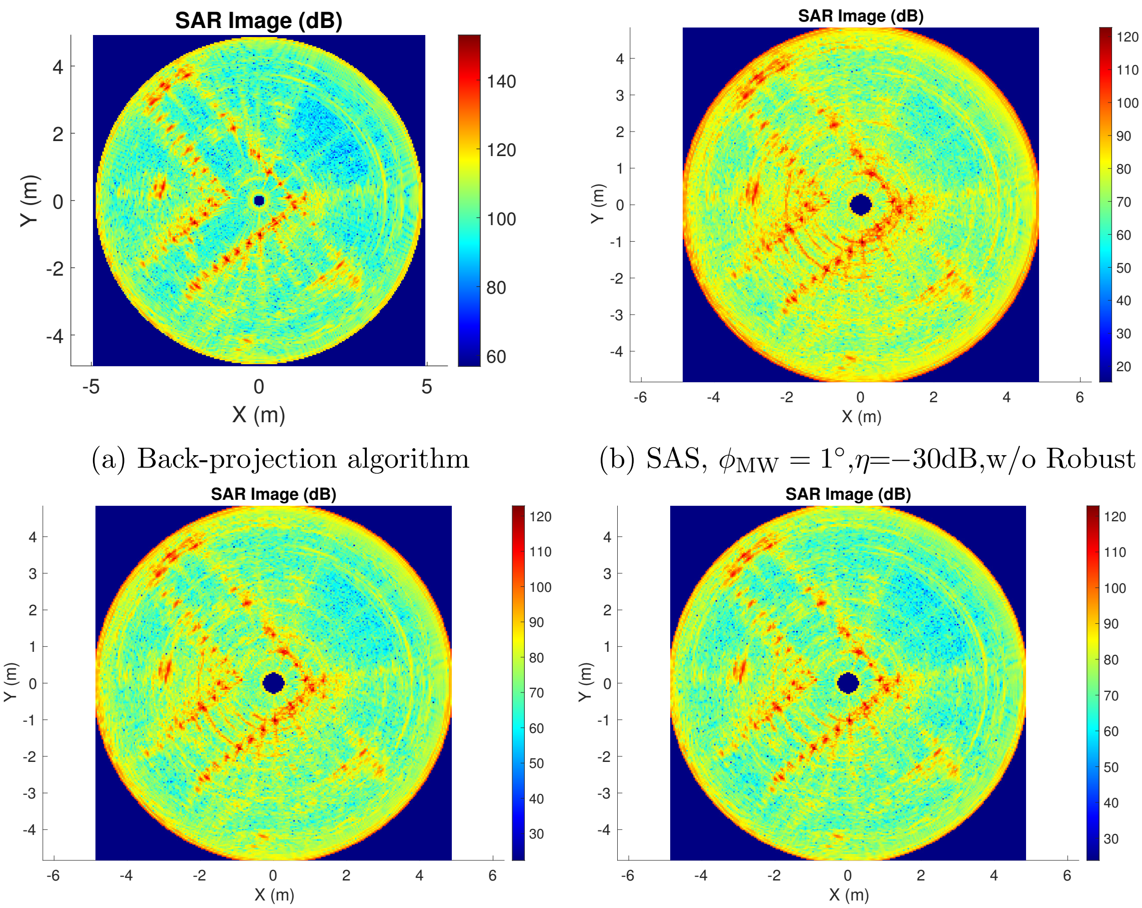


Figure 5.12: Floor map of the corridor corner

Table 5.7: Imaging Performance with Different Approaches in Corridor Corner Case

Algorithm	Settings	E_I	Time Cost (s)
BPA	N/A	**6.4436**	151.22
FFT+BPA	N/A	6.6881	21.85
SAS	ϕ_{MW}=1°,η=−30dB,w/o Robust	7.3802	27.75
	$\phi_{\text{MW}} = 1°$,$\eta = -30$dB,Robust	7.2871	29.30
	$\phi_{\text{MW}} = 1°$,$\eta = -33$dB,Robust	7.0251	30.25
FFT+SAS	ϕ_{MW}=1°,η=−30dB,w/o Robust	7.5309	**12.50**
	$\phi_{\text{MW}} = 1°$,$\eta = -30$dB,Robust	7.4562	**12.21**
	$\phi_{\text{MW}} = 1°$,$\eta = -33$dB,Robust	7.2425	**12.47**
RBPA	N/A	8.3671	25.35
FFT+RBPA	N/A	8.3801	13.28



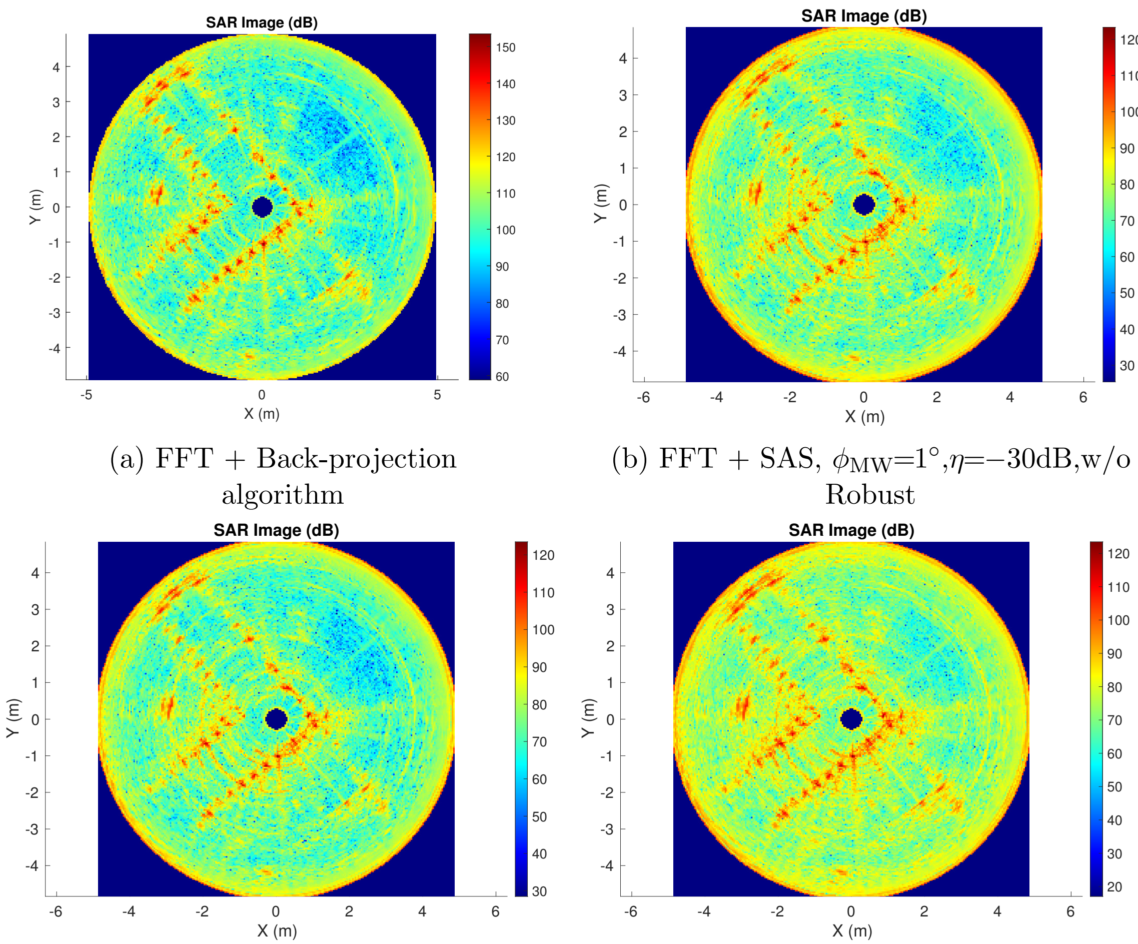
(a) Back-projection algorithm

(b) SAS, $\phi_{\mathrm{MW}} = 1°$,η=−30dB,w/o Robust

(c) SAS, $\phi_{\mathrm{MW}} = 1°$, $\eta = -30$dB, Robust

(d) SAS, $\phi_{\mathrm{MW}} = 1°$, $\eta = -33$dB, Robust

Figure 5.13: SAR images of corridor corner (BPA and SAS)



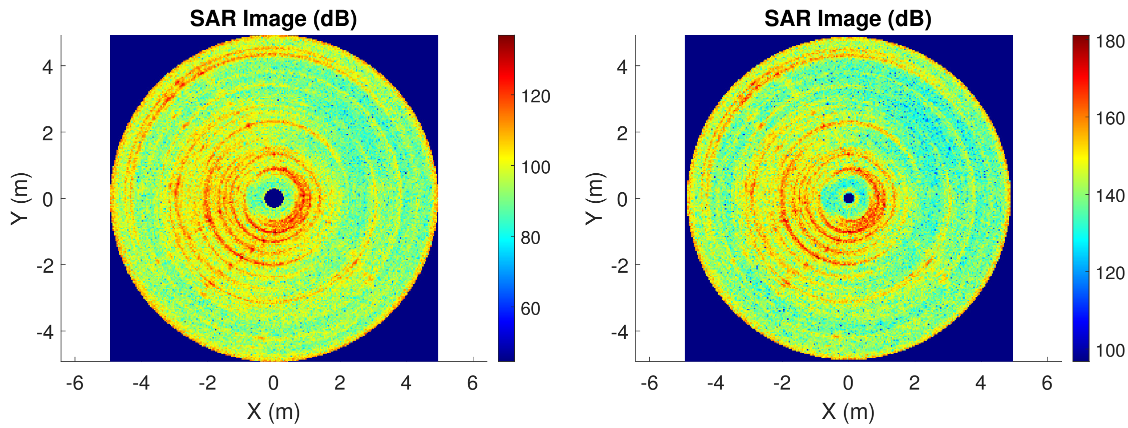
(a) FFT + Back-projection algorithm

(b) FFT + SAS, ϕ_{MW}=1°,η=−30dB,w/o Robust

(c) FFT + SAS, $\phi_{\mathrm{MW}} = 1°$, $\eta = -30$dB, Robust

(d) FFT + SAS, $\phi_{\mathrm{MW}} = 1°$, $\eta = -33$dB, Robust

Figure 5.14: SAR images of corridor corner (FFT + BPA and FFT + SAS)



(a) RBPA (b) FFT + RBPA

Figure 5.15: SAR images of corridor corner (RBPA and FFT + RBPA)

to pre-compute the complex weights of the imaging filter offline for one direction per range bin. Due to the external influence could affect the sidelobe level, we added robust design to maintain the image quality. To meet our pre-set expectation and solve this problem, our proposed algorithm employs feasible point pursuit and successive convex approximation technology. On that basis, we also gave another algorithm based on range-FFT to further reduce the computation complexity.

According to the simulation and testbed results, we can conclude that our approach can generate an SAR image with the quality comparable to that of BPA. Meanwhile, the proposed approach is able to reduce the computational cost significantly and is robust to the array error.

Nonetheless, we must sacrifice some image quality if employing FFT-based range-dimensional matched filtering. Thus, exploring a better approach for the processing based on range FFT is our future research direction.

Chapter 6

ROSAR Autofocusing

6.1 Introduction

Due to their ability to image a 360 view of the surrounding environments, ROSARs find applications in helicopter-borne remote sensing, indoor imaging, etc. As discussed in Chapter 5, applying BPA in ROSAR imaging require precise positions of the virtual antenna elements, which in turns imply either the precise control or measurements of radar movements. However, as profiled in Chapter 4, in practice, the rotation speed of a ROSAR system may not be constant, or the platform holding a ROSAR may not maintain a constant moving speed. In both cases, deviations may be caused by unforeseen disturbances or measurement errors. If the presumed or measured moving parameters are used to derive the locations of phase centers, the final SAR image is blurred. Therefore, autofocusing algorithms are needed to sharpen the images.

There are mainly two kinds of SAR autofocusing algorithms: (1) parametric-based methods, and (2) nonparametric-based methods. The core idea of parametric-based methods is to model the phase errors as a series of polynomials with different orders, and then estimate the coefficient of each term to improve image quality [71]. However, in practice, receiver-side noise and thermal noise make it difficult to estimate the parameters accurately. Nonparametric-based methods, on the other hand, do not require any pre-defined error model. Algorithms such as Dominant Scatter Algorithm (DSA) [59] and Multiple Scatterer Algorithm (MSA) [34] focus on the dominant scatters on an image. If the scatters cannot be identified, Phase Gradient Algorithm (PGA) [24] that can derive arbitrary phase errors through iterative Least-Square estimator is a good alternative. Another category of nonparametric-based methods perform autofocusing based on global image quality, such as Maximum Contrast Algorithm(MCA) [10], General Renyi Entropy based Algorithm (GREA) [49], and Minimum Entropy Algorithm (MEA) [37] by solving a non-convex optimization problem.

In this chapter, we consider the uncertainty in rotation and linear movements of ROSAR systems, and devise autofocusing methods for ROSAR imaging to improve the global quality of SAR images. Unlikely existing MEAs that directly compensate for phase errors independently, we solve for the optimal motion parameters in MEA. We analyze the optimization landscapes of the optimization problems and find that they are highly non-convex and contain many saddle points, local minima and maxima. We compare the solutions based on gradient descent and interior point. Simulation results show that although solutions to MEA cannot yield the actual deviations of the virtual phase centers, they still can successfully reduce the resulting image entropy and to generate sharper images.

The rest of the chapter is organized as follows. Section 6.2 gives two error models for the stationary and moving ROSAR platforms with their corresponding MEA-based problem formulations. Two iterative methods are employed to solve the optimization problem of MEA in Section 6.3, and the effectiveness of the MEA-based autofocusing is validated in Section 6.4. Section 6.5 concludes the chapter.

6.2 System Models

In this section, we introduce the two error models for the stationary and moving ROSAR platforms, and then give the formal problem formulation based on MEA.

6.2.1 Autofocusing for Error Model I: Stationary ROSAR Platform

Figure 6.1 illustrates the virtual phase center position errors due to noisy in rotation speed or measurements in a stationary ROSAR system. When the plate rotates at a constant speed, the radar mounted on the edge of a disk plate transmits chirp signals at a constant rate. However, in practice, the rotation speed may vary due to unexpected time-variant resistance or wheel encoder errors. This means that each virtual phase center may deviate from its assumed location by an angle of $\Delta\phi_n$ for $n = 1, 2, \ldots, N-1$, where N is the number of virtual phase centers. Such deviations affect the distances between the phase centers and the imaged pixels, resulting in a blurred image.

After down-converting and de-ramping, the sampled IF signal received at the n-th

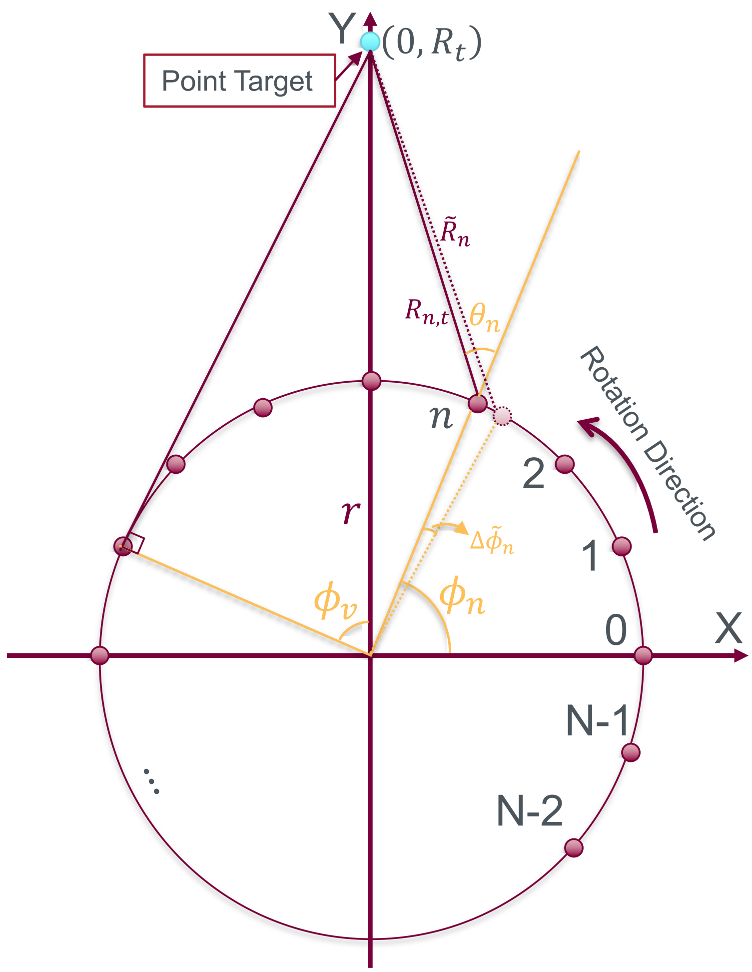


Figure 6.1: Error Model I

phase center is given by

$$\begin{aligned} y_{\mathrm{IF},n}(m) &= \sum_{t=1}^{T} \alpha_{n,t} e^{j2\pi[\tau_{n,t}K(mt_s+T_{Start})+f_c\tau_{n,t}]} + v_{\mathrm{IF}}(n,m) \\ &= \sum_{t=1}^{T} \alpha_{n,t} e^{j2\pi(mKt_s+KT_{Start}+f_c)\frac{2R_{n,t}}{c}} + v_{\mathrm{IF}}(n,m), \end{aligned} \tag{6.2.1}$$

where $\tau_{n,t} = 2R_{n,t}/c$ is the round-trip time delay between the n-th phase center and the t-th target; $R_{n,t} = \sqrt{R_t^2 + r^2 + 2R_t r\cos(\phi_t - \phi_n - \Delta\phi_n)}$; ϕ_t and R_t are the polar coordinates of the t-th target; ϕ_n and $\Delta\phi_n$ denote the presumed direction and the actual direction deviation of the n-th phase center, respectively; r is the radius of the rotation platform; $v_{\mathrm{IF}}(n,m)$ is the sampled noise after deramping; $\alpha_{n,t}$ represents the combination of complex reflection coefficient of the t-th target, radiation pattern of the n-th phase center and the channel fading. For simplicity, we only consider backbaffled omni-directional antenna, whose radiation pattern is given by

$$p(\theta) = \begin{cases} 1 & \theta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \\ 0 & otherwise. \end{cases} . \tag{6.2.2}$$

Due to the complex multipath reflection, wall penetration and the small rotation diameter, we assume that $\alpha_{n,t}$ is only affected by the radiation pattern, namely,

$$\alpha_{n,t} = p\left(\theta_{n,t}\right), \tag{6.2.3}$$

where $\theta_{n,t} = \arctan \frac{R_t \cdot \sin(|\phi_n - \phi_t|)}{R_t \cdot \cos(|\phi_n - \phi_t|) - r}$. Let $k_m = 2\pi\left(mKt_s + KT_{Start} + f_c\right)/c$. We have

$$y_{\mathrm{IF},n}(m) = \sum_{t=1}^{T} \alpha_{n,t} e^{j2k_m R_{n,t}} + v_{\mathrm{IF}}(n,m). \tag{6.2.4}$$

Let $N_{\min}$ and $N_{\max}$ be the smallest and the largest indices where the target is visible in the radar field-of-view (FOV), respectively. With BPA, the intensity of a pixel locating at (x,y) can be written as

$$I(x,y) = \sum_{n=N_{\min}}^{N_{\max}} \sum_{m=0}^{M-1} y_{\mathrm{IF},n}(m) e^{-j2k_m \widetilde{R}_n}, \tag{6.2.5}$$

where $\widetilde{R}_n = \sqrt{R^2 + r^2 + 2Rr\cos\left(\phi - \phi_n - \Delta\widetilde{\phi}_n\right)}$; ϕ and R are the polar coordinates of a pixel locating at (x, y) on the image, $\Delta\widetilde{\phi}_n$ is the estimated direction deviation of the n-th phase center. Since the actual direction deviation $\Delta\phi_n$ is not known a priori, assuming $\Delta\widetilde{\phi}_n$ is 0 would lead to a blurred image. The image entropy is calculated as

$$\begin{aligned} E_g &= -\sum_x \sum_y \frac{|I(x,y)|^2}{E} \ln \frac{|I(x,y)|^2}{E} \\ &= \ln E - \frac{E_2}{E}, \end{aligned} \tag{6.2.6}$$

where $E = \sum_x \sum_y |I(x,y)|^2$ and $E_2 = \sum_x \sum_y |I(x,y)|^2 \ln |I(x,y)|^2$. The goal of MEA is represented as

$$\Delta\widetilde{\phi}^\star = \arg\min_{\Delta\widetilde{\phi}} E_g, \tag{6.2.7}$$

where $\Delta\widetilde{\phi} = \left\{\Delta\widetilde{\phi}_n | n = 0, 1, \ldots, N-1\right\}$.

6.2.2 Autofocusing for Error Model II: Moving ROSAR Platform

In the second error model, a rover is assumed to move in a straight line at a contant speed v with a radar that rotates at a fixed angular speed (Fig. 6.2). Because of the rotational symmetry, we always set the heading direction to be the positive direction of X-axis. The radar's start capturing direction, i.e., the direction of the 0-th virtual phase center, is set to be 0°. Due to the imperfect control, the rover linear speed and start capturing direction may deviate from their predefined values. Let Δv, $\Delta\psi$, t_f be the speed deviation of the rover, start capturing direction deviation and

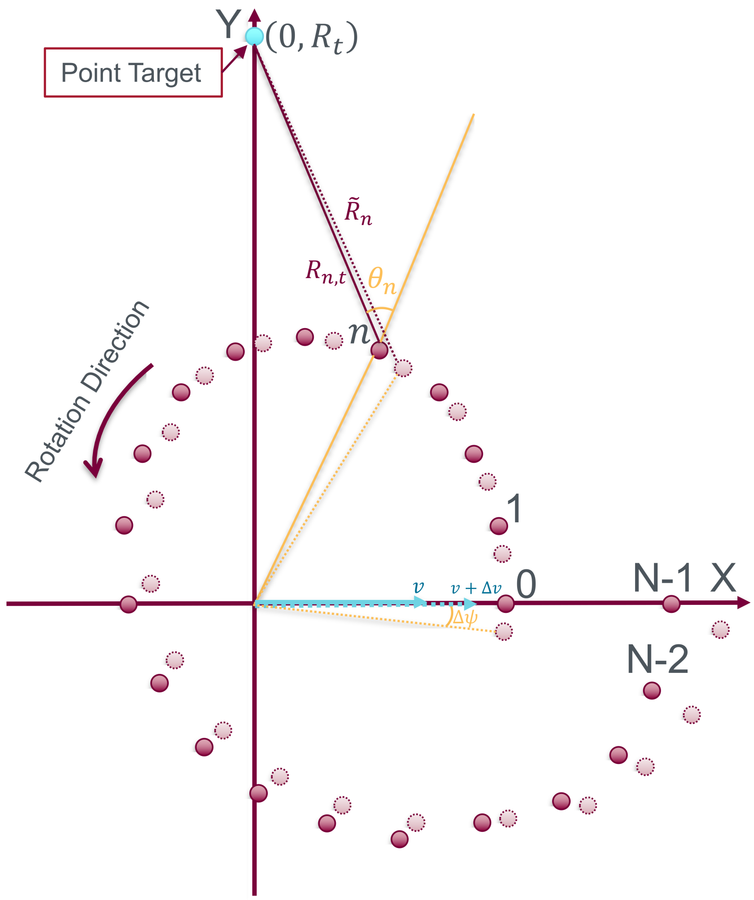


Figure 6.2: Error Model II

pulse repetition interval (also called "slow time"). For the n-th phase center, the corresponding actual rover location is $[(v + \Delta v)nt_f, 0]$ and its relative location w.r.t. the rover is $r \cdot \left[\cos\left(\frac{2\pi n}{N} + \Delta\psi\right), \sin\left(\frac{2\pi n}{N} + \Delta\psi\right)\right]$. Then, the absolute location of the n-th phase center is represented as

$$\left[(v + \Delta v)nt_f + r\cos\left(\frac{2\pi n}{N} + \Delta\psi\right), r\sin\left(\frac{2\pi n}{N} + \Delta\psi\right)\right]. \tag{6.2.8}$$

After down-converting and de-ramping, the sampled IF signal received at the n-th phase center is given by

$$y_{\mathrm{IF},n}(m) = \sum_{t=1}^{T} \alpha_{n,t} e^{j2\pi(mKt_s + KT_{Start} + f_c)\frac{2R_{n,t}}{c}} + v_{\mathrm{IF}}(n,m), \tag{6.2.9}$$

where the distance between the n-th phase center and t-th target is calculated as

$$R_{n,t} = \sqrt{\begin{aligned}&\left[(v+\Delta v)nt_f + r\cos\left(\frac{2\pi n}{N}+\Delta\psi\right) - R_t\cos(\phi_t)\right]^2\\ &+\left[r\sin\left(\frac{2\pi n}{N}+\Delta\psi\right) - R_t\sin(\phi_t)\right]^2\end{aligned}}. \tag{6.2.10}$$

By applying BPA, the intensity of a pixel locating at (x, y) can be formulated as

$$I(x,y) = \sum_{n=N_{\min}}^{N_{\max}} \sum_{m=0}^{M-1} y_{\mathrm{IF},n}(m) e^{-j2k_m \widetilde{R}_n}, \tag{6.2.11}$$

where the estimated distance $\widetilde{R}_n$ is defined as

$$\widetilde{R}_n = \sqrt{\left[(v+\Delta\widetilde{v})\,nt_f + r\cos\left(\frac{2\pi n}{N}+\Delta\widetilde{\psi}\right) - x\right]^2 + \left[r\sin\left(\frac{2\pi n}{N}+\Delta\widetilde{\psi}\right) - y\right]^2}, \tag{6.2.12}$$

$\Delta\widetilde{v}$ is the estimated speed deviation and $\Delta\widetilde{\psi}$ is the estimated start capturing direction deviation. Similarly, to get a sharper image, we choose MEA to optimize the entropy over these two variables. Thus, the optimization problem is stated as

$$\Delta\widetilde{v}^{\star}, \Delta\widetilde{\psi}^{\star} = \arg\min_{\Delta\widetilde{v},\Delta\widetilde{\psi}} E_g, \tag{6.2.13}$$

where E_g is defined in Eq. (6.2.6).

6.3 Solution Approach and Implementation

Due to the high nonlinearity of E_g, it is challenging to obtain the global optimal values corresponding to the minimum E_g. Thus, we consider two iterative methods, i.e., gradient descent and interior point. Both methods need the gradient information of the objective function.

For Error Model I, we only need to calculate the first derivative for any one of the N variables since they all have the same form. Thus, taking the first derivative to E_g with respect to an optimization variable $\Delta\widetilde{\phi}_{\widehat{n}}$, we have

$$\begin{aligned}\frac{\partial E_g}{\partial \Delta\widetilde{\phi}_{\widehat{n}}} &= \frac{1}{E}\frac{\partial E}{\partial \Delta\widetilde{\phi}_{\widehat{n}}} - \frac{1}{E}\frac{\partial E_2}{\partial \Delta\widetilde{\phi}_{\widehat{n}}} + \frac{E_2}{E^2}\frac{\partial E}{\partial \Delta\widetilde{\phi}_{\widehat{n}}} \\ &= \sum_x\sum_y\left(\frac{E+E_2}{E^2} - \frac{\ln|I(x,y)|^2}{E}\right)2Re\left\{I^*(x,y)\frac{\partial I(x,y)}{\partial \Delta\widetilde{\phi}_{\widehat{n}}}\right\}, \quad (6.3.1)\end{aligned}$$

and

$$\frac{\partial I(x,y)}{\partial \Delta\widetilde{\phi}_{\widehat{n}}} = j\sum_{m=0}^{M-1} y_{\mathrm{IF},\widehat{n}}(m)e^{-j2k_m\widetilde{R}_{\widehat{n}}}\frac{2k_mRr\sin\left(\phi-\phi_{\widehat{n}}-\Delta\widetilde{\phi}_{\widehat{n}}\right)}{\widetilde{R}_{\widehat{n}}}. \quad (6.3.2)$$

For Error Model II, we must calculate the first derivative of E_g with respect to both $\Delta\widetilde{v}$ and $\Delta\widetilde{\psi}$. The results are given by

$$\begin{aligned}\frac{\partial E_g}{\partial \Delta\widetilde{v}} &= \frac{1}{E}\frac{\partial E}{\partial \Delta\widetilde{v}} - \frac{1}{E}\frac{\partial E_2}{\partial \Delta\widetilde{v}} + \frac{E_2}{E^2}\frac{\partial E}{\partial \Delta\widetilde{v}} \\ &= \sum_x\sum_y\left(\frac{E+E_2}{E^2} - \frac{\ln|I(x,y)|^2}{E}\right)2Re\left\{I^*(x,y)\frac{\partial I(x,y)}{\partial \Delta\widetilde{v}}\right\}, \quad (6.3.3)\end{aligned}$$

$$
\begin{aligned}
\frac{\partial E_g}{\partial \Delta\widetilde{\psi}} &= \frac{1}{E}\frac{\partial E}{\partial \Delta\widetilde{\psi}} - \frac{1}{E}\frac{\partial E_2}{\partial \Delta\widetilde{\psi}} + \frac{E_2}{E^2}\frac{\partial E}{\partial \Delta\widetilde{\psi}} \\
&= \sum_x \sum_y \left(\frac{E+E_2}{E^2} - \frac{\ln |I(x,y)|^2}{E} \right) 2Re\left\{ I^*(x,y) \frac{\partial I(x,y)}{\partial \Delta\widetilde{\psi}} \right\}. \quad (6.3.4)
\end{aligned}
$$

The detailed derivation is given in Appendix B.

Gradient Descent Method Gradient descent is an iterative method that repeatedly moves the current point along the opposite direction of its gradient. Gradually, a local minimum value is founded. For Error Model I, the key iteration step for each optimization variable is given by

$$
\Delta\widetilde{\phi}_n^{(l+1)} = \Delta\widetilde{\phi}_n^{(l)} - \gamma \frac{\partial E_g}{\partial \Delta\widetilde{\phi}_n}\bigg|_{\Delta\widetilde{\phi}_n = \Delta\widetilde{\phi}_n^{(l)}}, \quad (6.3.5)
$$

where γ is the learning rate. Similarly, for Error Model II, we have

$$
\Delta\widetilde{v}^{(l+1)} = \Delta\widetilde{v}^{(l)} - \gamma \frac{\partial E_g}{\partial \Delta\widetilde{v}}\bigg|_{\Delta\widetilde{v} = \Delta\widetilde{v}^{(l)}}, \quad (6.3.6)
$$

$$
\Delta\widetilde{\psi}^{(l+1)} = \Delta\widetilde{\psi}^{(l)} - \gamma \frac{\partial E_g}{\partial \Delta\widetilde{\psi}}\bigg|_{\Delta\widetilde{\psi} = \Delta\widetilde{\psi}^{(l)}} \quad (6.3.7)
$$

Interior-Point Method Interior-point is an iterative method that searching an optimum value by traversing the feasible region given an initial feasible point. This method is usually more efficient than the simplex method and ellipsoid method. In MATLAB, “fmincon” is a powerful tool to find the minimum value of constrained nonlinear multivariable function. It uses interior-point as the optimization algorithm by default. In practice, “fmincon” can estimate the gradient automatically, but it can be very time consuming due to the high computation complexity of BPA. Therefore, we specify the gradients to accelerate the computation.

6.4 Performance Evaluation

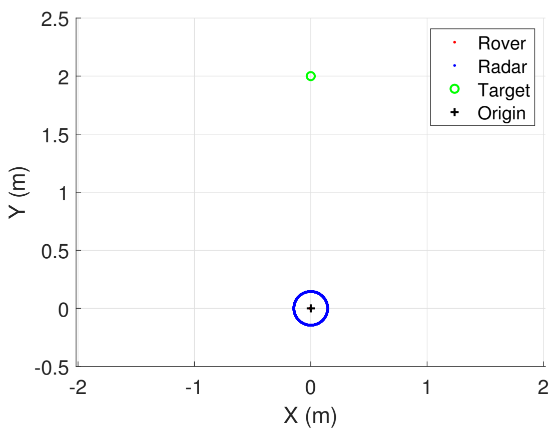
To simulate a ROSAR platform, we implement the radar antenna, the target platform and radar signal transmissions using the MATLAB Phased Array Toolbox. We use the Sensor Fusion and Tracking Toolbox to implement the movement of the radar antenna and the rover. These toolboxes allow us to obtain the IF signals collected by each virtual phase center and the location, velocity, and orientation of the radar antenna and the rover. The simulated parameters are set to be the same as the one in the hardware platform as summarized in Table 6.1.

We evaluate the performance of the autofocusing algorithms in four scenarios with one, two, four and five point targets. All the targets are stationary and have the same backscatter coefficient in all directions. The coordinates of the targets in each scenario are listed below and illustrated in Fig. 6.3.

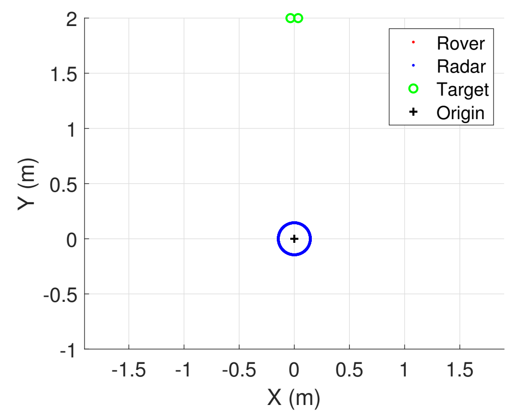
- 1 target: $(0, 2)$.
- 2 targets: $(0.0349, 1.9997)$, $(-0.0349, 1.9997)$.
- 4 targets: $(0, 2)$, $(2, 0)$, $(-2, 0)$, $(0, -2)$.
- 5 targets: $(0.145, 1)$, $(0.145, 1.5)$, $(0.545, 0.5)$, $(0.545, 1)$, $(0.545, 1.5)$.

Table 6.1: Simulation Settings

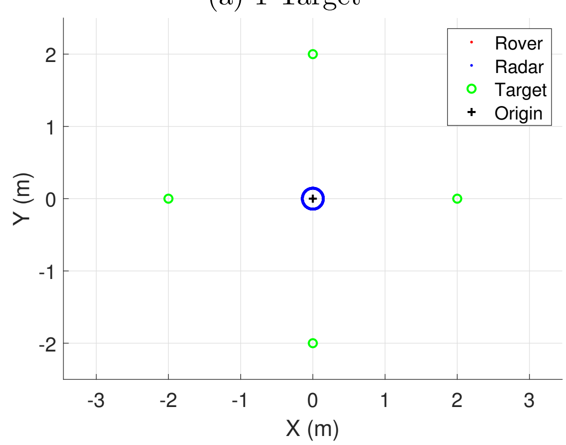
Radar Settings	
r	0.145 m
Rotation speed	60 RPM
Rotation time	1 s
# Of TX	1
# Of RX	1
Antenna pattern	Omni-directional
Antenna FOV	$[-90°, 90°]$
Start rotating direction	Same as rover heading direction
Rotation direction	Counter-clockwise
TX power	12 dBm
Antenna gain	7 dBi
RX gain	48 dB
RX noise power	0 dB
Chirp Signal Settings	
Start frequency	60 GHz
End frequency	64 GHz
Ramp start time	0 us
Ramp end time	58 us
T_{Start}	7 us
Sampling end time	57 us
F_s	4.5 MHz
M	225
N	800
t_s	1/800 s
K	6.8×10^{13} Hz/s
SAR Image Settings	
Image length	6.88 m
Image width	6.88 m
Grid length	0.01 m



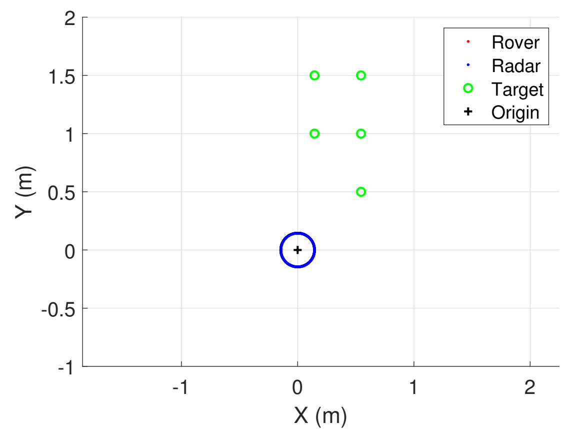
(a) 1 Target



(b) 2 Targets



(c) 4 Targets



(d) 5 Targets

Figure 6.3: Geometrical Relationship

6.4.1 Numerical Results and Analysis

Autofocusing for a Stationary ROSAR

In the experiments, the direction deviation of each virtual phase center is set to follow a uniform distribution between $[-0.225°, 0.225°]$. Fig. 6.4 shows the autofocusing results for the 1-target case. Fig. 6.4a displays the image entropies in each iteration using MEA with either gradient descent or interior-point method. Fig. 6.4b presents the image entropies based on different value combinations of $\Delta\widetilde{\phi}_{34}$ and $\Delta\widetilde{\phi}_{68}$ around the ground truth (the red dot) while setting remaining control values to their respective ground truth values. Fig. 6.4c and 6.4d compare the phase center deviations given by both methods, where the red line represents the ground truth deviations and the green line represents the estimated deviations. Figs. 6.4e, 6.4f, 6.4g and 6.4h show the SAR images in dB scale based on the ground truth, initial, MEA estimated (gradient descent and interior-point) deviations of each phase center, respectively. We also conduct the same experiments for the 2-target, 4-target and 5-target cases and present the results in Figs. 6.5, 6.6 and 6.7. Table 6.2 summarizes the numerical results for all scenarios. From these results, we have the following findings:

- From the optimization landscape figures (Figs. 6.4b, 6.5b, 6.6b and 6.7b), we can observe the existence of periodic local maxima and minima, with saddle points appearing between them. Furthermore, the ground truth location is not always situated at a local minimum.

- The initial image entropy may be lower than that of the ground truth image. This is because the ground truth may not have the global minimum entropy. In fact, setting all the estimated deviations to zero could blur the SAR image,

Table 6.2: Numerical Results for Error Model I

Case	Ground Truth	Initial	MEA (Gradient Descent)	MEA (Interior-Point)
1 Target	6.7526	6.3586	4.9786	4.8609
2 Targets	7.0692	7.0378	5.6060	5.5509
4 Targets	7.6686	7.5729	6.6167	6.6127
5 Targets	7.9033	7.5494	5.4579	5.4159

but not necessarily increase the entropy.

- Both gradient descent and interior-point methods can gradually decrease the image entropy and sharpen the images from the initial value settings. However, these methods usually restrict the searching range to the adjacent local minimum area, which may not contain the global minimum entropy or the ground truth.

- By using the final estimated deviations from both methods, we can obtain sharpened images, although the estimated deviations may not be consistent with the ground truth.

Based on the overall results in this error model, MEA can significantly reduce the image entropies and sharpen the images.

Autofocusing for a Moving ROSAR

In this experiment, the rover's moving speed and deviations of the ROSAR system are set in Table 6.3. Table 6.4 compares the SAR image entropies based on phase center locations according to the ground truth, initial, MEA estimated (gradient descent and interior-point) deviations of each phase center, respectively. Table 6.5 and 6.6 show the Δv and $\Delta\psi$ estimations. Fig. 6.8 shows the autofocusing results for the

1-target case. Fig. 6.8a, 6.8b and 6.8c displays the image entropies, the estimated Δv and the estimated $\Delta\psi$ in each iteration using MEA with either gradient descent or interior-point method. We also conduct the same experiments for the 2-target, 4-target and 5-target cases and present the results in Fig. 6.9, 6.10 and 6.11. From these figures, we can see the algorithm usually stops in a few iterations and the image entropy improvement is very small. In addition, the estimated Δv and $\Delta\psi$ are far from the ground truth ones.

Similar to Error Model I, we aim to understand the optimization landscape. To achieve this, we analyze image entropies based on various combinations of $\Delta\widetilde{v}$ and $\Delta\widetilde{\psi}$ around the ground truth (as depicted in 6.8d, 6.9d, 6.10d and 6.11d). The red dot represents the ground truth deviation values, while the initial optimization variables are consistently set to zero. Upon examining these figures, we observe that the difference between local maxima and local minima are relatively small. This phenomenon explains why the iteration process terminates quickly. Additionally, we've identified multiple local minima between the initial value and the ground truth. Algorithms that exclusively seek a descent direction, such as gradient descent, may be trapped in nearby local minima and struggle to converge to the global optimum or ground truth. In the context of Error Model II, MEA can marginally reduce image entropy but still rarely achieves ground truth deviations.

Table 6.3: Parameter Settings

v	0.05 m/s
Δv	0.01 m/s
$\Delta\psi$	$-5°$

Table 6.4: Numerical Results for Error Model II

Case	Ground Truth	Initial	MEA (Gradient Descent)	MEA (Interior-Point)
1 Target	5.4923	5.4590	5.4583	5.4555
2 Targets	7.0848	6.8370	6.8362	6.8100
4 Targets	6.7754	6.9143	6.8664	6.8266
5 Targets	6.6030	5.9612	5.9231	5.8465

Table 6.5: Δv Estimations

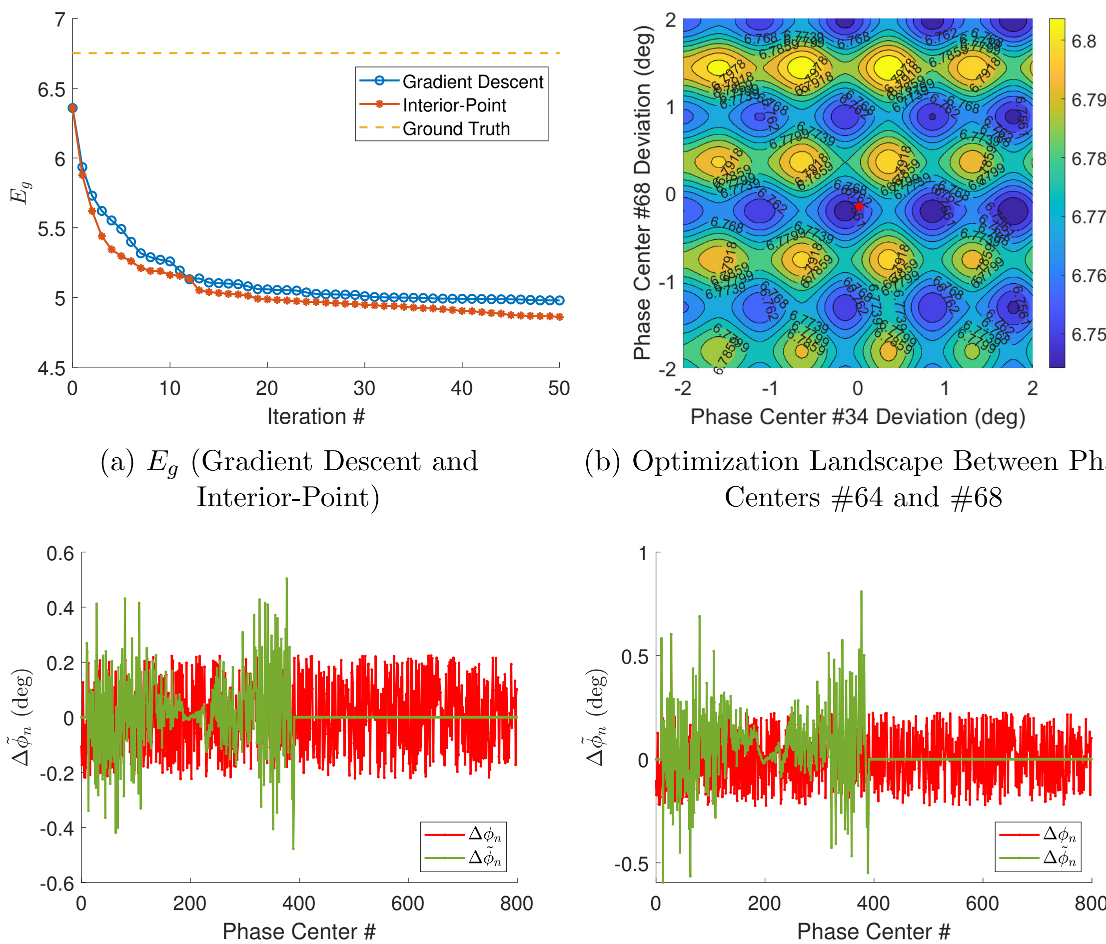
Case	Ground Truth	Initial	MEA (Gradient Descent)	MEA (Interior-Point)
1 Target	0.01	0	0.00007	-0.0005
2 Targets			0.00002	-0.0033
4 Targets			0.0008	0.0121
5 Targets			0.0009	0.0106

6.5 Conclusion

In this chapter, we investigated the problem of ROSAR image autofocusing. Motivated by the measurement study of the hardware ROSAR system in Chapter 4, we proposed two error models to characterize phase center deviations of the stationary and moving platforms. Among nonparametric-based autofocusing algorithms, we considered MEA to reduce the image entropy and to get sharp SAR images. Gradient descent and interior point methods were employed to solve the corresponding optimization problem of MEA. Simulation results show that both solutions can successfully generate sharper images although they cannot be used to estimate the ground truth deviations. This is because the ground truth deviations may not necessarily locate at the global minima or a local minima on the optimization landscape, and only iterative reducing the image entropy is not enough to achieve the desirable values. Furthermore, due to the complex optimization landscapes, both iterative methods tend to converge to a sub-optimal solution.

Table 6.6: $\Delta\psi$ Estimations

Case	Ground Truth	Initial	MEA (Gradient Descent)	MEA (Interior-Point)
1 Target	-5	0	-0.0040	-0.0274
2 Targets			0.0021	0.3783
4 Targets			0.0482	0.2407
5 Targets			0.0506	0.5958



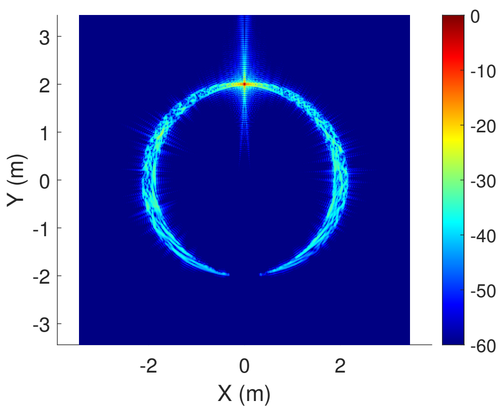
(a) E_g (Gradient Descent and Interior-Point)

(b) Optimization Landscape Between Phase Centers #64 and #68

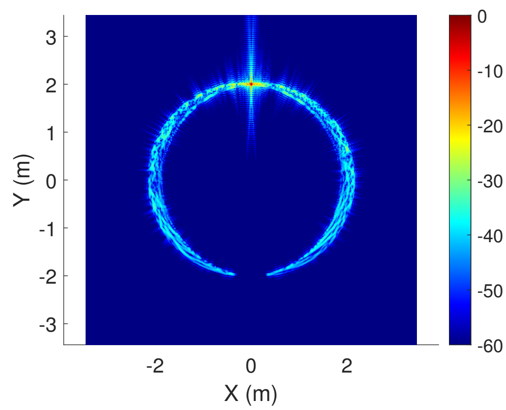
(c) $\Delta\phi_n$ and $\Delta\widetilde{\phi}_n$ (Gradient Descent)

(d) $\Delta\phi_n$ and $\Delta\widetilde{\phi}_n$ (Interior-Point)

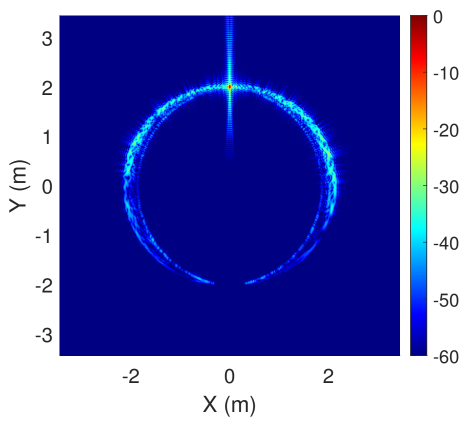
Figure 6.4: Numerical Results for 1-Target Case (Error Model I) (More sub-figures on the next page)



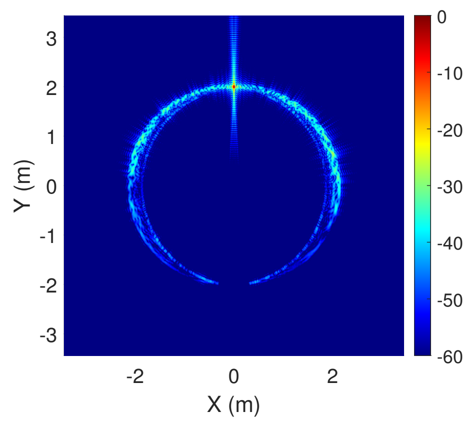
(e) Ground Truth SAR Image
$E_g = 6.7526$



(f) Initial SAR Image
$E_g = 6.3586$

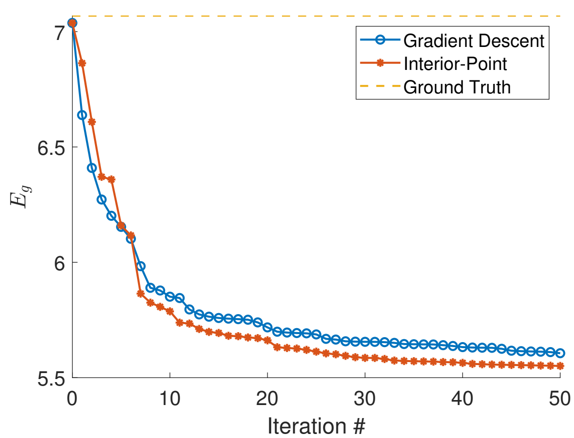


(g) Autofocused Image (Gradient Descent)
$E_g = 4.9786$

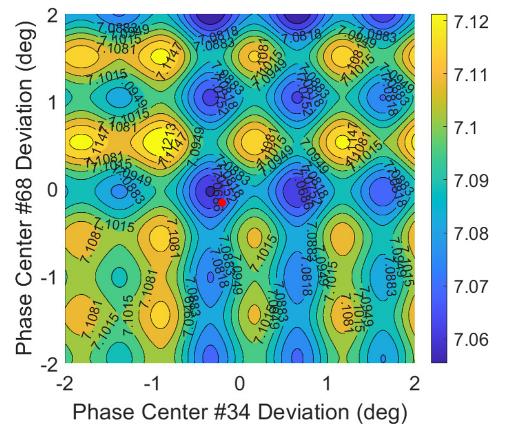


(h) Autofocused Image (Interior-Point)
$E_g = 4.8609$

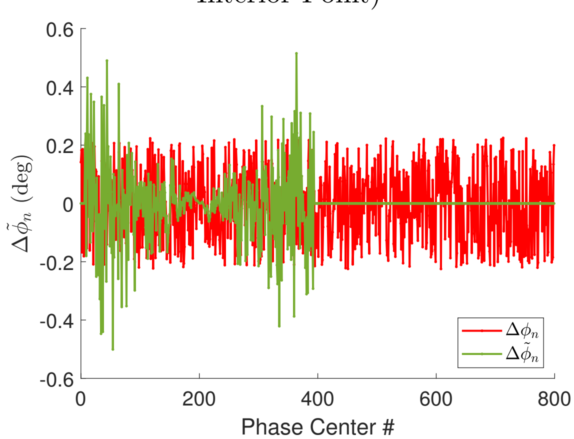
Figure 6.4: Numerical Results for 1-Target Case (Error Model I)



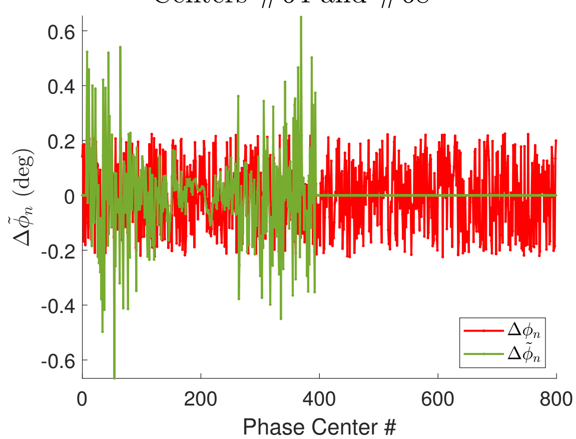
(a) E_g (Gradient Descent and Interior-Point)



(b) Optimization Landscape Between Phase Centers #64 and #68

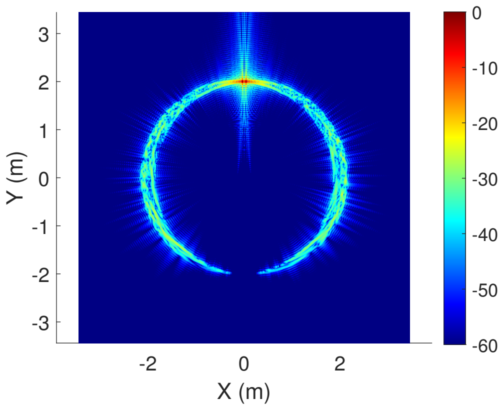


(c) $\Delta\phi_n$ and $\Delta\widetilde{\phi}_n$ (Gradient Descent)

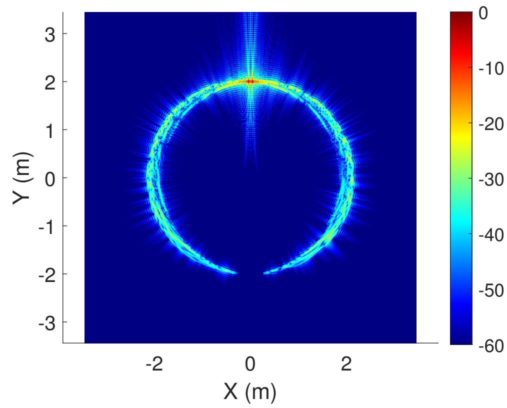


(d) $\Delta\phi_n$ and $\Delta\widetilde{\phi}_n$ (Interior-Point)

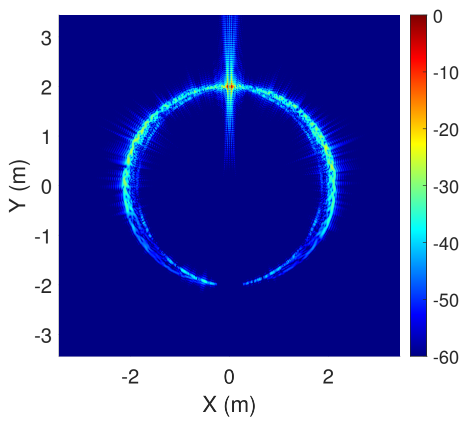
Figure 6.5: Numerical Results for 2-Target Case (Error Model I)
(More sub-figures on the next page)



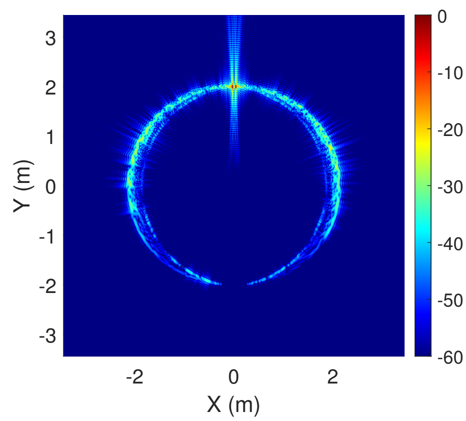
(e) Ground Truth SAR Image
$E_g = 7.0692$



(f) Initial SAR Image
$E_g = 7.0378$

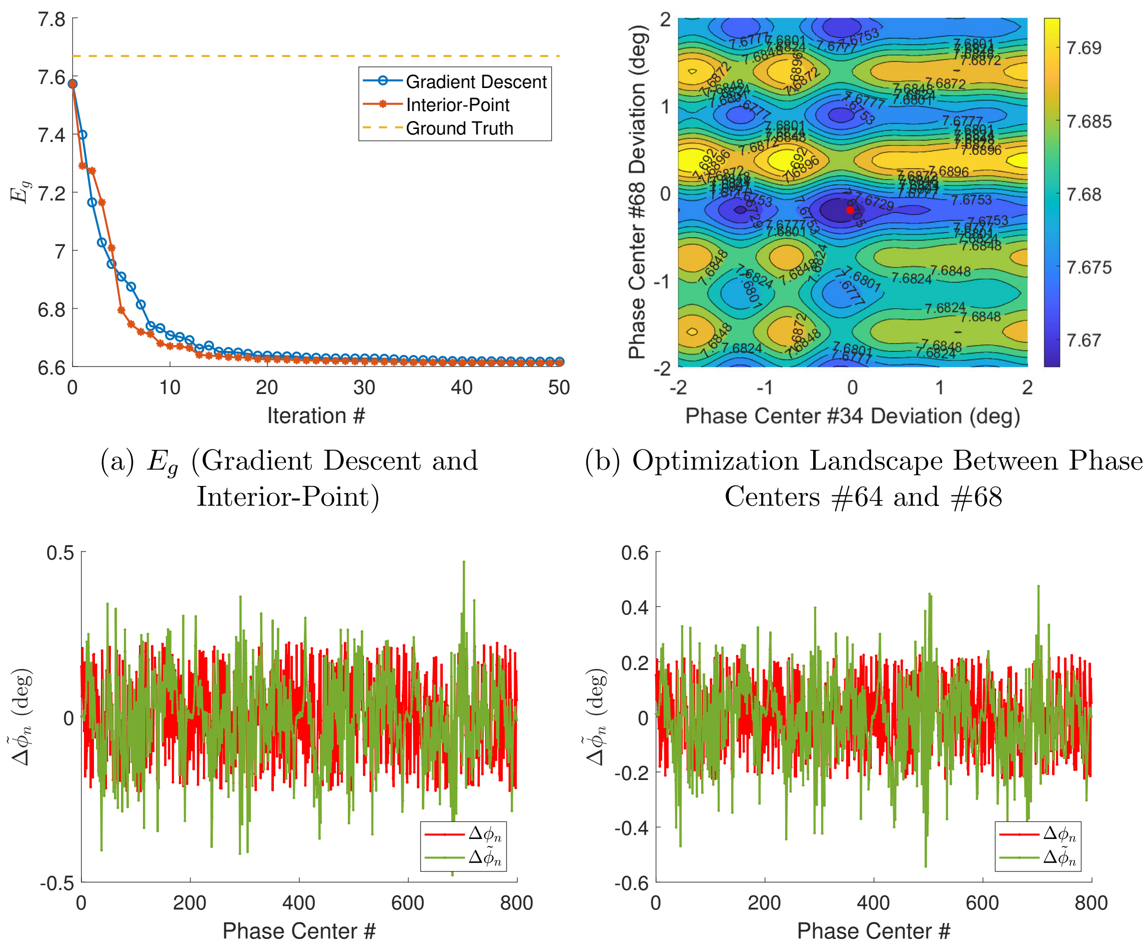


(g) Autofocused Image (Gradient Descent)
$E_g = 5.6060$



(h) Autofocused Image (Interior-Point)
$E_g = 5.5509$

Figure 6.5: Numerical Results for 2-Target Case (Error Model I)



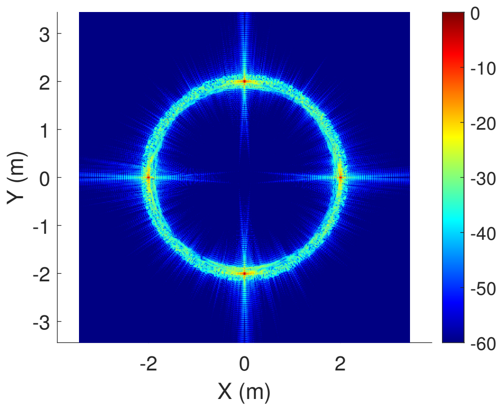
(a) E_g (Gradient Descent and Interior-Point)

(b) Optimization Landscape Between Phase Centers #64 and #68

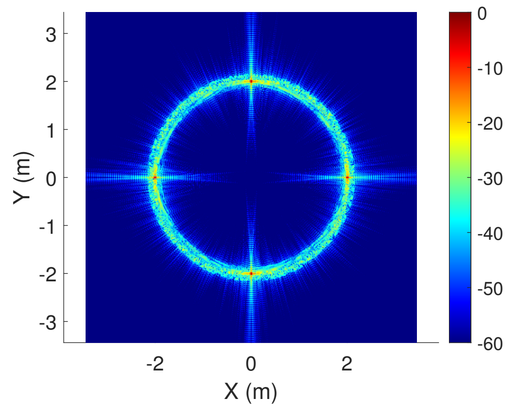
(c) $\Delta\phi_n$ and $\Delta\widetilde{\phi}_n$ (Gradient Descent)

(d) $\Delta\phi_n$ and $\Delta\widetilde{\phi}_n$ (Interior-Point)

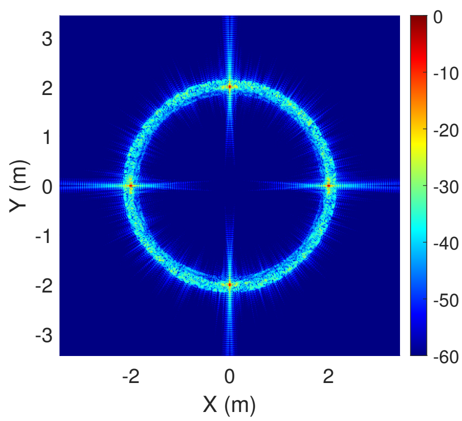
Figure 6.6: Numerical Results for 4-Target Case (Error Model I)
(More sub-figures on the next page)



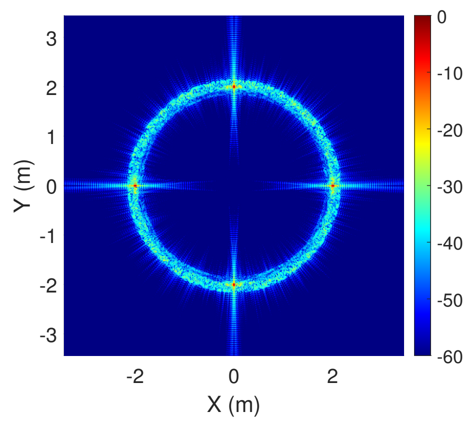
(e) Ground Truth SAR Image
$E_g = 7.6686$



(f) Initial SAR Image
$E_g = 7.5729$

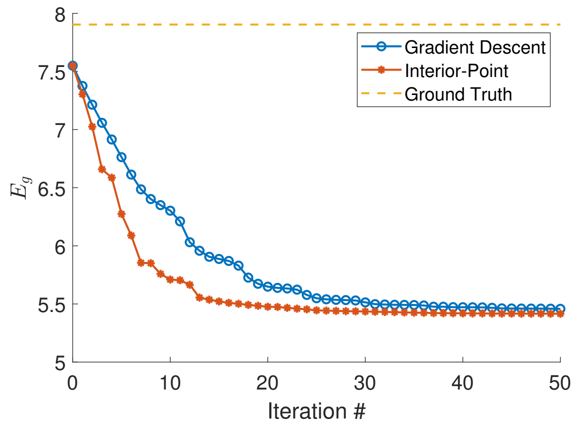


(g) Autofocused Image (Gradient Descent)
$E_g = 6.6167$

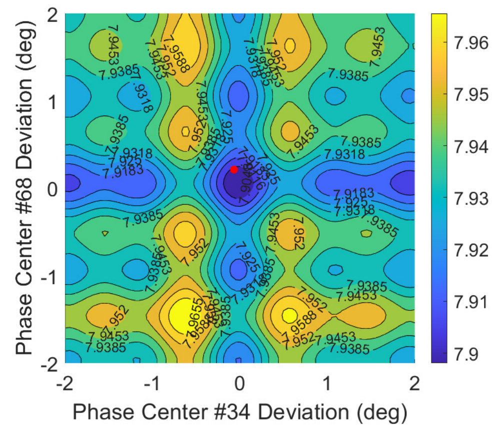


(h) Autofocused Image (Interior-Point)
$E_g = 6.6127$

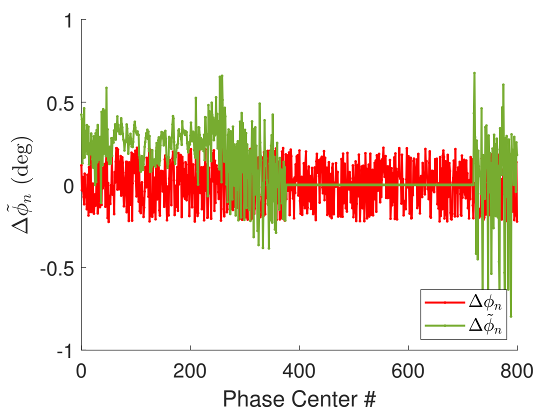
Figure 6.6: Numerical Results for 4-Target Case (Error Model I)



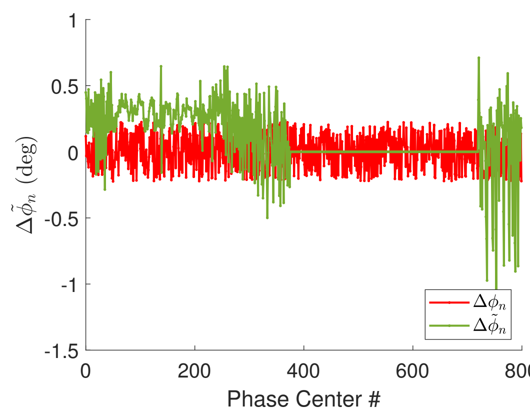
(a) E_g (Gradient Descent and Interior-Point)



(b) Optimization Landscape Between Phase Centers #64 and #68

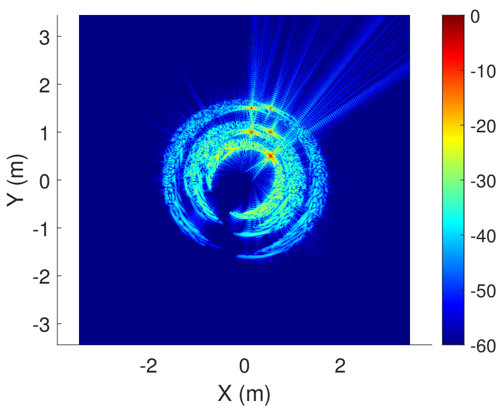


(c) $\Delta\phi_n$ and $\Delta\widetilde{\phi}_n$ (Gradient Descent)

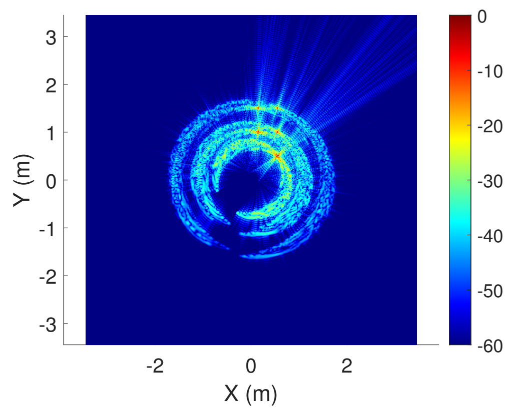


(d) $\Delta\phi_n$ and $\Delta\widetilde{\phi}_n$ (Interior-Point)

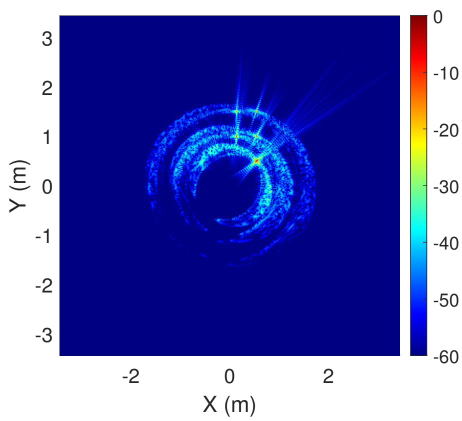
Figure 6.7: Numerical Results for 5-Target Case (Error Model I) (More sub-figures on the next page)



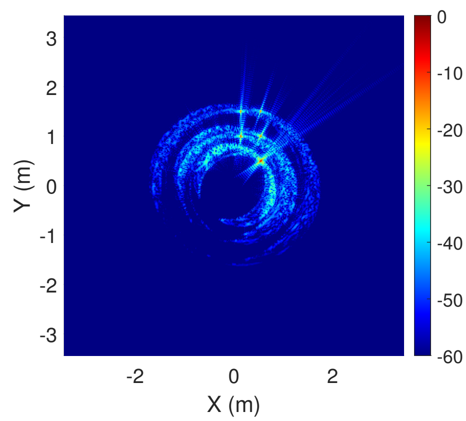
(e) Ground Truth SAR Image
$E_g = 7.9033$



(f) Initial SAR Image
$E_g = 7.5494$

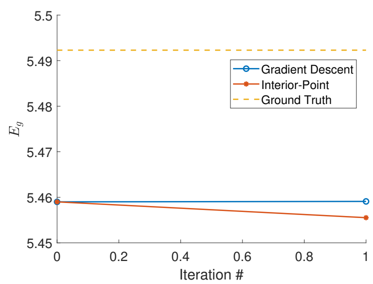


(g) Autofocused Image (Gradient Descent)
$E_g = 5.4579$

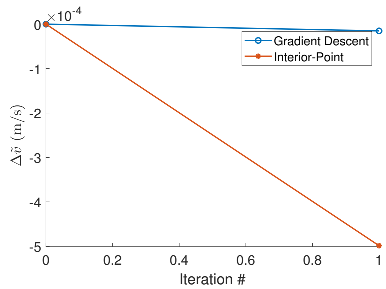


(h) Autofocused Image (Interior-Point)
$E_g = 5.4159$

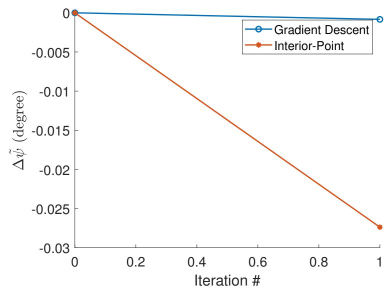
Figure 6.7: Numerical Results for 5-Target Case (Error Model I)



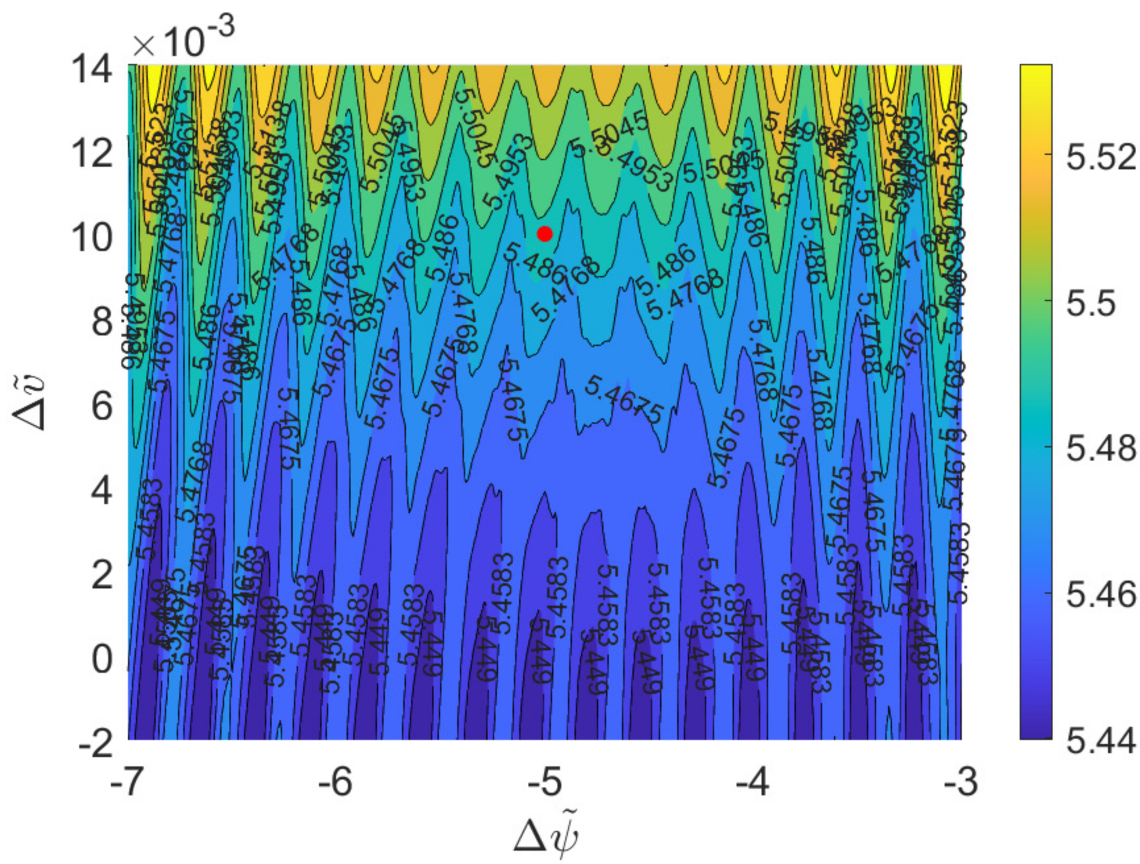
(a) E_g



(b) $\Delta\widetilde{v}$

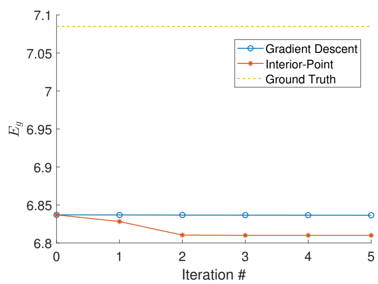


(c) $\Delta\widetilde{\psi}$

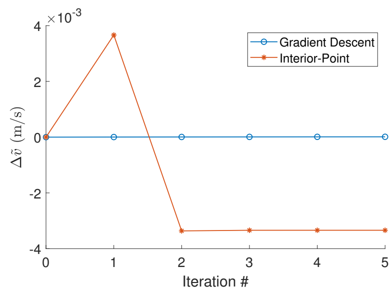


(d) Optimization Landscape between $\Delta\widetilde{v}$ and $\Delta\widetilde{\psi}$

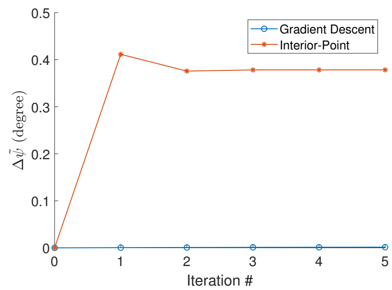
Figure 6.8: Numerical Results for 1-Target Case (Error Model II)



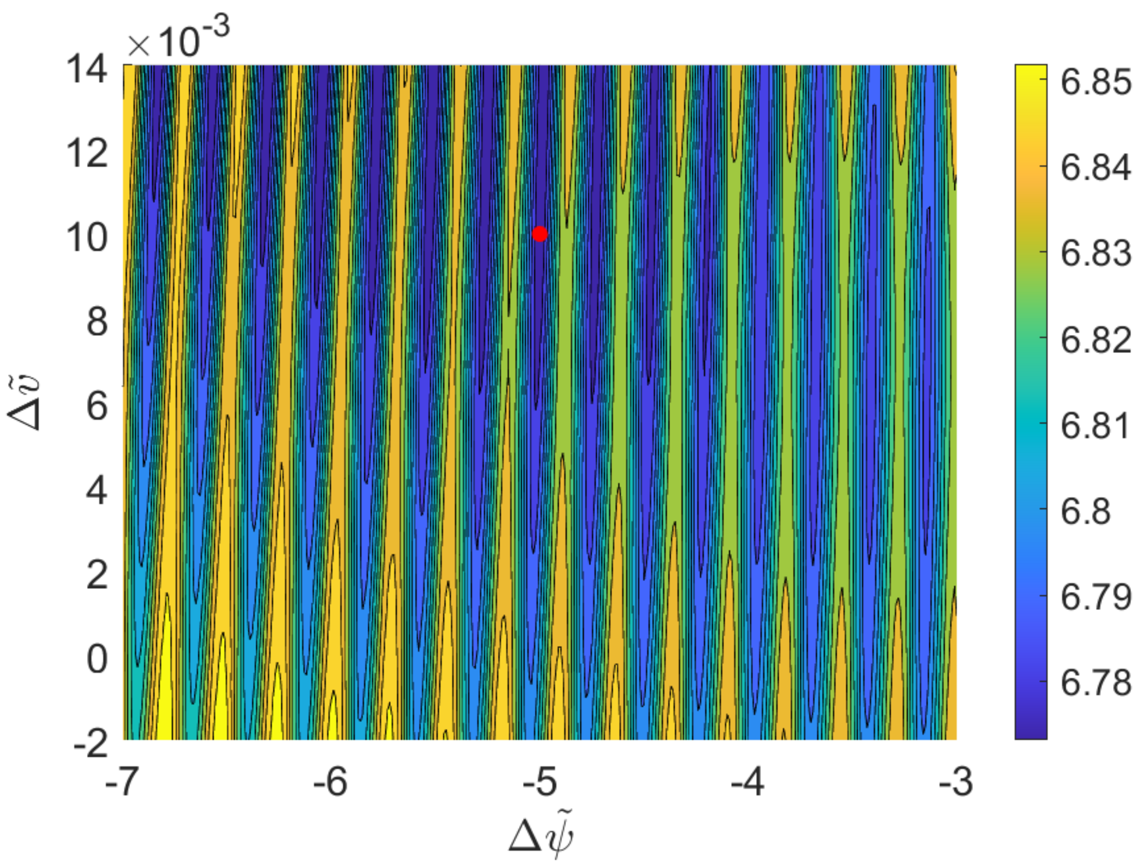
(a) E_g



(b) $\Delta\widetilde{v}$

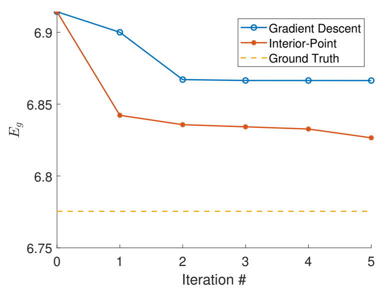


(c) $\Delta\widetilde{\psi}$

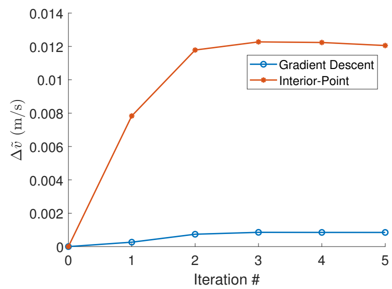


(d) Optimization Landscape between $\Delta\widetilde{v}$ and $\Delta\widetilde{\psi}$

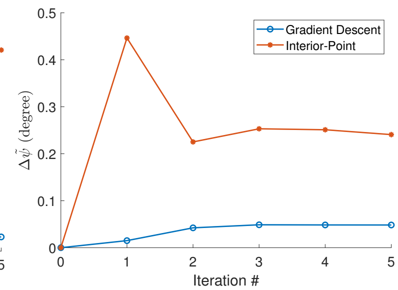
Figure 6.9: Numerical Results for 2-Target Case (Error Model II)



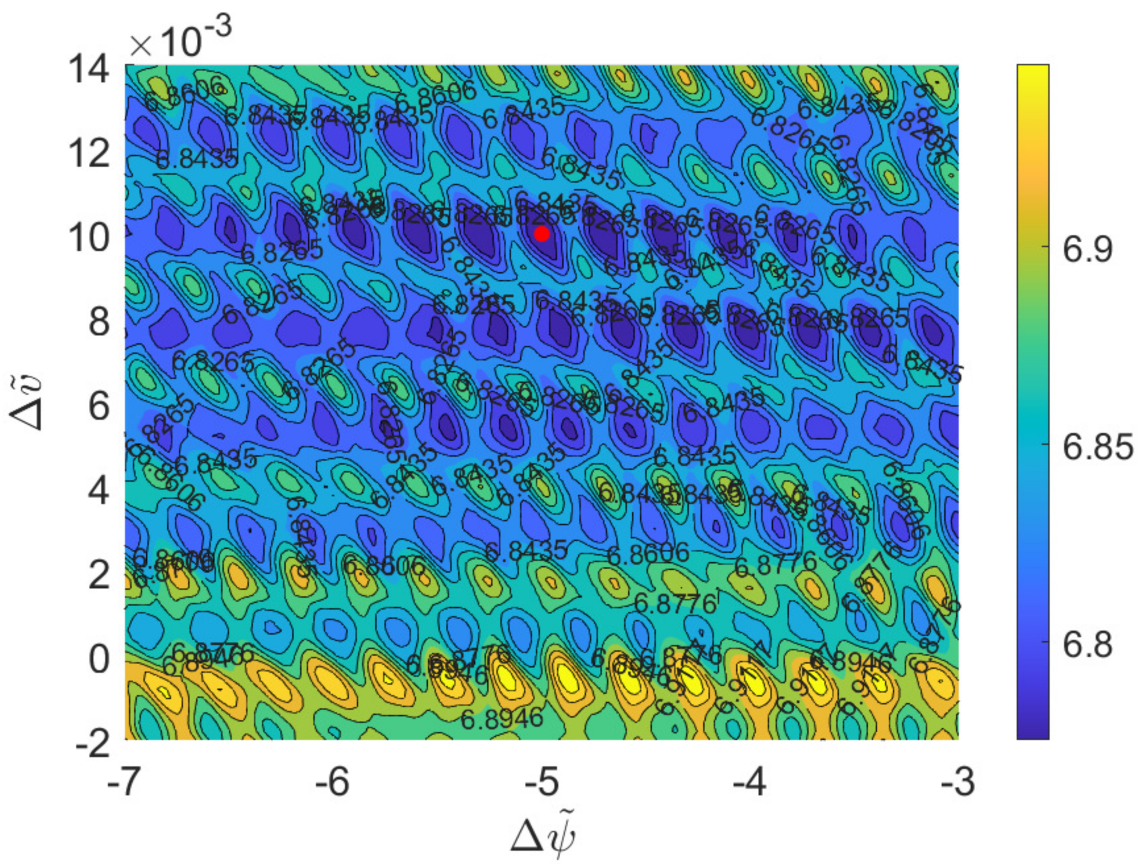
(a) E_g



(b) $\Delta\widetilde{v}$

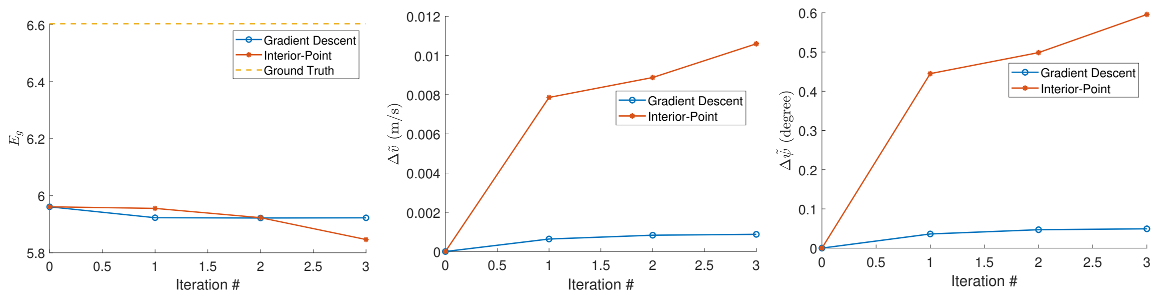


(c) $\Delta\widetilde{\psi}$

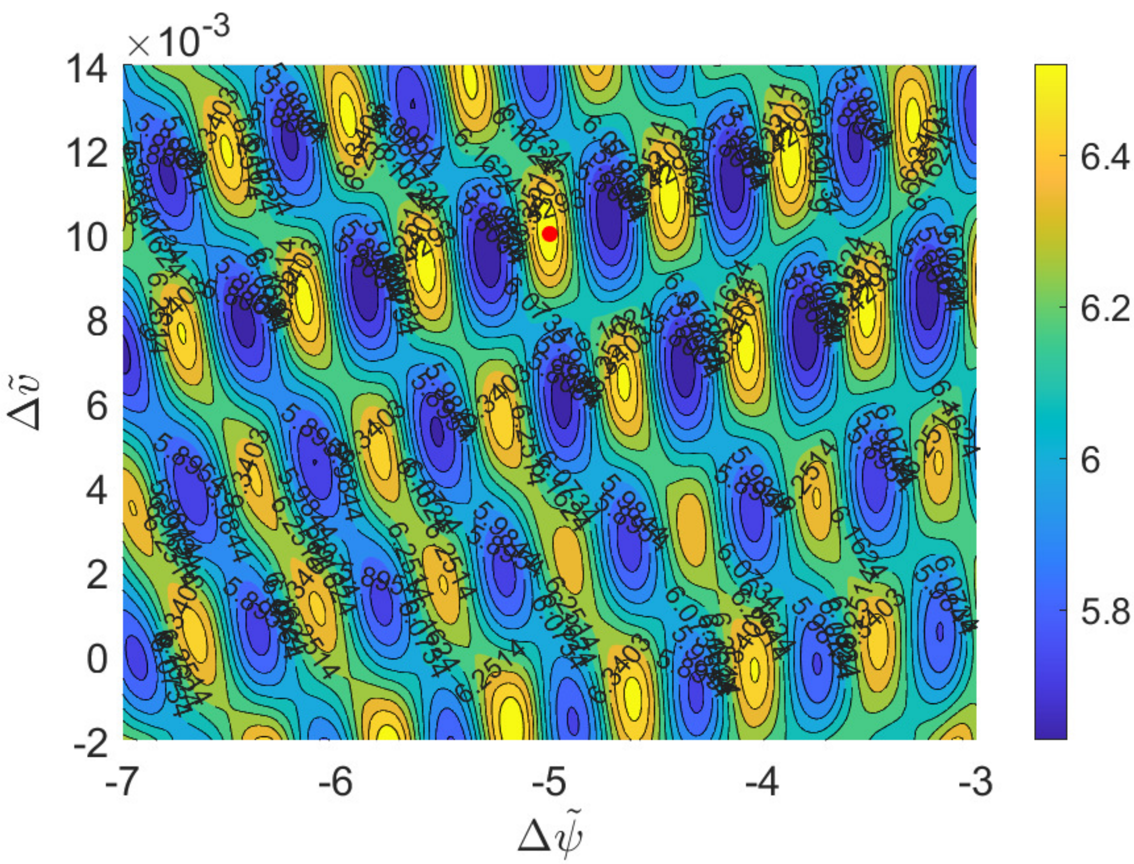


(d) Optimization Landscape between $\Delta\widetilde{v}$ and $\Delta\widetilde{\psi}$

Figure 6.10: Numerical Results for 4-Target Case (Error Model II)



(a) E_g (b) $\Delta\widetilde{v}$ (c) $\Delta\widetilde{\psi}$



(d) Optimization Landscape between $\Delta\widetilde{v}$ and $\Delta\widetilde{\psi}$

Figure 6.11: Numerical Results for 5-Target Case (Error Model II)

Chapter 7

Conclusion

7.1 Summary

In this thesis, we delved into the utilization of mmWave radars for target localization and imaging. The unique characteristic of mmWave radars lies in their short wavelength, which enables the integration of multiple antennas within a confined space. This advancement facilitates the implementation of sophisticated algorithms on IoT devices. Building upon these advantages, we achieved the following key objectives:

First, we introduced a novel method for locating multiple targets using an mmWave radar equipped with a trapezoid virtual antenna array. By employing Barone's method, we effectively separated the signals reflected from each target on the receiver antennas and estimated both the number of targets and their respective ranges. Then, the Least-Square algorithm was used to estimate the azimuth and elevation angles of the targets. The simulation and testbed experiments demonstrated that the proposed algorithm outperforms traditional 2D-FFT and MUSIC algorithms in terms of location accuracy. Moreover, the proposed algorithm does not heavily depend on the number of sampled data or the antenna arrangement to detect a large area.

Second, we analyzed the array pattern of ULA and UCA, where UCA represents a special case of ROSAR when the rotation platform remains stationary and rotation speed remains constant. Then, based on the characters of ULA and UCA, we gave the limitations on the designed parameters of ROSAR to meet application requirements. Next, we presented the built real-world robotic ROSAR system designed for evaluating imaging algorithms. The measurement study showed that the radar rotation speed and rover movement cannot be precisely controlled, potentially leading to the radar location deviations and blurry SAR images.

Third, we proposed an efficient and robust ROSAR imaging approach based on sparse array synthesis. To mitigate heavy computational complexity of tradition BPA and ensure robustness against array manifold errors induced by unstable radar rotation, our approach employs azimuth-dimension matched filtering using carefully selected phase centers with well-designed weights. We also considered mainlobe and sidelobe levels while deriving the optimal weights to maintain image quality. Thanks to the circular array's symmetry, these optimal weights only need to be computed offline in advance for one direction, making them usable for imaging in any direction. The resulting robust sparse array synthesis problem is a non-convex optimization problem with quadratic constraints. We devised an algorithm based on feasible point pursuit and successive convex approximation to solve it. Extensive simulation studies and experimental evaluations using the built ROSAR system demonstrated that the proposed algorithm achieves image quality comparable to that of BPA, but with a substantial reduction in computational time of up to 90%.

Fourth, we introduced an alternative method to mitigate image blurriness caused by array manifold errors, that is employing MEA to solve for optimal motion parameters and enhance the overall quality of SAR images. We modelled array manifold errors as unstable radar rotation and rover linear movements of the ROSAR system. To solve the corresponding optimization problem of MEA, we utilized gradient descent and interior-point methods. Simulation results indicated that MEA can successfully sharpen SAR images by reducing their entropies, although it cannot yield the actual deviations of the virtual phase centers.

7.2 Future Works

Here is a list of possible improvements and research directions.

First, the current robotic ROSAR system relies on an outdated Arlo robot, which has been discontinued for some time. To enhance reliability, we recommend to rebuild the system on a more robust platform. For example, newer robots like the TurtleBot 4. For the new platform, it's crucial to increase the robot's battery capacity to support power-consuming components such as the radar, the Lidar, and motors, ensuring enough operation time. Furthermore, alternative methods must be explored for measuring rover trajectory with higher accuracy. One can start with employing better sensors that do not suffer from wheel slippage.

Second, the proposed efficient and robust algorithm in Chapter 5 are expected be promoted to apply for general cases, e.g., moving ROSAR platforms. Then, the ROSAR system is able to undertake Simultaneous Localization and Mapping (SLAM) tasks.

Third, we have explored two types of error models and employed MEA for autofocusing in Chapter 6. However, only using MEA cannot guarantee the performance or estimate the ground truth. We would like to design new algorithm that can fit ROSAR autofocusing well and conduct real-world experiments to verify the algorithm.

Appendix A

Transformation of the Least Square Optimization Problem for (3.3.18)

We now show the detailed transformation steps of the Least Square optimization problem:

$$\min_{\alpha_i,\omega_{x,i},\omega_{z,i}} f\left(\alpha_i,\omega_{x,i},\omega_{z,i}\right), \tag{A.0.1}$$

where $f\left(\alpha_i,\omega_{x,i},\omega_{z,i}\right) = \left\|\widehat{\mathbf{c}}_i - \mathbf{c}_i\right\|^2$. Since $\left\|\mathbf{c}_i\right\|^2 = 12\alpha_i^2$,

$$\begin{aligned} f\left(\alpha_i,\omega_{x,i},\omega_{z,i}\right) &= \left\|\widehat{\mathbf{c}}_i\right\|^2 + \left\|\mathbf{c}_i\right\|^2 - \widehat{\mathbf{c}}_i^H\mathbf{c}_i - \mathbf{c}_i^H\widehat{\mathbf{c}}_i \\ &= \left\|\widehat{\mathbf{c}}_i\right\|^2 + 12\alpha_i^2 - 2\mathrm{Re}\left(\widehat{\mathbf{c}}_i^H\mathbf{c}_i\right). \end{aligned}$$

Let $\widetilde{\mathbf{c}}_i = e^{j\psi}\mathbf{h}_i^*$, where $\psi = 2\pi\left(\widehat{\tau}_i S T_{\mathrm{Start}} + f_c\widehat{\tau}_i - \frac{1}{2}S\widehat{\tau}_i^2\right)$. Then, $\mathbf{c}_i = \alpha_i\widetilde{\mathbf{c}}_i$ and

$$f\left(\alpha_i,\omega_{x,i},\omega_{z,i}\right) = \left\|\widehat{\mathbf{c}}_i\right\|^2 + 12\alpha_i^2 - 2\alpha_i\mathrm{Re}\left(\widehat{\mathbf{c}}_i^H\widetilde{\mathbf{c}}_i\right). \tag{A.0.2}$$

Take the derivative of (A.0.2) with respect to α_i, we have $\frac{\partial f}{\partial \alpha_i} = 24\alpha_i - 2\mathrm{Re}\left(\widehat{\mathbf{c}}_i^H \widetilde{\mathbf{c}}_i\right)$. From the KKT conditions, to minimize (A.0.1), we have $\widehat{\alpha}_i = \frac{\mathrm{Re}\left(\widehat{\mathbf{c}}_i^H \widetilde{\mathbf{c}}_i\right)}{12}$. Next, we substitute $\widehat{\alpha}_i$ in $f\left(\alpha_i, \omega_{x,i}, \omega_{z,i}\right)$ to get

$$\begin{aligned} f\left(\widehat{\alpha}_i, \omega_{x,i}, \omega_{z,i}\right) &= \left\|\widehat{\mathbf{c}}_i\right\|^2 + 12\left(\frac{\mathrm{Re}\left(\widehat{\mathbf{c}}_i^H \widetilde{\mathbf{c}}_i\right)}{12}\right)^2 - 2\frac{\mathrm{Re}\left(\widehat{\mathbf{c}}_i^H \widetilde{\mathbf{c}}_i\right)}{12}\mathrm{Re}\left(\widehat{\mathbf{c}}_i^H \widetilde{\mathbf{c}}_i\right) \\ &= \left\|\widehat{\mathbf{c}}_i\right\|^2 - \frac{1}{12}\left[\mathrm{Re}\left(\widehat{\mathbf{c}}_i^H \widetilde{\mathbf{c}}_i\right)\right]^2. \end{aligned}$$

Clearly, since $\left\|\widehat{\mathbf{c}}_i\right\|^2$ is a constant, in order to minimize $f\left(\widehat{\alpha}_i, \omega_{x,i}, \omega_{z,i}\right)$, we need to maximize $\left|\mathrm{Re}\left(\widehat{\mathbf{c}}_i^H \widetilde{\mathbf{c}}_i\right)\right|$. Due to the phase shift differences, we consider the bottom 8 virtual antennas and top 4 receiver antennas separately, and write $\left|\mathrm{Re}\left(\widehat{\mathbf{c}}_i^H \widetilde{\mathbf{c}}_i\right)\right|$ as $\left|\mathrm{Re}\left(\widehat{\mathbf{c}}_i^{1:8H} \widetilde{\mathbf{c}}_i^{1:8}\right) + \mathrm{Re}\left(\widehat{\mathbf{c}}_i^{9:12H} \widetilde{\mathbf{c}}_i^{9:12}\right)\right|$. Let

$$B^{1:8} = \left|B^{1:8}\right| e^{j\phi_{B^{1:8}}} = \widehat{\mathbf{c}}_i^{1:8H}\left[1, e^{-j\omega_{x,i}}, \ldots, e^{-j7\omega_{x,i}}\right]^T, \tag{A.0.3}$$

$$B^{9:12} = \left|B^{9:12}\right| e^{j\phi_{B^{9:12}}} = \widehat{\mathbf{c}}_i^{9:12H}\left[1, e^{-j\omega_{x,i}}, e^{-j2\omega_{x,i}}, e^{-j3\omega_{x,i}}\right]^T. \tag{A.0.4}$$

In order to maximize $\left|\mathrm{Re}\left(\widehat{\mathbf{c}}_i^{1:8H} \widetilde{\mathbf{c}}_i^{1:8}\right) + \mathrm{Re}\left(\widehat{\mathbf{c}}_i^{9:12H} \widetilde{\mathbf{c}}_i^{9:12}\right)\right|$, the following conditions must be achieved:

$$\begin{cases} \phi_{B^{1:8}} + \psi = 2k\pi \\ \phi_{B^{9:12}} + \psi + \omega_{z,i} - 2\omega_{x,i} = 2k\pi \end{cases} (k \in \mathbb{Z}). \tag{A.0.5}$$

Under the above conditions, (A.0.1) is equivalent to

$$\max_{\omega_{x,i}, \omega_{z,i}} \left|\left|B^{1:8}\right| + \left|B^{9:12}\right|\right|.$$

The form of $B^{1:8}$ and $B^{9:12}$ can be determined by taking Discrete Fourier Transform (DFT) of $\widehat{\mathbf{c}}_i^{1:8H}$ and $\widehat{\mathbf{c}}_i^{9:12H}$. Usually, to achieve a higher precision, we append 0s to $\widehat{\mathbf{c}}_i^{1:8H}$ and $\widehat{\mathbf{c}}_i^{9:12H}$ to make their lengths to a pre-set value, e.g., $\widehat{\mathbf{d}}_i^{1:8} = \begin{bmatrix} \widehat{\mathbf{c}}_i^{1:8H} \\ \mathbf{0} \end{bmatrix}$ and $\widehat{\mathbf{d}}_i^{9:12} = \begin{bmatrix} \widehat{\mathbf{c}}_i^{9:12H} \\ \mathbf{0} \end{bmatrix}$. Next, we can take DFT to $\widehat{\mathbf{d}}_i^{1:8}$ and $\widehat{\mathbf{d}}_i^{9:12}$ as

$$\begin{aligned} \widetilde{B}^{1:8}[l] &= \sum_{n=0}^{N_{\mathrm{DFT}}-1} \widehat{\mathbf{d}}_i^{1:8}[n]\, e^{-j\omega_{x,i}[l]n}, \\ \widetilde{B}^{9:12}[l] &= \sum_{n=0}^{N_{\mathrm{DFT}}-1} \widehat{\mathbf{d}}_i^{9:12}[n]\, e^{-j\omega_{x,i}[l]n}. \end{aligned}$$

Once we add the magnitude of the DFT results together, (A.0.1) has been transformed to

$$\widehat{l} = \arg_l \max \left| \left| \widetilde{B}^{1:8}[l] \right| + \left| \widetilde{B}^{9:12}[l] \right| \right|.$$

Appendix B

The Derivatives of E, E_2 and $I(x,y)$ in Eq. $(6.3.1) \sim (6.3.4)$

B.1 Error Model I

Detailed steps to obtain the first derivatives of E and E_2 with respect to $\Delta\widetilde{\phi}_{\widehat{n}}$ are give by

$$\begin{aligned}
\frac{\partial E}{\partial \Delta\widetilde{\phi}_{\widehat{n}}} &= \sum_x \sum_y \frac{\partial \left|I(x,y)\right|^2}{\partial \Delta\widetilde{\phi}_{\widehat{n}}} \\
&= \sum_x \sum_y I(x,y)\frac{\partial I^*(x,y)}{\partial \Delta\widetilde{\phi}_{\widehat{n}}} + I^*(x,y)\frac{\partial I(x,y)}{\partial \Delta\widetilde{\phi}_{\widehat{n}}} \\
&= \sum_x \sum_y 2Re\left\{I^*(x,y)\frac{\partial I(x,y)}{\partial \Delta\widetilde{\phi}_{\widehat{n}}}\right\},
\end{aligned}$$

where

$$\frac{\partial I(x,y)}{\partial \Delta\widetilde{\phi}_{\widehat{n}}} = j\sum_{m=0}^{M-1} y_{IF,\widehat{n}}(m)e^{-j2k_m\widetilde{R}_{\widehat{n}}}\frac{2k_m Rr\sin\left(\phi - \phi_{\widehat{n}} - \Delta\widetilde{\phi}_{\widehat{n}}\right)}{\widetilde{R}_{\widehat{n}}};$$

$$\begin{aligned}\frac{\partial E_2}{\partial \Delta \widetilde{\phi}_{\widehat{n}}} &= \sum_x \sum_y \left(1 + \ln |I(x,y)|^2\right) \frac{\partial |I(x,y)|^2}{\partial \Delta \widetilde{\phi}_{\widehat{n}}} \\ &= \sum_x \sum_y \left(1 + \ln |I(x,y)|^2\right) 2Re\left\{ I^*(x,y) \frac{\partial I(x,y)}{\partial \Delta \widetilde{\phi}_{\widehat{n}}} \right\}.\end{aligned}$$

B.2 Error Model II

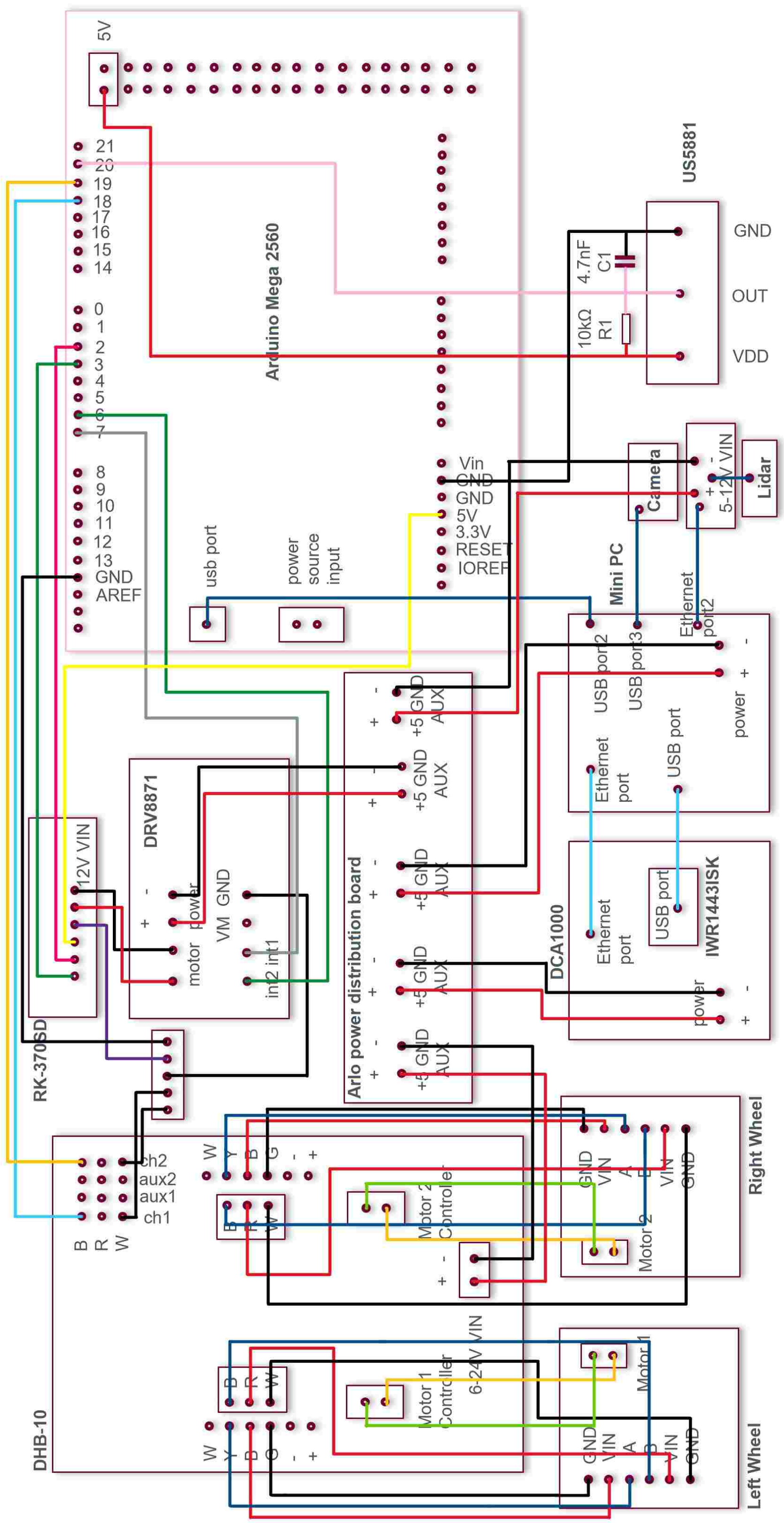
Detailed steps to derivate the first derivatives of $I(x,y)$ with respect to $\Delta\widetilde{v}$ and $\Delta\widetilde{\psi}$ are give by

$$\begin{aligned}\frac{\partial I(x,y)}{\partial \Delta \widetilde{v}} &= -j \sum_{n=N_{\min}}^{N_{\max}} \sum_{m=0}^{M-1} y_{IF,n}(m) e^{-j2k_m \widetilde{R}_n} \\ &\quad \cdot \frac{2k_m n t_f \left((v + \Delta\widetilde{v})\, n t_f + r\cos\left(\frac{2\pi n}{N} + \Delta\widetilde{\psi}\right) - x \right)}{\widetilde{R}_n},\end{aligned}$$

$$\begin{aligned}\frac{\partial I(x,y)}{\partial \Delta \widetilde{\psi}} &= -j \sum_{n=N_{\min}}^{N_{\max}} \sum_{m=0}^{M-1} y_{IF,n}(m) e^{-j2k_m \widetilde{R}_n} \\ &\quad \cdot \frac{2k_m r \left(\sin\left(\frac{2\pi n}{N} + \Delta\widetilde{\psi}\right) \left((v + \Delta\widetilde{v})\, n t_f - x\right) + \cos\left(\frac{2\pi n}{N} + \Delta\widetilde{\psi}\right) y \right)}{\widetilde{R}_n}\end{aligned}$$

Appendix C

The Circuit Diagram of the Robotic ROSAR System


Arduino Mega 2560
5V
usb port
power source input
US5881
GND
OUT
VDD
4.7nF C1
10kΩ R1
Camera
Lidar
5-12V VIN
Mini PC
Ethernet port2
USB port2
USB port3
USB port
Ethernet port
power
DCA1000
IWR1443ISK
Arlo power distribution board
DRV8871
RK-370SD
12V VIN
DHB-10
Motor 2 Controller
Motor 1 Controller
6-24V VIN
Motor 2
Motor 1
Right Wheel
Left Wheel

Bibliography

[1] H. Akaike. 1974. A new look at the statistical model identification. *IEEE Trans. Automat. Control* 19, 6 (1974), 716–723.

[2] Faiza Ali, Georg Bauer, and Martin Vossiek. 2014. A rotating synthetic aperture radar imaging concept for robot navigation. *IEEE Transactions on Microwave Theory and Techniques* 62, 7 (2014), 1545–1553.

[3] Faiza Ali, Alexander Urban, and Martin Vossiek. 2010. A high resolution 2D omnidirectional synthetic aperture radar scanner at K band. In *The 7th European Radar Conference*. IEEE, 503–506.

[4] Faiza Ali, Alexander Urban, and Martin Vossiek. 2011. A short range synthetic aperture imaging radar with rotating antenna. *International Journal of Electronics and Telecommunications* 57 (2011), 97–102.

[5] Mostafa Alizadeh, George Shaker, João Carlos Martins De Almeida, Plinio Pelegrini Morita, and Safeddin Safavi-Naeini. 2019. Remote monitoring of human vital signs using mm-wave FMCW radar. *IEEE Access* 7 (2019), 54958–54968.

[6] IEEE Standards Association et al. 2013. 145-2013-IEEE standard for definitions of terms for antennas. *IEEE, New York, NY* (2013).

[7] Piero Barone. 2008. A new transform for solving the noisy complex exponentials approximation problem. *Journal of Approximation Theory* 155, 1 (2008), 1–27.

[8] Jerald L Bauck. 2021. A rationale for backprojection in spotlight synthetic aperture radar image formation. *IEEE Aerospace and Electronic Systems Magazine* 37, 5 (2021), 44–66.

[9] Amir Beck, Aharon Ben-Tal, and Luba Tetruashvili. 2010. A sequential parametric convex approximation method with applications to nonconvex truss topology design problems. *Journal of Global Optimization* 47, 1 (2010), 29–51.

[10] Fabrizio Berizzi and Giovanni Corsini. 1996. Autofocusing of inverse synthetic aperture radar images using contrast optimization. *IEEE Trans. Aerospace Electron. Systems* 32, 3 (1996), 1185–1191.

[11] Hui Bi, Guoan Bi, Bingchen Zhang, Wen Hong, and Yirong Wu. 2019. From theory to application: Real-time sparse SAR imaging. *IEEE Transactions on Geoscience and Remote Sensing* 58, 4 (2019), 2928–2936.

[12] Ciro Cafforio, Claudio Prati, and Fabio Rocca. 1991. SAR data focusing using seismic migration techniques. *IEEE Trans. Aerospace Electron. Systems* 27, 2 (1991), 194–207.

[13] Emmanuel J Candes, Michael B Wakin, and Stephen P Boyd. 2008. Enhancing sparsity by reweighted L1 minimization. *Journal of Fourier Analysis and Applications* 14, 5 (2008), 877–905.

[14] J. Capon. 1969. High-resolution frequency-wavenumber spectrum analysis. *Proc. IEEE* 57, 8 (1969), 1408–1418. https://doi.org/10.1109/PROC.1969.7278

[15] Fang-Jiong Chen, Carrson C Fung, Chi-Wah Kok, and Sam Kwong. 2007. Estimation of two-dimensional frequencies using modified matrix pencil method. *IEEE Transactions on Signal Processing* 55, 2 (2007), 718–724.

[16] Mark T Crockett and David Long. 2013. An introduction to synthetic aperture radar: A high-resolution alternative to optical imaging. (2013).

[17] Ian Cumming and John Bennett. 1979. Digital processing of SEASAT SAR data. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 4. IEEE, 710–718.

[18] Nicholas C Currie and Charles E Brown. 1987. *Principles and applications of millimeter-wave radar.* Artech House Boston.

[19] Qi Dong, Guang-Cai Sun, Zemin Yang, Liang Guo, and Mengdao Xing. 2017. Cartesian factorized backprojection algorithm for high-resolution spotlight SAR imaging. *IEEE Sensors Journal* 18, 3 (2017), 1160–1168.

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

[21] Ulku Cilek Doyuran and Yalcin Tanik. 2008. detection of multiple targets in non-Gaussian clutter. In *2008 IEEE Radar Conference*. 1–5. `https://doi.org/10.1109/RADAR.2008.4721041`

[22] Warren P Du Plessis. 2011. Weighted thinned linear array design with the iterative FFT technique. *IEEE Transactions on Antennas and Propagation* 59, 9 (2011), 3473–3477.

[23] Chen Duofang, Chen Baixiao, and Qin Guodong. 2008. Angle estimation using ESPRIT in MIMO radar. *Electronics Letters* 44, 12 (2008), 770–771.

[24] Paul H Eichel and CV Jakowatz. 1989. Phase-gradient algorithm as an optimal estimator of the phase derivative. *Optics Letters* 14, 20 (1989), 1101–1103.

[25] Benjamin Fuchs. 2012. Synthesis of sparse arrays with focused or shaped beampattern via sequential convex optimizations. *IEEE Transactions on Antennas and Propagation* 60, 7 (2012), 3499–3503.

[26] Haniel Gabai and Avishay Eyal. 2016. SNR characterization in distributed acoustic sensing. In *Sixth European Workshop on Optical Fibre Sensors*, Vol. 9916. SPIE, 411–414.

[27] A George Jr et al. 1975. *Essentials of Padé approximants*. Elsevier.

[28] Michael C Grant and Stephen P Boyd. 2011. CVX Research, Inc. *CVX: Matlab software for disciplined convex programming, version* 2 (2011).

[29] Peter D Grünwald, In Jae Myung, and Mark A Pitt. 2005. *Advances in minimum description length: Theory and applications*. MIT press.

[30] Y. Hua. 1991. Estimating two-dimensional frequencies by matrix enhancement and matrix pencil. In *1991 International Conference on Acoustics, Speech, and Signal Processing*. 3073–3076 vol.5.

[31] Yingbo Hua and Tapan K Sarkar. 1990. Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 38, 5 (1990), 814–824.

[32] Seifallah Jardak, Tero Kiuru, Mikko Metso, Pekka Pursula, Janne Häkli, Mervi Hirvonen, Sajid Ahmed, and Mohamed-Slim Alouini. 2016. Detection and localization of multiple short range targets using FMCW radar signal. In *2016 Global Symposium on Millimeter Waves (GSMM) & ESA Workshop on Millimetre-Wave Technology and Applications*. IEEE, 1–4.

[33] Chen Jinli, Gu Hong, and Su Weimin. 2008. Angle estimation using ESPRIT without pairing in MIMO radar. *Electronics Letters* 44, 24 (2008), 1422–1423.

[34] Bongsoon Kang, Harish M Subbaram, and Bernard D Steinberg. 1990. Improved adaptive-beamforming target for self-calibrating a distorted phased array. *IEEE Transactions on Antennas and Propagation* 38, 2 (1990), 186–194.

[35] Will PMN Keizer. 2008. Linear array thinning using iterative FFT techniques. *IEEE Transactions on Antennas and Propagation* 56, 8 (2008), 2757–2760.

[36] Helmut Klausing. 1989. Feasibility of a synthetic aperture radar with rotating antennas (ROSAR). In *1989 19th European Microwave Conference*. IEEE, 287–299.

[37] Thomas J Kragh and A Alaa Kharbouch. 2006. Monotonic iterative algorithm for minimum-entropy autofocus. In *Adaptive Sensor Array Processing (ASAP) Workshop*, Vol. 40. 1147–1159.

[38] Yunseong Lee, Chanhong Park, Taeyoung Kim, Yeongyoon Choi, Kiseon Kim, Dongho Kim, Myung-Sik Lee, and Dongkeun Lee. 2021. Source Enumeration Approaches Using Eigenvalue Gaps and Machine Learning Based Threshold for Direction-of-Arrival Estimation. *Applied Sciences* 11, 4 (2021), 1942.

[39] Dong Li, Hongqing Liu, Yong Liao, and Xiaogang Gui. 2015. A novel helicopter-borne rotating SAR imaging model and algorithm based on inverse chirp-Z transform using frequency-modulated continuous wave. *IEEE Geoscience and Remote Sensing Letters* 12, 8 (2015), 1625–1629.

[40] Jian Li and Petre Stoica. 2009. *MIMO radar signal processing*. Vol. 7. Wiley Online Library.

[41] Ying-Chun Li, Byunggil Choi, Jong-Wha Chong, and Daegun Oh. 2018. 3D target localization of modified 3D MUSIC for a triple-channel K-band radar. *Sensors* 18, 5 (2018), 1634.

[42] Yanhui Liu, Qing Huo Liu, and Zaiping Nie. 2009. Reducing the number of elements in the synthesis of shaped-beam patterns by the forward-backward matrix pencil method. *IEEE Transactions on Antennas and Propagation* 58, 2 (2009), 604–608.

[43] Yanhui Liu, Zaiping Nie, and Qing Huo Liu. 2008. Reducing the number of elements in a linear antenna array by the matrix pencil method. *IEEE Transactions on Antennas and Propagation* 56, 9 (2008), 2955–2962.

[44] Jaime Xavier Lopez. 2011. *Inverse synthetic aperture radar imaging theory and applications*. The University of Texas-Pan American.

[45] Yin Luo, Fengjun Zhao, Ning Li, and Heng Zhang. 2018. A modified cartesian factorized back-projection algorithm for highly squint spotlight synthetic aperture radar imaging. *IEEE Geoscience and Remote Sensing Letters* 16, 6 (2018), 902–906.

[46] Omar Mehanna, Kejun Huang, Balasubramanian Gopalakrishnan, Aritra Konar, and Nicholas D Sidiropoulos. 2014. Feasible point pursuit and successive approximation of non-convex QCQPs. *IEEE Signal Processing Letters* 22, 7 (2014), 804–808.

[47] Alberto Moreira, Josef Mittermayer, and Rolf Scheiber. 1996. Extended chirp scaling algorithm for air-and spaceborne SAR data processing in stripmap and ScanSAR imaging modes. *IEEE Transactions on Geoscience and Remote Sensing* 34, 5 (1996), 1123–1136.

[48] Alberto Moreira, Pau Prats-Iraola, Marwan Younis, Gerhard Krieger, Irena Hajnsek, and Konstantinos P Papathanassiou. 2013. A tutorial on synthetic aperture radar. *IEEE Geoscience and Remote Sensing Magazine* 1, 1 (2013), 6–43.

[49] JM Munoz-Ferreras, Felix Perez-Martinez, and Mihai Datcu. 2010. Generalisation of inverse synthetic aperture radar autofocusing methods based on the minimisation of the Renyi entropy. *IET Radar, Sonar & Navigation* 4, 4 (2010), 586–594.

[50] Siew Eng Nai, Wee Ser, Zhu Liang Yu, and Huawei Chen. 2010. Beampattern synthesis for linear and planar arrays with antenna selection by convex optimization. *IEEE Transactions on Antennas and Propagation* 58, 12 (2010), 3923–3930.

[51] Yijiang Nan, Xiaojing Huang, and Y Jay Guo. 2022. A Panoramic Synthetic Aperture Radar. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), 1–13.

[52] Shivakumar Ramakrishnan, Vincent Demarcus, Jerome Le Ny, Neal Patwari, and Joel Gussy. 2002. Synthetic aperture radar imaging using spectral estimation techniques. *Advanced Signal Processing* (2002), 65.

[53] R Keith Raney, Hartmut Runge, Richard Bamler, Ian G Cumming, and Frank H Wong. 1994. Precision SAR processing using chirp scaling. *IEEE Transactions on Geoscience and Remote Sensing* 32, 4 (1994), 786–799.

[54] Sandeep Rao, Adeel Ahmad, June Chul Roh, and S Bharadwa. 2017. 77GHz single chip radar sensor enables automotive body and chassis applications. *Texas Instruments* (2017).

[55] Mark A Richards et al. 2005. *Fundamentals of radar signal processing.* Vol. 1. Mcgraw-Hill New York.

[56] Hermann Rohling. 1983. Radar CFAR thresholding in clutter and multiple target situations. *IEEE Trans. Aerospace Electron. Systems* 4 (1983), 608–621.

[57] Ralph Schmidt. 1986. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation* 34, 3 (1986), 276–280.

[58] Krishnasamy T Selvan and Ramakrishna Janaswamy. 2017. Fraunhofer and Fresnel Distances: Unified derivation for aperture antennas. *IEEE Antennas and Propagation Magazine* 59, 4 (2017), 12–15.

[59] B.D. Steinberg. 1988. Microwave imaging of aircraft. *Proc. IEEE* 76, 12, 1578–1592.

[60] Lars MH Ulander, Hans Hellsten, and Gunnar Stenstrom. 2003. Synthetic-aperture radar processing using fast factorized back-projection. *IEEE Trans. Aerospace Electron. Systems* 39, 3 (2003), 760–776.

[61] Harry L Van Trees. 2002. *Optimum array processing: Part IV of detection, estimation, and modulation theory*. John Wiley & Sons.

[62] Martin Vossiek, Alexander Urban, Stephan Max, and Peter Gulden. 2007. Inverse synthetic aperture secondary radar concept for precise wireless positioning. *IEEE Transactions on Microwave Theory and Techniques* 55, 11 (2007), 2447–2453.

[63] Dian Wang, Fazhi An, and Seung Wook Yoon. 2018. Advanced EWLB (Embedded Wafer Level Ball Grid Array) Solutions for Mmwave Applications. In *2018 International Wafer Level Packaging Conference (IWLPC)*. 1–6.

[64] Qixin Wang, Wei-Peng Chen, Rong Zheng, Kihwal Lee, and Lui Sha. 2003. Acoustic target tracking using tiny wireless sensor devices. In *Information Processing in Sensor Networks*. Springer, 642–657.

[65] Mati Wax and Thomas Kailath. 1985. Detection of signals by information theoretic criteria. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 33, 2 (1985), 387–392.

[66] Mati Wax and Ilan Ziskind. 1989. Detection of the number of coherent signals by the MDL principle. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37, 8 (1989), 1190–1196.

[67] CA Willey. 1985. Synthetic aperture radars–a paradigm for technology evolution. *IEEE Trans. Aerospace Electron. Systems* 21 (1985), 440–443.

[68] Yuteng Xiao, Jihang Yin, Honggang Qi, Hongsheng Yin, and Gang Hua. 2017. MVDR algorithm based on estimated diagonal loading for beamforming. *Mathematical Problems in Engineering* 2017 (2017).

[69] Gang Xu, Lei Yang, Lei Zhang, Ya-Chao Li, and Meng-Dao Xing. 2011. Weighted minimum entropy autofocus algorithm for ISAR imaging. *Journal of Electronics & Information Technology* 33, 8 (2011), 1809–1815.

[70] Hui Xu, Guodong Qin, and Lina Zhang. 2007. Monopulse radar 3-D imaging and application in terminal guidance radar. In *MIPPR 2007: Automatic Target Recognition and Image Analysis; and Multispectral Image Acquisition*, Vol. 6786. SPIE, 1453–1459.

[71] Jia Xu, Yingning Peng, and Xiang-Gen Xia. 2004. Parametric autofocus of SAR imaging-inherent accuracy limitations and realization. *IEEE Transactions on Geoscience and Remote Sensing* 42, 11 (2004), 2397–2411.

[72] Muhammet Emin Yanik and Murat Torlak. 2019. Near-Field MIMO-SAR Millimeter-Wave Imaging With Sparsely Sampled Aperture Data. *IEEE Access* 7 (2019), 31801–31819.

[73] DaZhi Zeng, Tao Zeng, Cheng Hu, and Teng Long. 2006. Back-projection algorithm characteristic analysis in forward-looking bistatic SAR. In *2006 CIE International Conference on Radar*. IEEE, 1–4.

[74] BingChen Zhang, Wen Hong, and YiRong Wu. 2012. Sparse microwave imaging: Principles and applications. *Science China Information Sciences* 55, 8 (2012), 1722–1754.

[75] Jun Zhang, Guisheng Liao, Shengqi Zhu, Jingwei Xu, and Feiyang Liu. 2018. Wavenumber-domain autofocus algorithm for helicopter-borne rotating synthetic aperture radar. *IET Signal Processing* 12, 3 (2018), 294–300.

[76] Lei Zhang, Hao-lin Li, Zhi-jun Qiao, and Zhi-wei Xu. 2014. A fast BP algorithm with wavenumber spectrum fusion for high-resolution spotlight SAR imaging. *IEEE Geoscience and Remote Sensing Letters* 11, 9 (2014), 1460–1464.

[77] X Zhang, Z Xu, L Xu, and D Xu. 2011. Trilinear decomposition-based transmit angle and receive angle estimation for multiple-input multiple-output radar. *IET Radar, Sonar & Navigation* 5, 6 (2011), 626–631.

[78] Yubin Zhao, Alexander Yarovoy, and Francesco Fioranelli. 2022. Angle-insensitive human motion and posture recognition based on 4D imaging radar and deep learning classifiers. *IEEE Sensors Journal* 22, 12 (2022), 12173–12182.

[79] Rongqiang Zhu, Jianxiong Zhou, Ge Jiang, and Qiang Fu. 2017. Range migration algorithm for near-field MIMO-SAR imaging. *IEEE Geoscience and Remote Sensing Letters* 14, 12 (2017), 2280–2284.