MULTIDIMENSIONAL RADAR SIGNAL PROCESSING BASED ON SPARSE FOURIER TRANSFORMS

By

SHAOGANG WANG

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

written under the direction of

Athina P. Petropulu and Vishal M. Patel

and approved by

____

____

____

____

New Brunswick, New Jersey

May, 2019

ABSTRACT OF THE DISSERTATION

Multidimensional Radar Signal Processing Based on Sparse Fourier Transforms

By SHAOGANG WANG

Dissertation Director:

Athina P. Petropulu and Vishal M. Patel

The conventional radar signal processing typically employs the Fast Fourier Transform (FFT) to detect targets and identify their parameters. The sample and computational complexity of the N-point FFT are $O(N)$ and $O(N \log N)$, respectively. In modern Digital Beamforming (DBF) and Multiple-Input Multiple-Output (MIMO) radars, N is large due to the increased dimensions of processing (i.e., range, Doppler and angle) and the need for high radar resolution in each dimension. Hence, the FFT-based radar processing is still challenging for DBF/MIMO radars of constrained computation resources, such as the state-of-the-art automotive radars.

Sparse Fourier Transform (SFT) is a family of low-complexity algorithms that implement Discrete Fourier Transform (DFT) for frequency-domain sparse signals. State-of-the-art SFT algorithms achieve sample complexity of $O(K)$ and computational complexity of $O(K \log K)$ for a K-sparse signal. When $K << N$, the sample and computational savings of SFT are significant as compared with that of the FFT. In radar applications, the number of radar targets is usually much smaller than the number of resolution cells in the multidimensional frequency domain, i.e., the radar signal is sparse in the frequency domain; thus, it is tempting to replace the FFT with SFT to reduce sample and computational complexity of signal

processing. However, applying SFT in radar signal processing is not trivial for the following reasons:

- Most existing SFT algorithms are designed for one-dimensional, ideal signals, which are noiseless and contain on-grid frequencies; those SFT algorithms are not practical for radar applications as the radar signals are multidimensional, noisy, and contain off-grid frequencies.
- The signal processing schemes of different radar architectures need to be properly designed to support the application of SFT. When the radar signal is not naturally sparse, proper preprocessing is required to sparsify the signal.
- The application of SFT in radar signal processing involves tradeoffs between sample/computational savings and radar detection performance. Such tradeoff needs to be characterized and the design of various parameters of SFT algorithms need to be investigated to achieve the optimal tradeoff.

This dissertation aims to formulate SFT-based frameworks for radar signal processing and address the above issues by proposing two new SFT algorithms, and adapting them to DBF and MIMO radars. The proposed SFT algorithms are the Robust Sparse Fourier Transform (RSFT) and MultidimensionAl Random Slice based Sparse Fourier Transform (MARS-SFT).

RSFT extends the basic SFT algorithm to multidimensional, noisy signals that contain off-grid frequencies. By incorporating Neyman-Pearson detection, frequency detection in the RSFT does not require knowledge of the exact sparsity of the signal and is robust to noise. The computational savings versus detection performance tradeoff is investigated, and the optimal threshold is found by solving a constrained optimization problem. The application of RSFT in DBF and MIMO radars is investigated. A uniform processing scheme based on RSFT is proposed for MIMO radar that employs fast-time coded and slow-time coded pulse-compression waveform.

Although RSFT-based radar signal processing achieves significant computational savings as compared to FFT-based processing, it does not offer sample complexity savings.

To reduce sample as well as computational complexity, we propose MARS-SFT, a sparse Fourier transform for multidimensional, frequency-domain sparse signals, inspired by the idea of the Fourier projection-slice theorem. MARS-SFT identifies frequencies by operating on one-dimensional slices of the discrete-time domain data, taken along specially designed lines; those lines are parametrized by slopes that are randomly generated from a set at runtime. The DFTs of the data slices represent DFT projections onto the lines along which the slices were taken. On designing the line lengths and slopes so that they allow for orthogonal and uniform frequency projections, the multidimensional frequencies can be recovered from their projections with low sample and computational complexity. To apply MARS-SFT to real-world radar signal processing, which involves noisy signals and off-grid frequencies, we propose the robust MARS-SFT, and demonstrate its performance in digital beamforming automotive radar signal processing. In that context, the robust MARS-SFT is used to identify range, velocity and angular parameters of targets with low sample and computational complexity.

Finally, we propose a new automotive radar architecture. Such radar achieves high resolution in range, range rate, azimuth and elevation angles of extended targets by leveraging two orthogonally-placed digital beamforming linear arrays of a few channels. A deep learning based beam matching method is developed for the proposed radar to address the beam association challenges. In sparse scenarios, the proposed robust MARS-SFT can be employed in the beamforming, and range-Doppler imaging procedures to reduce computation.

Acknowledgments

The past four and half years of my graduate study toward the Ph.D. is the greatest adventure in the journey of my life. Looking back, I couldn't have possibly made it without many helping hands.

First of all, I'd like to express my deepest gratitude to my advisor, Prof. Athina P. Petropulu, not only for her continuous guidance in my academic exploration but also for her enormous inspiration to pursue excellence. Prof. Petropulu has been set up a role model of mine by her constant passion in research, unexhausted energy, and great dedication to students, the ECE department at Rutgers, and the broad community in signal processing research.

Next, I am thankful to Prof. Vishal M. Patel for his generous support and invaluable advice. Prof. Patel's research in artificial intelligence has been inspired me to bridge conventional radar signal processing to the state-of-the-art machine learning. I have been enjoyed many one-to-one talks with Prof. Patel, as well as happy hours with friends from his lab.

Moreover, I'd like to thank Profs. Waheed Bajwa, Laleh Najafizadeh, and Yimin Zhang (from Temple University) for their precious time to serve as members of my dissertation defense committee. I also thank Profs. Zoran Gajic, Kristin Dana, Emina Soljanin, and Wade Trappe to serve as the committee members for my qualifying exam.

Also, I'd like to thank Mr. Zhenya Hu from Shanghai Academy of Spaceflight Technology, for his encouragement and support to my pursuing of the Ph.D. study. I also thank Prof. Predrag Spasojevic for accepting my application at Rutgers and offering many helps during the initial stage of my oversea life.

Last but not least, I am deeply grateful to my family, including my beloved parents, grandparents, my wife and life-long companion Zhenzhen, and son Wenwen, for the endless love that they bring to me. I shall not take this for granted.

Dedication

To Zhenzhen and Wenwen

Table of Contents

Abstract ii

Acknowledgments v

Dedication vi

List of Tables xii

List of Figures xiii

1. Introduction 1

1.1. Radar Basics 1

1.2. FMCW Radar 4

1.3. Digital Beamforming Radar 7

1.4. MIMO Radar 9

1.5. Sparse Fourier Transform 13

1.5.1. Simple and Practical SFT 19

1.5.2. SO-SFT 23

1.6. Contribution of the Dissertation 26

1.6.1. SFT Algorithms that Address Real-World Multidimensional Signals . . 26

1.6.2. SFT-based Radar Signal Processing Frameworks for DBF and MIMO Radars 27

1.6.3. Two-dimensional DBF Radar with Orthogonal Linear Arrays Based on SFT and Deep Learning 28

1.7. Outline of the Dissertation 28

1.8. Notation 29

2. Robust Sparse Fourier Transform 30

2.1. Introduction 30

2.2. Signal Model and Overview of the RSFT Algorithm 32

2.3. Leakage Suppression of Off-grid Frequencies 33

2.4. NP Detection in the RSFT 36

2.4.1. The First Stage Detection 37

2.4.2. The Second Stage Detection 41

2.4.3. The Optimization Problem 44

2.5. Computational Complexity Analysis 46

2.6. Multidimensional Extensions 48

2.7. Numerical Results 50

2.7.1. Improvement of Signal Sparsity by Pre-permutation Windowing 52

2.7.2. The Effect of Signal Sparsity and SNR of the Co-existing Frequencies on Detection Performance 54

2.7.3. The Effect of Frequency Locations on Detection Performance 55

2.7.4. The Trade-off between Complexity and Detection Performance 56

2.7.5. Comparison of RSFT and SFT 56

2.7.6. The Variance and Its Upper Bound for $[\bar{\mathbf{a}}]_i$ 58

2.7.7. SNR Loss due to RSFT as Compared with the FFT-based Method . . 60

2.7.8. Characterize the Detection Performance for K Targets 62

2.8. Summary 63

3. RSFT-based DBF and MIMO Radar Signal Processing 64

3.1. RSFT-based DBF Radar 65

3.1.1. Conventional Signal Processing of DBF Radar 65

3.1.2. RSFT-based DBF Radar Signal Processing 67

3.2. RSFT-based MIMO Radar 70

3.2.1. Signal Model 73

3.2.2. Slow-time and Fast-time Coded Waveforms Processing 74

3.2.3. RSFT-based Collocated MIMO Radar Signal Processing 76

3.2.4. Range Pre-processing 78

3.2.5. A Uniform Processing Structure for MIMO-RSFT Radar 78

3.2.6. Complexity Analysis for MIMO-RSFT Radar 78

3.2.7. Doppler Processing for the Fast-time Coded Waveform 79

3.2.8. Simulation 79

3.3. Summary 82

4. Multidimensional Random Slice Based Sparse Fourier Transform 83

4.1. Introduction 83

4.2. Signal Model 87

4.3. Overview of MARS-SFT 87

4.4. Convergence of MARS-SFT 92

4.5. Multidimensional Extension and Complexity Analysis 95

4.5.1. Multidimensional Extension 95

4.5.2. Complexity analysis 95

4.6. Numerical Results 96

4.6.1. Comparison between MARS-SFT and SO-SFT 96

4.6.2. Comparison between MARS-SFT and the SFT of [1, 2] 97

4.6.3. Line Slope of MARS-SFT 98

4.6.4. Convergence of MARS-SFT 99

4.7. Summary 100

5. Robust MARS-SFT and Its Application in DBF Radar 102

5.1. Introduction 102

5.2. Signal Model 103

5.3. Window Design 104

5.4. Voting-Based Frequency Decoding 105

5.5. Lower Bound of the Probability of Correct Localization and Convergence of Robust MARS-SFT 105

5.6. Numerical Results 107

5.6.1. Effect of Windowing on Frequency Localization of Robust MARS-SFT 107

5.6.2. Effect of Voting on Frequency Localization of Robust MARS-SFT . . . 107

5.6.3. Effect of SNR and Sparsity Level on the Convergence of Robust MARS-SFT 109

5.6.4. Amplitude and Overall Error of the Robust MARS-SFT 110

5.7. MARS-SFT Based DBF Radar 111

5.8. Summary 113

6. Two-Dimensional DBF Automotive Radar with Orthogonal Linear Arrays 114

6.1. Introduction 114

6.2. System Model 118

6.2.1. Target Model 118

6.2.2. Radar Architecture 119

6.2.3. Signal Model 119

6.3. Signal and Data Processing 121

6.3.1. Overview 121

6.3.2. SFT-based Pre-processing 121

6.3.3. Beam Matching via Deep Learning 122

6.3.4. Resource Management 124

6.4. Simulation 126

6.4.1. RDI Patch Matching 126

6.4.2. Target Reconstruction 128

6.4.3. Resource Management 130

6.5. Summary 131

7. Conclusions and Future Research Directions 132

7.1. Conclusions 132

7.2. Future Research Directions 134

7.2.1. Clutter and Interference Suppression for SFT-based Radar Processing 134

7.2.2. Deterministic SFT-based Method for Radar Signal Processing 134

7.2.3. Characterization of Detection Performance for K frequencies in RSFT 135

Appendices 136

A. Proof of Lemma 1 137

B. Proof of Property 3 137

C. Proof of Property 4 137

D. Proof of Lemma 2 138

E. Proof of Lemma 3 138

F. Proof of Remark 2 139

G. Proof of Lemma 4 139

H. Proof of Lemma 5 139

I. Proof of Theorem 2 141

J. Proof of Theorem 3 141

K. Proof of Lemma 6 142

L. Proof of (5.3) 143

M. Detection Performance of the FFT-based Bartlett Method 145

N. Pseudo-code of robust MARS-SFT 147

References 149

List of Tables

2.1. Computational Complexity of RSFT 47

2.2. Computational Complexity of the Bartlett Method 47

2.3. Effect of Pre-permutation Window. 54

3.1. DBF radar parameters 69

3.2. Target Parameters 69

3.3. DBF radar parameters 80

3.4. Target Parameters 80

5.1. Radar Parameters 113

6.1. Radar Parameters 127

List of Figures

1.1. FMCW waveform. The signal frequency changes linearly in time, with a repetition interval T. A coherent processing interval (CPI) contains M repetitions. The received signal is a delayed version of the transmitted signal. . . 5

1.2. DBF radar system architecture. A broad beam pattern is formed with an omnidirectional transmit antenna, while multiple narrow beams are formed simultaneously by the receiving array. Each receiving channel is mixed with a coupled signal from the transmitter to demodulate (de-chirp) the FMCW signal, before AD conversion. The digitized received signal is processed by a Digital Signal Processor (DSP), while the transmit waveform is generated by a Direct Digital Synthesizer (DDS). 8

1.3. DOA estimation with a ULA. 9

1.4. Conventional FFT-based processing scheme for the DBF FMCW radar. . . . 10

1.5. Collocated MIMO Radar System with ULA. Each transmit antenna transmits an orthogonal waveform, which is generated by a DDS. The orthogonality of the waveforms results into an omni-directional transmit beam pattern, while multiple narrow beams are formed simultaneously by beamforming in the DSP. 11

1.6. Virtual array. $M_t = 3, M_r = 2, d_r = M_t d_t$. 13

1.7. Time domain subsampling and frequency domain aliasing. (a) The real part of a time domain single tone, i.e., (1.34) with $N = 15, f = 11, a = 1$. (b) 15-point DFT of (1.34). (c) 5-subsampled version of (1.34), i.e., $x_3(n), n = 0, 1, 2$. (d) $\hat{x}_3(k), k = 0, 1, 2$. (e) 3-subsampled version of (1.34), i.e., $x_5(n), n = 0, 1, \cdots, 4$. (f) $\hat{x}_5(k), k = 0, 1, \cdots, 4$. 16

1.8. SFT high level illustration. (a) One-shot based SFT. (b) Iterative-based SFT. 18

1.9. A frequency domain flat window. $N = 1024, B = 16$. (a) Time domain. (b) Frequency domain. Shifted by $N/2$. 21

1.10. Illustration of simple and practical SFT. The left part of the figure shows the main steps of the localization loop of the SFT algorithm, while the right part shows the signal representations in the discrete frequency domain due to each step. 22

1.11. Illustration of SO-SFT [3]. Frequencies are recovered iteratively by alternatively applying DFT along rows and columns of the data matrix. 25

1.12. Deadlock situations for SO-SFT 25

2.1. The effect of pre-permutation windowing on sparsity of the signal in the discrete frequency domain. The signal contains two significant frequency components, one of which is $35dB$ stronger than the other; $SNR_{min} = -8dB$. A Dolph-Chebyshev window with $77dB$ PSR is applied. Windowed signal after permutation appears sparser in the frequency domain as compared to the permuted signal without windowing. The spectrum is computed via 1024-point DFT. (a) Spectrum of signal without windowing. (b) Spectrum of signal without windowing after permutation with $\sigma_s = 85$. (c) Spectrum of windowed signal. (d) Spectrum of windowed signal after permutation with $\sigma_s = 85$. . . . 35

2.2. Pre-permutation window. (a) 1024-point DFT of a Dolph-Chebyshev window with $77dB$ PSR. (b) Main-lobe broadening parameter η_m versus the PSR for Dolph-Chebyshev windows with 1024-points DFT. 36

2.3. Windowing, permutation and aliasing. The frequency representation of the signal from pre-permutation windowing to aliasing is presented. Only one significant frequency is shown for conciseness. 39

2.4. Mapping and reverse mapping. Due to the different permutations, a significant frequency may be mapped into different locations in each iteration of the first stage detection. The detected frequencies, including the false alarms in the first stage, are reverse mapped to the original discrete frequency set. The true location of the significant frequency as well as ambiguous frequencies are obtained. The occurrence on the true location grows steadily during accumulation, provided that the SNR is high enough, and thus the true location can be recovered in the second stage of detection with proper thresholding. However, false alarms may also occur in the second stage detection, due to both ambiguous frequencies and false alarms from the first stage of detection. 43

2.5. Comparison of complexity. $N = 1024, T = 50, \eta_m = 2, \eta_p = 1$, $B \in \{8, 16, 32, 64, 128, 256, 512, 1024\}$. 48

2.6. Permutation and aliasing in 2-D. (a) Original 2-D data forms a 4×8 matrix. (b) Permutation in $x-$dimension, $\sigma_x = 3$. (c) Permutation in $y-$dimension, $\sigma_y = 3$. After permutation, data is divided into four 2×4 sub-matrices. (d) Aliasing by adding sub-matrices from (c). 49

2.7. The values of $\alpha(p, \sigma_s, \omega_m), \beta(\sigma_s)$ with respect to all possible values of the permutation parameter σ_s. 51

2.8. Solving for P_d^* via a brute force searching of $\gamma \in (0, 0.1)$, with a step size $\Delta\gamma = 5 \times 10^{-4}$. 52

2.9. The two stages of detection. $K = K_{max} = 16$. (a) Signal spectrum after pre-permutation windowing. The magenta dot denotes the amplitude of ω_m. (b) First stage detection. (c) Second stage detection. Data and threshold are normalized by T. (d) ROC curves of the two detection stages. 53

2.10. The effect of sparsity and SNR of co-existing frequencies on detection performance. (a) Changing K for $D = 30dB, \eta_p = 1/\bar{P}_d(\omega_m)$. (b) Changing D for $K = K_{max} = 10$; $\eta_p = 1$ when $D = 0dB$, $\eta_p = 1/\bar{P}_d(\omega_m)$ when $D = 30dB$ and $50dB$. (c) ROC curves for various K_{max} values. (d) The second stage detection when $K = 4, K_{max} = 16$. 55

2.11. The effect of frequency locations on detection performance. (a) The fluctuation of P_d^* due to different locations of the frequency is mainly caused by the off-grid loss. (b) ROC curves with respect to different locations of the frequency. 56

2.12. The effect of the reduced length B on detection performance when $K_{max} = 4$. A large B provides a better detection performance. 57

2.13. Comparison of localization error for RSFT and SFT. $B = 256, C_{md} = C_{fa} = 1$. The localization error may be decomposed into averaged ℓ_1 distance, cost of miss detection and cost of false alarm. (a) The impact of p_1 on l_{err}^{SFT} when $p_2 = 20, L = 6$. (b) The impact of p_2 on l_{err}^{SFT} when $p_1 = 40, L = 6$. (c) The impact of L on l_{err}^{RSFT} when $p_1 = 40, p_2 = 20$. (d) The impact of L on l_{err}^{SFT} when $p_1 = 40, p_2 = 20$. 59

2.14. Normalized approximation error between the actual variance $\sigma_{a1}^2(\omega_m)$ and its upper bound, versus number of iterations. 60

2.15. Worst case SNR and complexity tradeoff. $N = 1024, T = 50, D = 0dB, B = \{8, 16, 32, 64, 128, 256, 512, 1024\}, P_d = 0.9, P_{fa} = 10^{-6}, K = \{5, 10, 100\}, \omega_m = \Delta\omega_N/2$. The red dot shows the performance of the Bartlett method. 61

2.16. Recall and precision versus SNR for different K. $P_{fa} = 10^{-2}, D = 0dB, T = 50, N = 1024, B = 256, K = K_{max}$. 63

3.1. Conventional FFT-based processing scheme for FMCW DBF radars. 67

3.2. RSFT-based processing scheme for the DBF radar. 68

3.3. Target reconstruction via 3-D simple and practical SFT and RSFT; all targets have the same SNR. Both simple and practical SFT and RSFT based methods can reconstruct all the targets, while simple and practical SFT has better resolution. 71

3.4. Target reconstruction via 3-D simple and practical SFT and RSFT with different SNR for the 4 Targets. Simple and practical SFT based processing recovers the side-lobes of the stronger targets, while the RSFT-based method only recovers the main-lobes of targets. 71

3.5. Conventional MIMO radar signal processing for slow-time coded waveforms. The red arrows indicate the dimension of related processing. (a) Overview of MIMO radar processing for slow-time coded waveforms. WD, PC, transmit and receive beamforming are processed on the burst basis. For better detection performance, a non-coherent accumulation over T/L bursts is applied. (b) Range pre-processing for slow-time coded waveforms. 76

3.6. Conventional MIMO radar signal processing for fast-time coded waveforms. (a) Overview of MIMO radar processing for fast-time coded waveforms. The matched filtering, transmit and receive beamforming are applied in each pulse. A non-coherent accumulation over T pulses is applied. (b) Matched filtering and range pre-processing for fast-time coded waveforms. 77

3.7. MIMO-RSFT radar signal processing for both slow-time and fast-time coding schemes. In each iteration, the input for slow-time and fast-time coding is a burst of pulses and a single pulse, respectively. 79

3.8. Target reconstruction via 3-D FFT, RSFT and SFT. Comparing to the 3D-FFT, 3D-RSFT can recover the targets exactly, while the recovery via the 3D-SFT results into many false alarms due to the leakage from Target 1. (a) FFT vs RSFT. (b) FFT vs SFT. 81

3.9. Complexity Ration, simple and practical FFT over RSFT. $U = 2^{24}, V = \{2^{10}, 2^{14}, 2^{18}, 2^{22}\}, \eta_m = 4, \eta_p = 1$. Other parameters are the same as in Table 3.1. (a) slow-time coding. $L = N, T = 4L$. (b) fast-time coding. 82

4.1. Demonstration of projection of 2-D frequencies onto 1-D. The colored blocks mark significant frequencies. The projection onto the column or the row causes collisions, while the projection onto the diagonal creates 1-sparse bins. 88

4.2. An orthogonal pair of time and frequency domain lines. $N_0 = 16, N_1 = 8, L = 16, \boldsymbol{\alpha} = [1, 3]^T, \boldsymbol{\tau} = [0, 0]^T$. 91

4.3. Number of iterations of MARS-SFT versus K/L. 94

4.4. Comparison between MARS-SFT and SO-SFT. (a) Probability of exact recovery versus sparsity level, K. (b) Ratio of samples needed for exact recovery versus K. 97

4.5. Comparison between MARS-SFT and SFT of [1, 2]. The robust MARS-SFT is adopted in noisy cases. (a) Localization success rate versus K in noiseless cases. (b) Localization success rate versus K in noisy cases. Frequencies are on the grid. 98

4.6. Effect of the line slope to the MARS-SFT. (a) Number of iterations of exact recovery versus sparsity. (b) Effect of the size of the slope parameter set ($|\mathcal{A}'|$) to the exact recovery probability. 99

4.7. Number of iterations of MARS-SFT. (a) Number of iterations versus sparsity. (b) Ratio of samples versus sparsity. 100

4.8. Original (red circles) and recovered frequencies (black dots) using one iteration of MARS-SFT. $N_0 = 32, N_1 = 31, K = 20N_0 = 640$. 101

5.1. Frequency localization success rate of the first iteration of robust MARS-SFT versus window PSR. The Dolph-Chebyshev windows with various PSR is applied. $(n_s, n_d) = (3, 2)$. 108

5.2. 2-D frequency recovery with different windows. $K' = 10$, $\sigma_n = 1, a_{min} = a_{max}$, $SNR_{max} = 30dB$, $(n_s, n_d) = (3, 2), T = 30$. The ground truth represents the discrete frequency clusters of $\mathbb{S}$. The PSR of the window equals $45dB$ in (a) and $70dB$ in (b). 108

5.3. Effect of voting on 2-D frequency recovery. $K' = 10$, $\sigma_n = 1, a_{min} = a_{max}$, $SNR_{max} = 30dB$. $T = 30$. Dolph-Chebyshev windows with $\rho_w = 70dB$ is applied. (a) $(n_d, n_s) = (1, 1)$. (b) $(n_d, n_s) = (3, 1)$. (c) $(n_d, n_s) = (3, 3)$. . . . 109

5.4. Effect of SNR and sparsity level on the convergence of robust MARS-SFT. . 110

5.5. Amplitude and overall error of the robust MARS-SFT. $(n_s, n_d) = (3, 2)$. Dolph-Chebyshev windows are applied for the off-grid cases, while the rectangular window is used in the on-grid cases. (a) Amplitude error. (b) Overall error. 111

5.6. Radar target reconstruction via FFT, MARS-SFT and RSFT. (a) Reconstruction of three targets. (b) Details of the frequency locations reconstructed for one of the three targets. 113

6.1. The proposed radar architecture. (a) Radar architecture. (b) Beam positions. The radar forms N_a, N_e ($N_a = N_e = 4$) beams in azimuth and elevation, respective. 120

6.2. Overview of signal and data processing. 121

6.3. The beam matching net. (a) Beam matching net. (b) Siamese net. 123

6.4. The proposed resource management algorithm. 126

6.5. Examples of similar and dissimilar range-Doppler patches. (a) Similar patch. (b) Dissimilar patch. 127

6.6. Clustering of feature vectors from the train and test set. (a) Train. (b) Test. . 128

6.7. Measurement and reconstruction from individual array. (a) Azimuth-range measurement from the azimuth array. (b) Reconstruction in X-Y. (c) Elevation-range measurement from the elevation array. (d) Reconstruction in X-Z. . . . 129

6.8. Reconstruction in the X-Y-Z coordinate. 130

6.9. Comparison of resource management algorithms. (a) Comparison of the proposed algorithm and the time-balanced algorithm. (b) Effect of the cost function. The unit of average and maximum delay is number of schedule intervals. $C = 3$. 131

Chapter 1

Introduction

In this chapter, we cover radar basics and provide the necessary background of the conventional Digital Beamforming (DBF) and collocated Multi-Input Multi-Output (MIMO) radar signal processing. Those state-of-the-art radar architectures enjoy better target parameter identifiability compared to conventional phased array radars as they provide more degrees of freedom in the angular domain. However, this is at the expense of more complex processing due to a larger amount of data. We thus are interested in reducing the sample and computational complexity of such radar signal processing by employing Sparse Fourier Transform (SFT) based approaches, aiming to enable the real-time processing with low-cost hardware.

SFT is a family of low-complexity algorithms for the implementation of the Discrete Fourier Transform (DFT) of signals that are sparse in the frequency domain. The SFT algorithms can be divided into one-dimensional (1-D) and multidimensional algorithms. Here, we provide a high level review of SFT. Specifically, we review simple and practical SFT [4] and the Sample-Optimal SFT (SO-SFT) [5] algorithms, which are examples of 1-D and multidimensional SFT algorithms, respectively. The two SFT algorithms also form the basis of our proposed SFT algorithms, i.e., Robust Sparse Fourier Transform (RSFT) and MultidimensionAl Random Slice based Sparse Fourier Transform (MARS-SFT), respectively.

The notations used throughout the dissertation are listed in Section 1.8.

1.1 Radar Basics

Radars are electrical devices that leverage electromagnetic waves to detect targets and estimate their parameters including range, range rate (radial velocity), and Direction of Arrival (DOA). To this end, radars use transmit antennas to transmit modulated waveforms, which propagate through space, and are reflected back from targets. The reflected waves are captured by the receive antennas, from which the information about the targets are extracted by the receiver chains and processing pipelines of radars.

To illustrate basic principles of range and range rate estimation, let us consider a monostatic pulse radar configuration, where the transmit and receive antennas are collocated. The transmit waveform is modeled in a complex form, expressed as [6]

$$x(t) \triangleq s(t)e^{j2\pi ft} = a(t)e^{j\phi(t)}e^{j2\pi ft},\ 0 \leq t \leq \tau, \tag{1.1}$$

where f, τ are the carrier frequency and the pulse width, respectively. The term $s(t) = a(t)e^{j\phi(t)}$ represents the baseband signal, and $a(t), \phi(t)$ are its amplitude and phase components, respectively. The simplest form of a pulse waveform is the rectangular pulse where $s(t) = 1$ within $[0, \tau]$ and 0, otherwise.

The simplest model of a target is a stationary point; the reflected signal from such target is modeled as

$$r(t) \triangleq \alpha x(t - t_0) + n(t) = \alpha s(t - t_0)e^{j2\pi ft} + n(t), \tag{1.2}$$

where α is a complex-valued constant, representing the amplitude attenuation and phase rotation of the transmit waveform due to antenna gains, target radar cross section (RCS), and propagation losses. The term $n(t)$ is the additive noise; in the simplest form, it is modeled as the zero-mean white Gaussian noise. The time delay t_0 is introduced by the round-trip distance between the antennas and the target. Assume that the one-way distance is R, and the waves travel at the speed of light of c, then, the target range can be calculated as

$$R = \frac{ct_0}{2}. \tag{1.3}$$

Note that the phase rotation term of the carrier sinusoid introduced by t_0, i.e., $e^{-j2\pi ft_0}$ is absorbed by α.

If the target moves at a radial velocity of v with respect to the radar, then, such movement introduces an additional time-delay of $\frac{2vt}{c}$, and the received signal is expressed as

$$r(t) \triangleq \alpha s(t - t_0 - \frac{2vt}{c})e^{j2\pi f(t - \frac{2vt}{c})} + n(t). \tag{1.4}$$

Such target movement yields a frequency shift of f_d, called the Doppler frequency, and

$$f_d \triangleq \frac{2v}{\lambda} = \frac{2vf}{c}, \tag{1.5}$$

where λ is the wavelength.

The effect of the time-delay to the baseband signal due to target movement can be neglected under the narrow-band assumption, i.e., [7]

$$\frac{2v\tau}{c} << \frac{1}{B}, \tag{1.6}$$

where τ, B are the pulse width and bandwidth of the baseband signal of $s(t)$, respectively.

Hence, (1.4) is simplified to

$$r(t) \approx \alpha s(t - t_0)e^{j2\pi f(t - \frac{2vt}{c})} + n(t) = \alpha s(t - t_0)e^{j2\pi(f - f_d)t} + n(t). \tag{1.7}$$

The estimation of t_0, f_d and hence R, v is ideally achieved by a matched filtering process, where the received signal is convolved with the conjugate of the transmit signal to maximize the output Signal-to-Noise Ratio (SNR), mathematically

$$\begin{aligned} m(u, \nu) &= \int_{-\infty}^{\infty} r(t)x^*(t - u)e^{-j2\pi\nu t}dt \\ &= \tilde{\alpha}A(u - t_0, \nu - f_d) + \int_{-\infty}^{\infty} n(t)s^*(t - u)e^{-j2\pi(f+\nu)t}dt, \end{aligned} \tag{1.8}$$

where $\tilde{\alpha} = ae^{j2\pi fu}$, and

$$A(u, \nu) \triangleq \int_{-\infty}^{\infty} s(t)s^*(t - u)e^{j2\pi\nu t}dt. \tag{1.9}$$

$|A(u, \nu)|$ is the ambiguity function [8] of the baseband waveform $s(t)$, which characterizes the most important properties of the waveform. The peak location of $|m(u, \nu)|$ corresponds to the estimate of (t_0, f_d) in the $u - \nu$ plane, which is the maximum likelihood estimation under the assumption that the noise is Gaussian [7]. When multiple targets present, multiple peaks can be detected from $|m(u, \nu)|$, with each peak location representing the time-delay and Doppler frequency of each target, respectively.

In real-world radar signal processing, the above two-dimensional (2-D) matched filtering process is usually implemented by two consecutive one-dimensional (1-D) matched filtering process to reduce computation. To achieve this, a pulse radar transmit M consecutive pulses with a repetition interval of T. The time within each repetition interval is called fast-time, while the time across repetition intervals is called slow-time. The matched filtering in the range domain is computed at the fast-time for each repetition interval, while the matched filtering in the Doppler domain is computed across the slow-time for M intervals at each range bin. Subsequently, a detection process is applied jointly in the range and Doppler domains.

1.2 FMCW Radar

While pulse radars are appealing in long range applications, Continuous Waveform (CW) radars play an important role in near range applications for the following reasons [9, 10]

- Monostatic pulse radars suffer from the blind range problem since transmission and reception cannot work simultaneously; CW radars do not have a blind range.
- CW radars are cost effective due to their simple structure and low power.

The CW radar without modulation cannot be used for ranging as its baseband signal is a constant. To enable the ranging capability, various waveforms can be used, among which, (linear) Frequency Modulation Continuous Waveform (FMCW), also known as the *chirp*, is the most widely used one. As will be explained in the following, FMCW achieves high range resolution with low cost.

The transmit FMCW with unit amplitude is modeled as [11]

$$x(t) \triangleq e^{j2\pi(ft+\frac{\rho t^2}{2})},\ 0 \leq t \leq T, \tag{1.10}$$

where $\rho = B/T$ is the chirp rate, and B, T are bandwidth and modulation period, respectively. It is clear that the frequency of $x(t)$ is changing linearly along time.

The noiseless received signal from a stationary point target is modeled as

$$r(t) \triangleq \alpha e^{j2\pi(f(t-t_0)+\frac{\rho(t-t_0)^2}{2})}, \tag{1.11}$$

where $\alpha \in \mathbb{C}$ represents the attenuation factor. (1.11) is a t_0-delayed version of the transmit signal as illustrated in Fig. 1.1. By absorbing the constant phase terms into the amplitude, (1.11) is simplified to

$$r(t) = \tilde{\alpha} e^{j2\pi(ft + \frac{\rho t^2}{2} - t_0 \rho t)}, \tag{1.12}$$

where $\tilde{\alpha} = \alpha e^{j2\pi(\frac{\rho t_0^2}{2} - f t_0)}$.

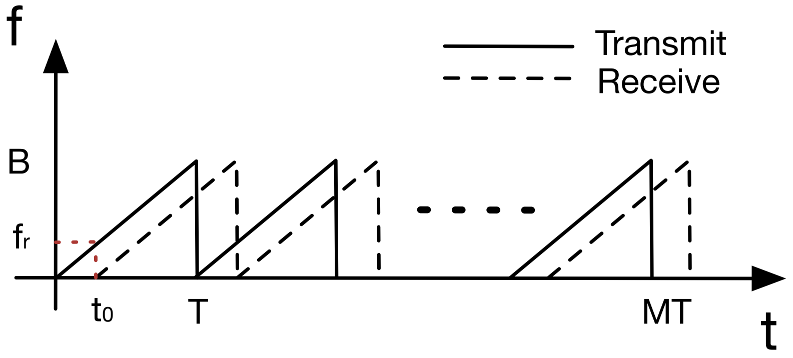


Figure 1.1: FMCW waveform. The signal frequency changes linearly in time, with a repetition interval T. A coherent processing interval (CPI) contains M repetitions. The received signal is a delayed version of the transmitted signal.

To demodulate the chirp, we multiply the received signal with the conjugate of the transmit signal, and the demodulated signal is expressed as

$$b(t) \triangleq r(t)x^*(t) = \tilde{\alpha} e^{j2\pi t_0 \rho t} = \tilde{\alpha} e^{j2\pi f_r t}. \tag{1.13}$$

This is a sinusoid with frequency $f_r = \rho t_0$; such frequency is called the beat frequency [11], which can be trivially evaluated by the Fourier transform (FT), subsequently, the target range can be deduced from the beat frequency by

$$R = \frac{c f_r}{2\rho}. \tag{1.14}$$

The above demodulation-FT process is the matched filtering of FMCW in the range domain. The demodulation process is implemented in the Radio Frequency (RF) front-end by mixing the received signal with the coupled transmit signal followed by a low-pass filter, while the FT is implemented digitally after sampling the demodulated signal. In near range applications, the maximum time-delay, t_{max}, introduced by the maximum target range, R_{max}, is much shorter than the modulation period, i.e., $t_{max} << T$. As a result, the bandwidth of the beat frequency, $B_b = \rho t_{max}$, is significantly smaller than the signal bandwidth B. Hence, low speed Analog-to-Digital Converters (ADCs) can be used to sample the demodulated signal, and the beat frequencies can be computed by the Fast Fourier Transform (FFT) with low-cost Digital Signal Processors (DSPs) or Field Programmable Gate Arrays (FPGAs).

The range resolution of FMCW is determined by the signal bandwidth B. To understand this, let us rewrite (1.14) as follows

$$\Delta R = \frac{c\Delta f_r}{2\rho}, \tag{1.15}$$

where $\Delta R, \Delta f_r$ denote the range resolution and beat frequency resolution, respectively. Since beat frequencies are evaluated over the modulation period T, we have

$$\Delta f_r = \frac{1}{T}, \tag{1.16}$$

therefore

$$\Delta R = \frac{c}{2\rho T} = \frac{c}{2B}. \tag{1.17}$$

Hence, conventional FMCW radars could achieve high range resolution with low-cost hardware. For instance, with $B = 1GHz$, the range resolution equals $0.15m$. A $50MHz$ ADC is sufficient for $T = 100us$ and $R_{max} = 300m$, since $B_b = 40MHz$. This makes FMCW radars be popular in automotive applications, in which the FMCW radar is adopted as one of primary sensors in automotive driver assistance systems (ADAS) and self-driving systems.

To measure the radial velocity of a moving target, similar to pulse radars, FMCW radars transmit M consecutive chirps as illustrated in Fig. 1.1; the noiseless demodulated signal from the m-th, $m \in [M]$, pulse is expressed as [11]

$$b(t, m) \triangleq \tilde{a}e^{j2\pi(f_r+f_d)(t-mT)}, \tag{1.18}$$

from which the Doppler frequency, f_d, is evaluated across the slow time for each range bin via FT. In fact, the ranges and Doppler frequencies of targets are usually evaluated via a 2-D FFT. Notably, when Doppler presents, the range and Doppler of the FMCW radar is coupled as shown in (1.18). In automotive radar scenarios, such range-Doppler coupling can be usually neglected since the Doppler shift is usually smaller than the resolution of the beat frequency, i.e., $f_d < \Delta f_r$.

1.3 Digital Beamforming Radar

In the above sections, we have discussed range and range rate estimation in radars. Another important target parameter estimation problem is the DOA estimation. The DOA of targets can be measured at azimuth and (or) elevation depending on the configuration of antennas. Conceptually, the DOA estimation is to find the bearing direction of the beam, where the output yields the maximum SNR. Modern radar systems use a set of arrays to transmit and receive waveform, which allows the radar form beam patterns that point to different directions without physically steering the antennas. This can be achieved by the phased array and DBF approaches [12, 13]. The former electrically steers different beams in a time-duplex manner by changing the phases of each antenna during transmitting and receiving, while the latter simultaneously forms multiple reception beams in the digital signal processor (DSP). The advantage of the latter is that the DBF radar could see targets everywhere within its field-of-view (FOV) at anytime without (physical) beam steering [14].

Let us consider a typical automotive DBF radar configuration illustrated in Fig. 1.2 [11], where a broad transmit beam pattern is achieved by an omni-directional transmit antenna, and multiple narrow beams are formed simultaneously by a Uniform Linear Array (ULA) of N antenna elements. The geometry of the incident signal and the ULA is illustrated in Fig. 1.3, where the antenna element-wise spacing is d. Assume that a stationary target is located in the far field and its distance to the nearest receive antenna is R, and the round trip delay is $t_0 = \frac{2R}{c}$, then, the noise-free receive signal at the i-th, $i \in [N]$ array element can be written as

$$r_i(t) \triangleq \alpha s(t - t_0 - \frac{id\sin(\theta)}{c})e^{j2\pi f(t-\frac{id\sin(\theta)}{c})}. \tag{1.19}$$

Under the assumption that the maximum time-delay between the two most distant antennas is much smaller than the reciprocal of the bandwidth of $s(t)$, i.e., [15]

$$\frac{(N-1)d}{c} << \frac{1}{B}, \tag{1.20}$$

the received signal from the i-th array element can be simplified to

$$r_i(t) \approx \alpha s(t - t_0)e^{j2\pi(ft - i\kappa)}, \tag{1.21}$$

where λ is the wavelength. The term κ is known as the spatial frequency [7], defined as

$$\kappa \triangleq \frac{d\sin(\theta)}{\lambda}. \tag{1.22}$$

For any time snapshot, the samples from all the antennas can be viewed as samples of a complex sinusoid with the (spatial) frequency of κ. Hence, such spatial frequency can be evaluated by DFT, subsequently, the target DOA is calculated as

$$\theta = \arcsin\left(\frac{\kappa\lambda}{d}\right). \tag{1.23}$$

Note that the largest unambiguous range of θ is $[-90°, 90°]$, which corresponds to the half-wavelength element-wise spacing, i.e., $d = \frac{\lambda}{2}$.

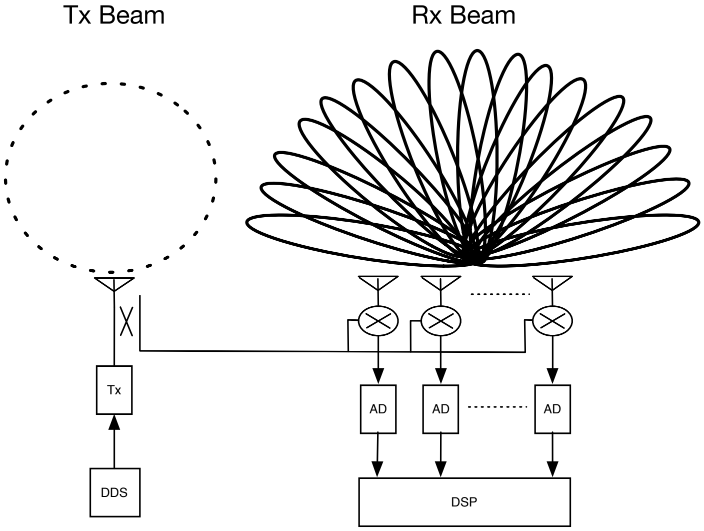


Figure 1.2: DBF radar system architecture. A broad beam pattern is formed with an omnidirectional transmit antenna, while multiple narrow beams are formed simultaneously by the receiving array. Each receiving channel is mixed with a coupled signal from the transmitter to demodulate (de-chirp) the FMCW signal, before AD conversion. The digitized received signal is processed by a Digital Signal Processor (DSP), while the transmit waveform is generated by a Direct Digital Synthesizer (DDS).

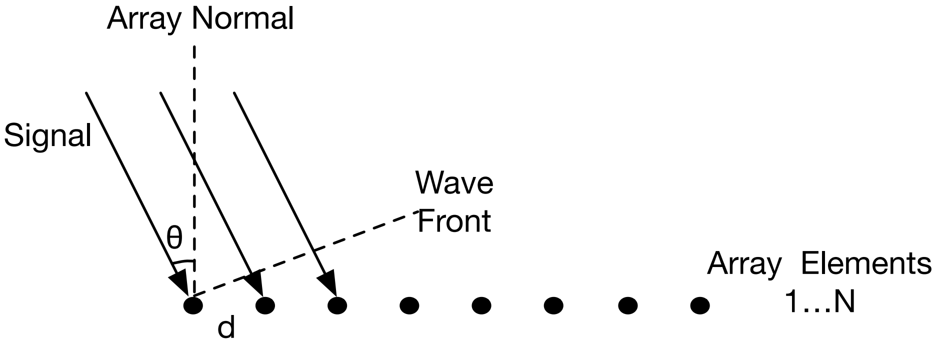


Figure 1.3: DOA estimation with a ULA.

Now, let us assume that the FMCW depicted in Fig. 1.1 is employed in the DBF radar, for the i-th, $i \in [N]$ antenna element and the m-th, $m \in [M]$ chirp, the noiseless demodulated signal of (1.18) is extended to

$$b(t, m, i) \triangleq \tilde{a} e^{j2\pi((f_r + f_d)(t - mT) + i\kappa)}. \tag{1.24}$$

This can be viewed as a 3-dimensional (3-D) complex sinusoid, whose frequencies correspond to the range (coupled with the Doppler), range rate, and DOA of the target. After AD conversion of each receiving channel, we can use the processing scheme shown in Fig. 1.4 to detect the targets as well as estimate their range, velocity and DOA. More specifically, grid-based versions of f_r, f_d, κ can be calculated by applying a 3-D FFT on the windowed data cube [11], followed by a detection procedure.

1.4 MIMO Radar

The DBF radar configuration in Fig. 1.2 with one transmit antenna and multiple receive antennas can be viewed as a Single-Input-Multi-Output (SIMO) radar. By increasing the degree of freedom of the transmit channels, such SIMO radar becomes a collocated MIMO radar [14, 16] as illustrated in Fig. 1.5. For collocated MIMO radars, the transmit and receive antennas are closely located and thus the target RCS experienced by different transmit and receive pairs could be viewed as identical; this allows collocated MIMO radars to exploit phase differences in target returns induced by transmit and receive antennas, i.e., employ

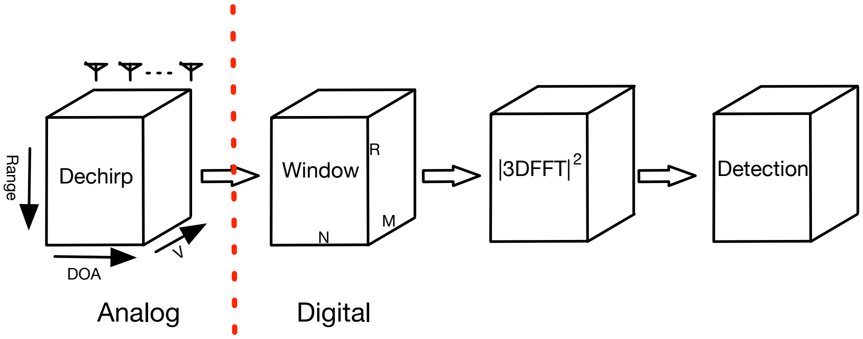


Figure 1.4: Conventional FFT-based processing scheme for the DBF FMCW radar.

coherent processing to effectively increase the array aperture and achieve high resolution as well as high SNR.

Another kind of MIMO radars are distributed MIMO radars [17, 18, 19], where the transmit and receive antennas are widely separated from each other compared with their distance to the targets. In such scenario, the transmit antennas emit independent waveforms, which propagate through independent paths from transmitters to receivers via the targets. As a result, distributed MIMO radars enjoy spatial diversity to reduce the RCS scintillation of targets.

In the following, we illustrate a basic collocated MIMO radar. Let us consider the MIMO radar configuration of Fig. 1.5, where the transmission and reception are accomplished by two ULAs of M_t and M_r antenna elements, respectively. The element-wise spacing of the two arrays are assumed to be d_t, d_r, respectively. During transmission, a set of orthogonal Code Division Multiple Access (CDMA) waveforms are transmitted by each transmit antenna. The L-coded discrete baseband signal transmitted by the $u_{\text{th}}, u \in [M_t]$ antenna is expressed as

$$s_u(t) \triangleq \frac{1}{T_b} \sum_{l=0}^{L-1} s_{ul} \text{Rect}[t - (l-1)T_b], \; 0 \leq t \leq T, \tag{1.25}$$

where Rect() is the rectangular window of width T_b, and T_b, T represent the sub-pulse and pulse duration, respectively; s_{ul} is the l-th, $l \in [L]$ code, and the code vector transmitted

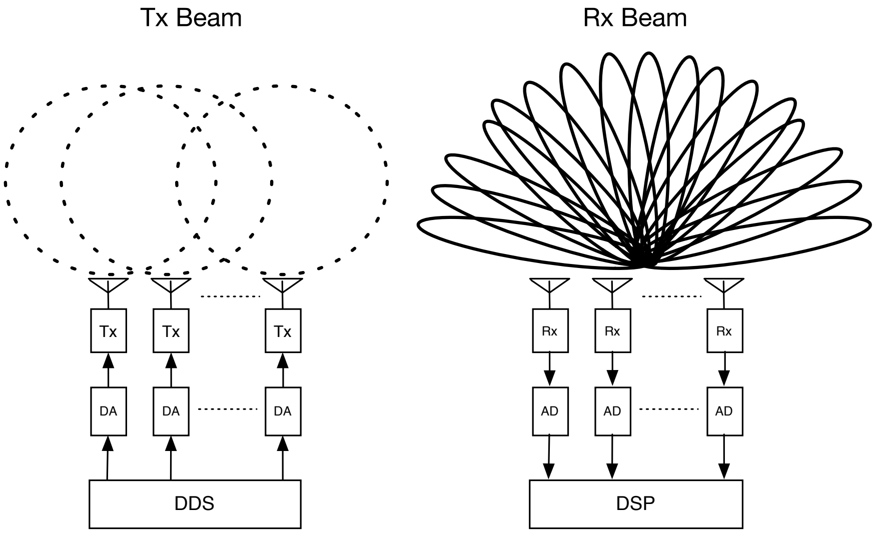


Figure 1.5: Collocated MIMO Radar System with ULA. Each transmit antenna transmits an orthogonal waveform, which is generated by a DDS. The orthogonality of the waveforms results into an omni-directional transmit beam pattern, while multiple narrow beams are formed simultaneously by beamforming in the DSP.

from the u-th antenna is denoted as

$$\mathbf{s}_u \triangleq [s_{u0}, \cdots, s_{u(L-1)}]^T. \tag{1.26}$$

Let us assume that the code is unimodal [20], i.e., $|s_{ul}| = 1, u \in [M_t], i \in [L]$. Ideally, code vectors transmitted from different antennas are mutually orthogonal, i.e., $\frac{1}{L}\mathbf{s}_u^T\mathbf{s}_v = 1$ if $u = v$, otherwise 0, hence

$$\frac{1}{L}\int_{t=0}^{T} s_u(t)s_v(t)dt = \delta_{uv}. \tag{1.27}$$

From the i-th, $i \in [M_r]$ receive antenna, the noiseless received baseband signal from a stationary point target is a t_0-delayed version of the superposition of the M_t transmitted signals, modeled as

$$r_i(t) \triangleq \alpha \sum_{u=0}^{M_t-1} s_u(t-t_0)e^{j2\pi(u\kappa_t + i\kappa_r)}, \tag{1.28}$$

where $\alpha \in \mathbb{C}$ absorbs the constants, and

$$\begin{aligned}\kappa_t &\triangleq \frac{d_t\sin(\theta)}{\lambda},\\ \kappa_r &\triangleq \frac{d_r\sin(\theta)}{\lambda},\end{aligned} \tag{1.29}$$

are the spatial frequencies introduced during transmission and reception, respectively, with θ, λ representing the target DOA and wavelength, respectively.

Owning to the orthogonality of waveforms, each transmit waveform of (1.28) can be separated through matched filtering with a filter bank of M_t filters, i.e., correlating the received signal with $s_u(t), u \in [M_t]$. For the snapshot corresponding to t_0, the matched output of the receive array is expressed as

$$\mathbf{r} = \tilde{a}\mathbf{v}_r(\kappa_r) \bigotimes \mathbf{v}_t(\kappa_t), \tag{1.30}$$

where $\tilde{a}$ is a constant, and

$$\begin{aligned} \mathbf{v}_r(\kappa_r) &\triangleq [0, e^{-j2\pi\kappa_r}, \cdots, e^{-j2\pi\kappa_r(M_r-1)}]^T, \\ \mathbf{v}_t(\kappa_t) &\triangleq [0, e^{-j2\pi\kappa_t}, \cdots, e^{-j2\pi\kappa_t(M_t-1)}]^T, \end{aligned} \tag{1.31}$$

are the receive and transmit steering vectors, respectively.

When $d_r = M_t d_t$,

$$\mathbf{r} = \tilde{a}[0, e^{-j2\pi\kappa_t}, \cdots, e^{-j2\pi\kappa_t(M_tM_r-1)}]^T, \tag{1.32}$$

which can be viewed as the response from a virtual array of M_tM_r antennas with element-wise spacing of d_t; this achieves savings of $M_tM_r - M_r - M_r$ antennas and subsequent receive chains as compared to a conventional ULA of the same resolution. Fig. 1.6 illustrates a virtual array of 6 elements is formed by 3 transimit antennas and 2 receive antennas.

The output of the matched filter bank can be also arranged in a matrix format as

$$\mathbf{R} = \tilde{a} \begin{bmatrix} 0, & e^{-j2\pi\kappa_t}, & \cdots, & e^{-j2\pi\kappa_t(M_t-1)} \\ e^{-j2\pi\kappa_r}, & e^{-j2\pi(\kappa_r+\kappa_t)}, & \cdots, & e^{-j2\pi(\kappa_r+\kappa_t(M_t-1))} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-j2\pi\kappa_r(M_r-1)}, & e^{-j2\pi(\kappa_r(M_r-1)+\kappa_t)}, & \cdots, & e^{-j2\pi(\kappa_r(M_r-1)+\kappa_t(M_t-1))} \end{bmatrix}, \tag{1.33}$$

which is a 2-D complex sinusoid, whose discrete (spatial) frequency, (κ_r, κ_t), can be evaluated by a 2-D DFT. This is the 2-D transmit and receive beamforming. Unlike phased array radars that form transimit beams at transmission, collocated MIMO radars are able to form transmit beams at reception [14]. Compared to DBF radars with the same receive aperture that only apply receive beamforming, the two-way beamforming of MIMO radars increases SNR and achieves a narrower two-way beam pattern [14].

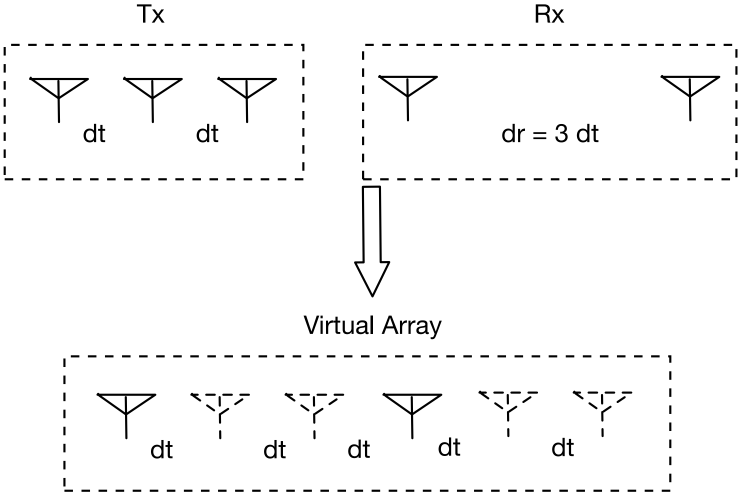


Figure 1.6: Virtual array. $M_t = 3, M_r = 2, d_r = M_t d_t$.

1.5 Sparse Fourier Transform

As discussed in the preceding sections, conventional radar signal processing methods involve multidimensional DFT, which are typically implemented by the FFT. The sample complexity of the FFT is $O(N)$, where N is the number of samples in the multidimensional sample space. For N equal to a power of 2, the computational complexity of the FFT is $O(N \log N)$.

Recently, by leveraging the sparsity of signals in the frequency domain, a series of SFT algorithms have been proposed [4, 5, 21, 22, 23, 24, 3, 25, 26, 27, 28, 29, 30, 31, 32]; this is a family of low-complexity DFT algorithms. The state-of-the-art SFT algorithms [3, 32] achieve sample complexity of $O(K)$ and computational complexity of $O(K \log K)$ for exactly K-sparse (in the frequency domain) signals. When $K << N$, those SFT algorithms achieve significant savings both in sample and computation compared to the FFT. SFT algorithms have been investigated for several applications including a fast global positioning system receiver, wide-band spectrum sensing, and bio-medical signal processing [33, 34, 2, 1].

In SFT algorithms, the reduction of sample and computational complexity is achieved by reducing data samples. This is usually implemented via a well designed subsampling procedure, which leverages the resulting frequency domain aliasing. We demonstrate such technique in the following trivial example, where the signal is 1-sparse in the frequency domain, i.e., only contains a single frequency, $f \in [N]$. Such signal is expressed as

$$x(n) \triangleq ae^{j\frac{2\pi}{N}fn},\ n = 0, 1, \cdots, N-1. \tag{1.34}$$

The N-point DFT of (1.34) yields

$$\hat{x}_N(k) \triangleq \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}nk},\ k = 0, 1, \cdots, N-1. \tag{1.35}$$

It is clear that $\hat{x}(k)/N$ equals to a when $k = f$, and 0, otherwise. An example of $x(n), \hat{x}(k)$ with $N = 15, f = 11, a = 1$ is shown in Figs. 1.7 (a) and (b), respectively.

Let us assume that $B \in \mathbb{N}$ divides N, and $L = N/B$. The L-subsampled version of (1.34) is expressed as

$$x_B(n) \triangleq x(nL) = ae^{j\frac{2\pi}{B}fn},\ n = 0, 1, \cdots, B-1. \tag{1.36}$$

The B-point DFT of such subsampled signal becomes

$$\hat{x}_B(k) \triangleq \sum_{n=0}^{B-1} x_B(n)e^{-j\frac{2\pi}{B}nk} = \sum_{n=0}^{B-1} ae^{-j\frac{2\pi}{B}n(k-f)},\ k = 0, 1, \cdots, B-1. \tag{1.37}$$

We can see that $\hat{x}_B(k)/B = a$ when $k \equiv f \bmod B$, and 0, otherwise. In another words, the N frequency buckets of $\hat{x}_N(k)$ is aliased to B buckets of $\hat{x}_B(k)$, such that

$$\hat{x}_B(k) = \frac{B}{N}\sum_{l=0}^{L-1} \hat{x}_N(lB + k),\ k = 0, 1, \cdots, B-1. \tag{1.38}$$

Now, let us assume special cases when N can be factorized into a series of relative prime (co-prime) numbers, e.g., $N = 15 = 3 \times 5$. We then compute $\hat{x}_3(k), k = 0, 1, 2$ and $\hat{x}_5(k), k = 0, 1, \cdots, 4$. Next, for some $k_1 \in [3], k_2 \in [5]$, we obtain

$$\hat{x}_3(k_1)/3 = \hat{x}_5(k_2)/5 = a. \tag{1.39}$$

Since

$$\begin{aligned} k_1 &\equiv f \mod 3, \\ k_2 &\equiv f \mod 5, \end{aligned} \tag{1.40}$$

one can uniquely determine f due to the *Chinese Reminder Theorem*, which is stated as follows [22, 35].

Theorem 1. ***(Chinese Reminder Theorem):*** *Any integer x is uniquely specified modulo N by its remainders modulo m relatively prime integers $p_0, p_1, \cdots, p_{m-1}$ as long as $\prod_{i\in[m]} p_i \geq N$.*

To complete the example, let us assume that $k_1 = 2$, $k_2 = 1$, then, there exists unique $u, v \in [N]$, such that

$$3u + 2 = 5v + 1 = f. \tag{1.41}$$

This allows us to solve $f = 11$. In this case the frequency and its DFT coefficient is calculated using $3 + 5 = 8$ samples and via two short DFTs. Thus, compared to the conventional 15-point DFT, savings in samples and computation are achieved. Such Chinese Remainder Theorem based time domain subsampling and the resulting frequecy domain aliasing is illustrated in Fig. 1.7.

The above 1-sparse case is the simplest scenario for SFT. In a general K-sparse case, multiple original frequencies may land into a same frequency bucket of short DFTs due to aliasing. This is refereed to as *frequency collision*; this prevents us to associate the same frequencies from different copies of short DFTs, which results into failures of frequency recovery. One commonly used technique in SFT algorithms to detect frequency collision and solve the frequency from a non-collision bucket is called *phase encoding* or *OFDM-trick* [22, 3]. To illustrate this, let us consider a 2-sparse case, i.e., the signal contains two frequencies $f_0, f_1 \in [N]$, and

$$x(n) \triangleq ae^{j\frac{2\pi}{N}f_0 n} + be^{j\frac{2\pi}{N}f_1 n},\ n = 0, 1, \cdots, N-1. \tag{1.42}$$

Let us extract two L-subsampled versions of (1.42) with the same subsampling rate of L but different offsets, i.e., 0 and 1. The two sabsampled version for $n = 0, 1, \cdots, B-1$, is expressed as

$$\begin{aligned} x_{B,0}(n) &\triangleq x(nL), \\ x_{B,1}(n) &\triangleq x(nL+1). \end{aligned} \tag{1.43}$$

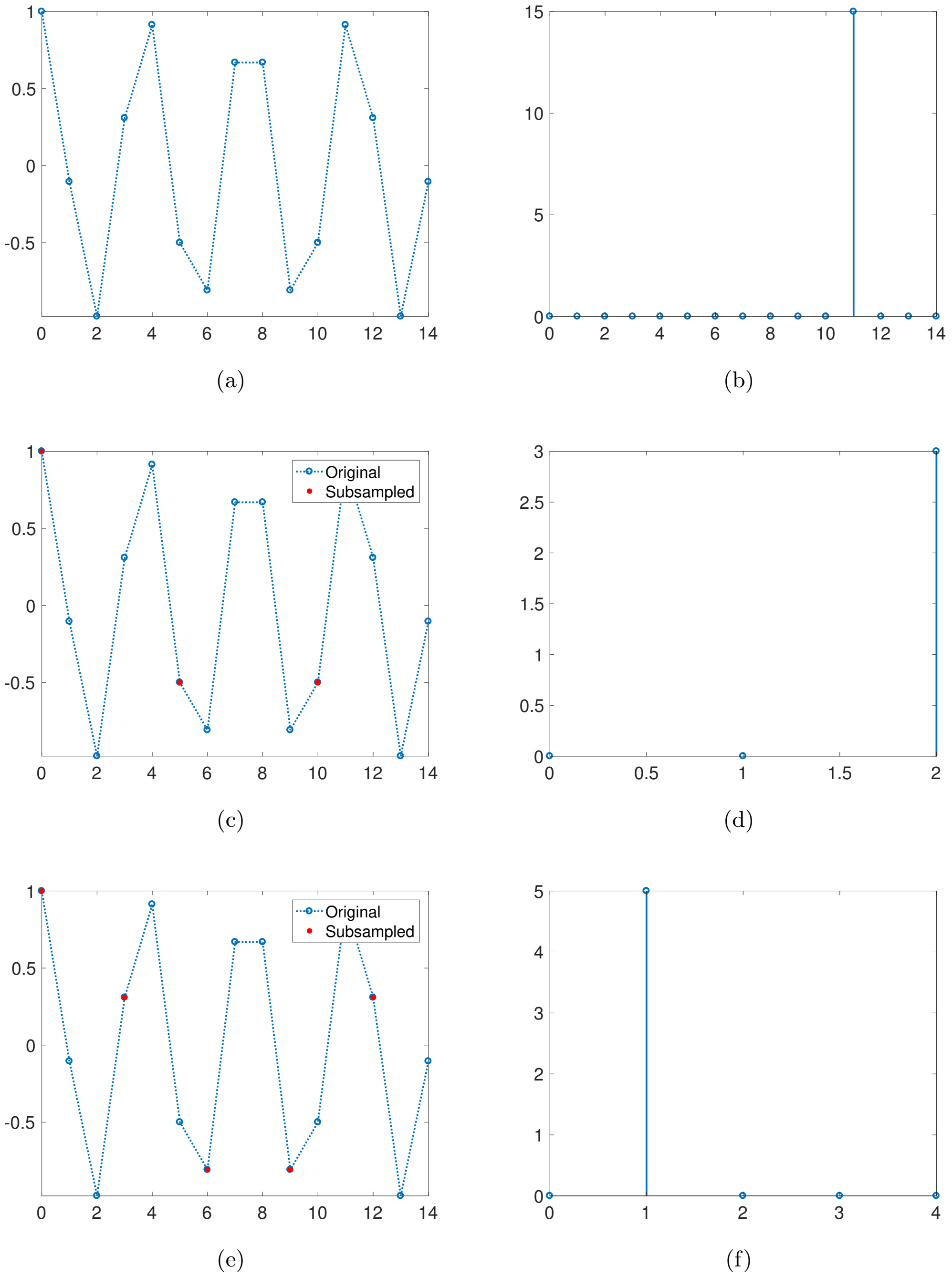


Figure 1.7: Time domain subsampling and frequency domain aliasing. (a) The real part of a time domain single tone, i.e., (1.34) with $N = 15, f = 11, a = 1$. (b) 15-point DFT of (1.34). (c) 5-subsampled version of (1.34), i.e., $x_3(n), n = 0, 1, 2$. (d) $\hat{x}_3(k), k = 0, 1, 2$. (e) 3-subsampled version of (1.34), i.e., $x_5(n), n = 0, 1, \cdots, 4$. (f) $\hat{x}_5(k), k = 0, 1, \cdots, 4$.

Then, for $k = 0, 1, \cdots, B-1$, their B-point DFTs are computed as

$$\begin{aligned}
\hat{x}_{B,0}(k) &\triangleq \sum_{n=0}^{B-1} x_{B,0}(n) e^{-j\frac{2\pi}{B}nk} = a\sum_{n=0}^{B-1} e^{-j\frac{2\pi}{B}n(k-f_0)} + b\sum_{n=0}^{B-1} e^{-j\frac{2\pi}{B}n(k-f_1)}, \\
\hat{x}_{B,1}(k) &\triangleq \sum_{n=0}^{B-1} x_{B,1}(n) e^{-j\frac{2\pi}{B}nk} = ae^{j\frac{2\pi}{B}f_0}\sum_{n=0}^{B-1} ae^{-j\frac{2\pi}{B}n(k-f_0)} + be^{j\frac{2\pi}{B}f_1}\sum_{n=0}^{B-1} e^{-j\frac{2\pi}{B}n(k-f_1)},
\end{aligned} \tag{1.44}$$

respectively. Note that additional phase shift terms are introduced to each frequency in $\hat{x}_{B,1}(k)$ due to the shift (by 1) in time-domain subsampling.

Next, let us investigate *each* bucket of $\hat{x}_{B,0}(k)$ and $\hat{x}_{B,1}(k)$. There are three cases:

- Case 1: There is no frequency landed in the bucket, as a result, $|\hat{x}_{B,0}(k)| = |\hat{x}_{B,1}(k)| = 0$.

- Case 2: Either f_0 or f_1 lands in the bucket, i.e., either $k \equiv f_0 \mod B$ or $k \equiv f_1 \mod B$. Without loss of generality, let us assume that the former is true, then, $|\hat{x}_{B,0}(k)| = |\hat{x}_{B,1}(k)| = a$. Moreover

$$f_0 = \frac{B}{2\pi}\phi\left(\hat{x}_{B,1}(k)/\hat{x}_{B,0}(k)\right), \tag{1.45}$$

  where $\phi(x)$ is the phase of x.

- Case 3: Both f_0 and f_1 land in the same bucket, i.e., frequency collision. This can be detected by $|\hat{x}_{B,0}(k)| \neq |\hat{x}_{B,1}(k)|$.

Therefore, the OFDM-trick provides a low-complexity (O(1)) operation to detect frequency collision from short DFTs, while solving original frequencies from 1-sparse bins. The collided frequencies in one short DFT maybe separated from another short DFT applied on another copy of subsampled signal using a different sabsampling rate, which introduces a different aliasing pattern in the frequency domain.

Generally, SFT admits a *subsampling-localization-estimation* procedure. First, the original signal is subsampled, then, the significant (dominant) frequencies contained in the original signal are localized and the corresponding DFT coefficients are estimated with low-complexity operations. Such subsampling-localization-estimation procedure is carried out in an *iterative* manner in several SFT algorithms [21, 22, 3, 32], while for other SFT algorithms [4, 2, 1], the localization and estimation are implemented in *one-shot* after gathering sufficient copies of subsampled signals corresponding to different subsampling parameters, e.g., subsample rate, offset and number of samples; this approach makes the one-shot based SFT algorithms less sensitive to noise as compared to the iterative method, since the accumulation of many copies of the subsampled signal effectively reduces the noise [4]. However, iterative-based SFT algorithms usually exhibit lower complexity as compared to one-shot based SFT algorithms, since in the former, in each iteration, the contribution of the recovered frequencies are removed from the signal, which yields a sparser signal (an easier problem) in the next iteration. High level illustrations of one-short and iterative-based SFT are shown in Fig. 1.8.

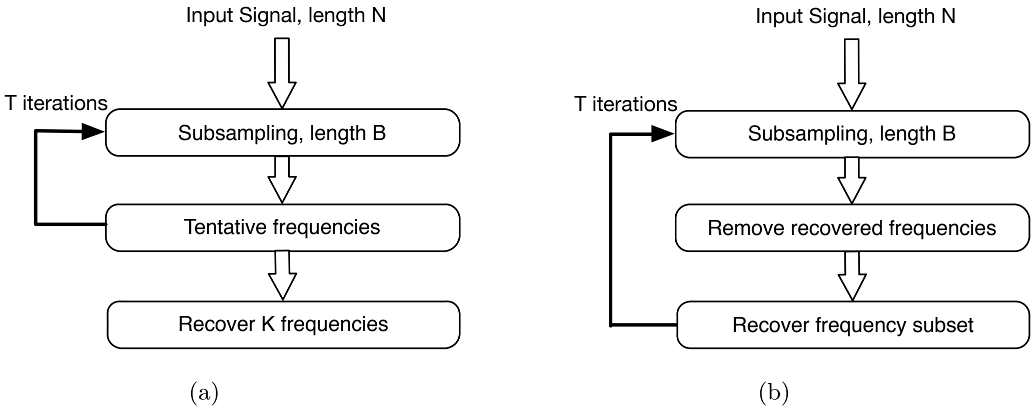


Figure 1.8: SFT high level illustration. (a) One-shot based SFT. (b) Iterative-based SFT.

Most of existing SFT algorithms are designed for 1-D signals; unlike the FFT, the extension of 1-D SFT to multidimensional SFT is usually not straightforward; this is because the SFT algorithms are not separable in each dimension due to the fact that operations such as detection of significant frequencies in the subsampled signal within an SFT algorithm must be considered jointly for all the dimensions [36]. In the following, we review simple and practical SFT [4] and SO-SFT [3], which belong to 1-D and multidimensional SFT, respectively. Simple and practical SFT is based on the one-shot approach, while the SO-SFT is an iterative method. Those two SFT algorithms form the foundation of our proposed RSFT [36] and MARS-SFT [37] algorithms, respectively.

1.5.1 Simple and Practical SFT

As opposed to the SFT introduced in the above section that leverages the time-domain subsampling to reduce sample and computational complexity, simple and practical SFT leverages the frequency domain subsampling achieved by the time-domain aliasing of data. Before outlining such algorithm, we provide some key definitions and properties for signal $\mathbf{x} \in \mathbb{C}^N$.

Definition 1. ***(Permutation):*** *Let $\sigma \in [N]$, with σ being invertible, i.e., there exists $\sigma^{-1} \in [N]$, such that*

$$[\sigma\sigma^{-1}]_N = 1. \tag{1.46}$$

Then, a permutation *of $\mathbf{x}$ is defined as*

$$[\mathbf{P}_{\sigma,b}\mathbf{x}]_i = [\mathbf{x}]_{[\sigma i+b]_N}, \tag{1.47}$$

where $b \in [N]$; $\mathbf{P}_{\sigma,b} \in \{0,1\}^{N\times N}$ is the permutation matrix, which reorders entries of $\mathbf{x}$ modularly.

The permutation has the following property.

Property 1. *A permutation of the data in time domain results in a modular dilation in the frequency domain introduced by σ, and a phase term introduced by b. Let $\hat{\mathbf{x}}$ be the N-point DFT of $\mathbf{x}$, then*

$$[\widehat{\mathbf{P}_{\sigma,b}\mathbf{x}}]_{[\sigma i]_N} = [\hat{\mathbf{x}}]_{\mathbf{i}}\mathbf{e}^{-\mathbf{jb}\boldsymbol{\Delta}\omega_{\mathbf{N}}}, \tag{1.48}$$

where $\Delta\omega_N \triangleq 2\pi/N$.

Note that since the phase rotation introduced by b does not contributes to the localization of frequencies in our algorithm, in the following, we will assume that $b = 0$, and simplify the notation of the permutation matrix as $\mathbf{P}_\sigma$.

Definition 2. ***(Time-domain aliasing):*** *Let $\mathbf{y} \in \mathbb{C}^B$, with N multiple of B. For $L = N/B$, a time-domain aliased version of $\mathbf{x}$ is defined as*

$$[\mathbf{y}]_{\mathbf{i}} = \sum_{\mathbf{j}=\mathbf{0}}^{\mathbf{L}-\mathbf{1}} [\mathbf{x}]_{\mathbf{i}+\mathbf{Bj}}, \quad \mathbf{i} = \mathbf{0}, \mathbf{1}, \cdots, \mathbf{B}-\mathbf{1}. \tag{1.49}$$

Property 2. *Aliasing in the time domain results in down-sampling in the frequency domain. If $\hat{\mathbf{y}}$ is the B-point DFT of $\mathbf{y}$, then*

$$[\hat{\mathbf{y}}]_{\mathbf{i}} = [\hat{\mathbf{x}}]_{\mathbf{iL}}. \tag{1.50}$$

Definition 3. ***(Mapping):*** *Let $i, \sigma \in [N]$, where σ satisfies (1.46). We define the mapping $\mathcal{M}(i, \sigma) \in [B]$ such that*

$$\mathcal{M}(i, \sigma) \equiv [\lfloor \frac{B}{N}[i\sigma]_N \rceil]_B. \tag{1.51}$$

Definition 4. ***(Reverse-mapping):*** *Let $\sigma^{-1} \in [N]$, σ^{-1} satisfies (1.46), and $j \in [B]$. Define $\mathcal{R}(j, \sigma^{-1})$ a reverse-mapping such that*

$$\mathcal{R}(j, \sigma^{-1}) \equiv \{[\sigma^{-1}u]_N \mid u \in \mathbb{S}\}, \tag{1.52}$$

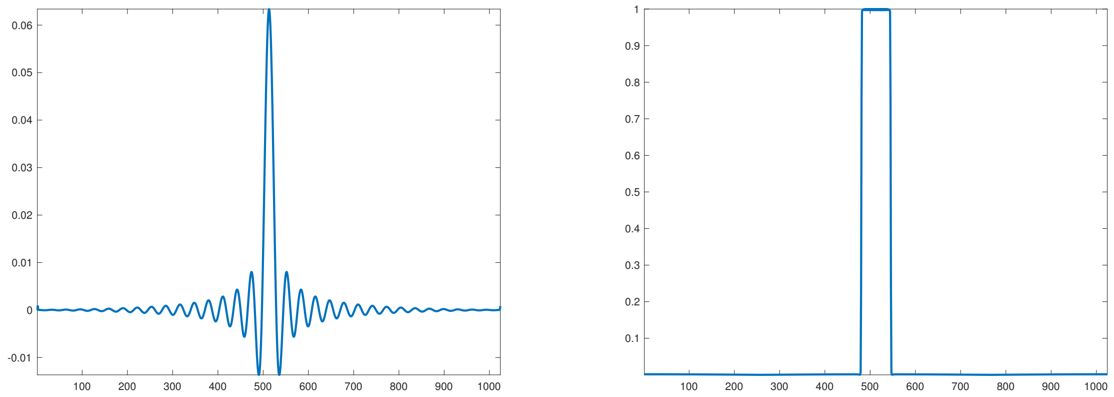
where

$$\mathbb{S} \triangleq \{v \in [N] \mid j\frac{N}{B} \leq v < (j+1)\frac{N}{B}]\}. \tag{1.53}$$

At a high level, simple and practical SFT runs two loops, namely the *Localization* loop and the *Estimation* loop. The former finds the indices of the K most significant frequencies from the input signal, while the latter estimates the corresponding Fourier coefficients. Here, we emphasize on Localization more than Estimation, since the former is more relevant to the radar application that we consider; the Localization step provides frequency locations, which in the radar case are directly related to target parameters.

In the Localization loop, as illustrated in Fig. 1.10, a *permutation* procedure reorders the input data $\mathbf{x}$ in the time domain, causing the frequencies to also reorder. The permutation causes closely spaced frequencies in $\hat{\mathbf{x}}$ to appear in well separated locations with high probability. Then, a *frequency-domain flat-window* (flat-window hereafter) [4], as illustrated in Fig. 1.9, is applied on the permuted signal. The flat-window is a sinc-like function in the time domain; the multiplication of the time-domain signal to such window is equivalent to convolving the signal spectrum to a boxcar in the frequency domain; this extends a single frequency into a (nearly) boxcar of width $L = N/B$ in the frequency domain, for a reason that will become apparent in the following. The flat-windowed data are aliased, as in Definition 2. The frequency domain equivalent of this aliasing is undersampling by L

(see Property 2). The flat-window used at the previous step ensures that no peaks from the significant frequencies are lost due to the effective down-sampling in the frequency domain. After this stage, an FFT of length B is employed.



(a)

(b)

Figure 1.9: A frequency domain flat window. $N = 1024, B = 16$. (a) Time domain. (b) Frequency domain. Shifted by $N/2$.

The permutation and the aliasing procedure effectively map the discrete signal frequencies from the set $\mathbb{S}_N \triangleq \{0, \Delta\omega_N, \cdots, (N-1)\Delta\omega_N\}$ to $\mathbb{S}_B \triangleq \{0, \frac{2\pi}{B}, \cdots, \frac{(B-1)2\pi}{B}\}$, where the first stage detection procedure finds the significant frequencies' peaks, and subsequently their locations are reverse-mapped back into $\mathbb{S}_N$ to restore the frequency resolution. However, the reverse-mapping yields not only the true locations of the significant frequencies, but also L ambiguous locations for each frequency. To remove the ambiguity, multiple iterations of Location with randomized permutation are performed. Finally, the second stage detection procedure locates the K most significant frequencies from the accumulated data after running T iterations.

Simple and practical SFT is designed for 1-D signal, which is not applicable for multidimensional signal processing. Moreover, simple and practical SFT assumes that we know the exact sparsity level, K, and those K frequenceis are all on the grid; those assumptions are not practical in real-world radar signal processing. We address those problems in Chapter 2, where we propose the RSFT.

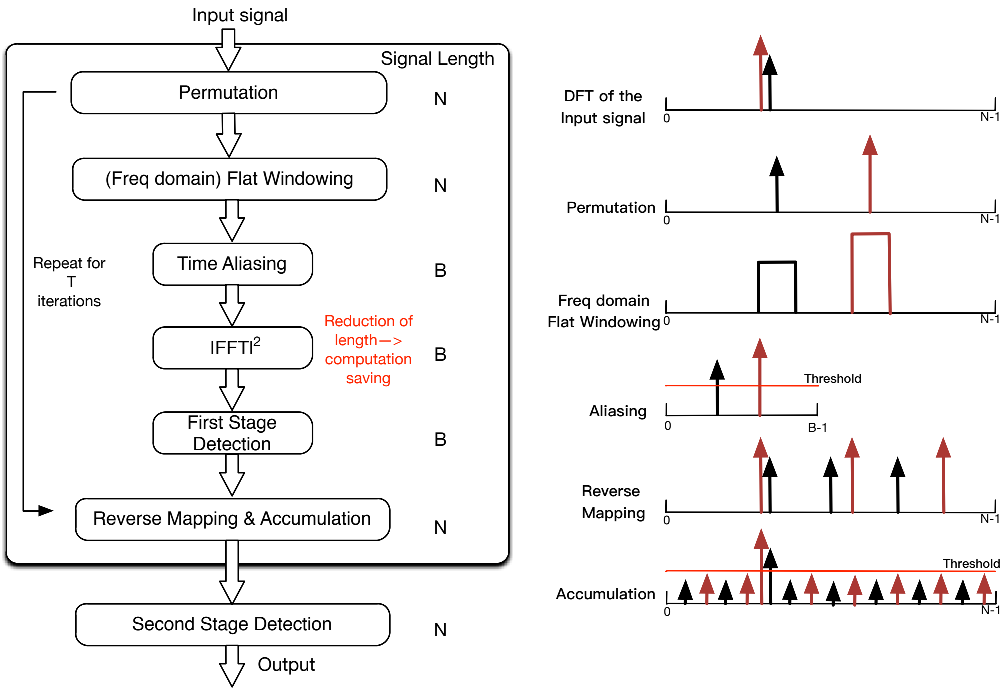


Figure 1.10: Illustration of simple and practical SFT. The left part of the figure shows the main steps of the localization loop of the SFT algorithm, while the right part shows the signal representations in the discrete frequency domain due to each step.

1.5.2 SO-SFT

Multidimensional SFT algorithms are investigated in [3, 1, 2]. The SO-SFT [3] follows the subsampling-localization-estimation iteration, while the SFT algorithms of [1, 2] are one-shot approaches. The idea behind those SFT algorithms is to reduce a multidimensional DFT into a series of 1-D DFTs along slices extracted from the input multidimensional data. SO-SFT achieves the sample and computational complexity lower bounds of all known SFT algorithms, i.e., $O(K)$ and $O(K \log K)$, respectively, by reducing a 2-D DFT into 1-D DFTs along a few columns and rows of a data matrix.

Let us consider the following 2-D signal model, which is a superposition of K 2-D complex sinusoids, i.e.,

$$x(\mathbf{n}) \triangleq \sum_{(a,\boldsymbol{\omega})\in\mathbb{S}} ae^{j\mathbf{n}^T\boldsymbol{\omega}},\ \mathbf{n} \triangleq [n_0, n_1]^T \in \mathcal{X} \triangleq [N_0] \times [N_1], \tag{1.54}$$

where N_0, N_1 denote the sample length of the two dimensions, respectively. $(a, \boldsymbol{\omega})$ represents a 2-D frequency whose amplitude is a with $a \in \mathbb{C}, a \neq 0$ and frequency is $\boldsymbol{\omega} \triangleq [\omega_0, \omega_1]^T \triangleq [2\pi m_0/N_0, 2\pi m_1/N_1]^T$ with $[m_0, m_1]^T \in \mathcal{X}$. The set $\mathbb{S}$ with $|\mathbb{S}| = K$ includes all the 2-D frequencies. We assume that the signal is sparse in the frequency domain, i.e., $K << N \triangleq N_0N_1$. We are interested in the recovery of all the frequencies from samples of $x(\mathbf{n})$.

In SO-SFT, in order to recover $\mathbb{S}$, 1-D DFTs are applied on a subset of columns and rows of the data matrix. The N_0-point DFT of the i-th, $i \in [N_1]$ column of the data equals

$$\begin{aligned}
\hat{c}_i(m) &\triangleq \frac{1}{N_0}\sum_{l=0}^{N_0-1} x(l,i)e^{-j\frac{2\pi}{N_0}ml} \\
&= \frac{1}{N_0}\sum_{(a,\boldsymbol{\omega})\in\mathbb{S}}\sum_{l=0}^{N_0-1} ae^{j\frac{2\pi}{N_1}m_1 i}e^{j\frac{2\pi}{N_0}l(m_0-m)} \\
&= \sum_{(a,\boldsymbol{\omega})\in\mathbb{S}} ae^{-j\frac{2\pi}{N_1}m_1 i}\delta(m-m_0),\ m = 0, 1, \cdots, N_0 - 1, \\
\boldsymbol{\omega} &= [2\pi m_0/N_0, 2\pi m_1/N_1]^T, [m_0, m_1]^T \in \mathcal{X},
\end{aligned} \tag{1.55}$$

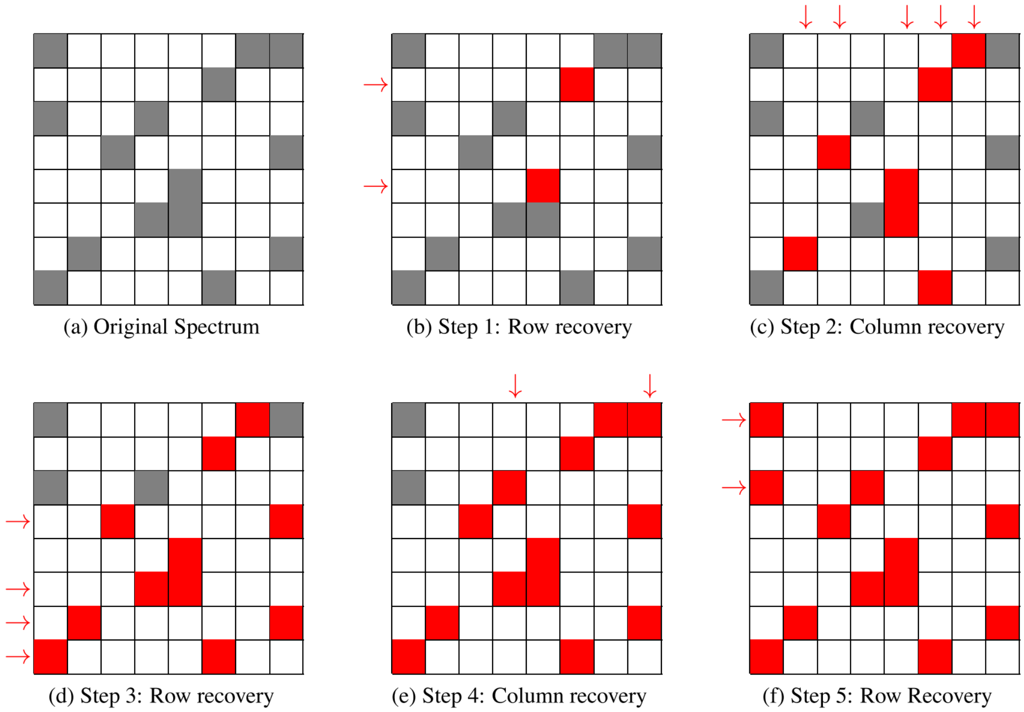
where $\delta(\cdot)$ is the unit impulse function. Hence $\hat{c}_i(m_0)$ can be viewed as the summation of modulated amplitudes of 2-D sinusoids whose row frequency indices equal to m_0. Hence $\hat{c}_i(m), m = 0, 1, \cdots N_1 - 1$ is a projection of the 2-D spectrum, $\hat{x}(m_0, m_1), [m_0, m_1]^T \in \mathcal{X}$, onto the column. Similarly, the N_1-point DFT applied on a row of (1.54) is a projection of the 2-D spectrum on the row.

Since the signal is sparse in the frequency domain, if $|\hat{c}_i(m)| \neq 0$, with high probability, there will be only one significant frequency projected to the frequency bin of m; in other words, the frequency bin is '1-sparse', and $\hat{c}_i(m)$ is reduced to $\hat{c}_i(m) = \hat{c}_i(m_0) = ae^{j\frac{2\pi}{N_1}m_1 i}$. In such case, $a = \hat{c}_0(m_0)$, and the other frequency component, m_1, can be solved by the OFDM-trick, i.e., $m_1 = \phi\left(\hat{c}_1(m_0)/\hat{c}_0(m_0)\right)\frac{N_1}{2\pi}$.

The contribution of the recovered sinusoids is removed from the signal, so that the following processing can be applied on a sparser signal, which is easier to solve in the subsequent processing.

A frequency bin that is not 1-sparse based on column processing might be 1-sparse based on row processing. Because the removal of sinusoids in the column (row) processing may cause bins in the row (column) processing to be 1-sparse, SO-SFT runs iteratively and alternatively between columns and rows and the algorithm stops after a finite number of iterations. Such iterative recovery of frequencies is illustrated in Fig. 1.11.

SO-SFT succeeds with high probability only when the frequencies are very sparse, and requires that either a row or a column of the DFT contains a 1-sparse bin. However, in many applications, the signal frequency exhibits a block sparsity pattern [38], i.e., the significant frequencies are clustered. In those cases, even when the signal is very sparse, 1-sparse bin may not exist; this is referred to as a 'deadlock' case [3]. As shown in Fig. 1.12, neither column nor row DFT yields a 1-sparse bin in the two cases, hence the frequencies, although are sufficiently sparse, cannot be recovered. Such problem is addressed in Chapter 4, where we propose the MARS-SFT. By projecting multidimensional frequencies to lines of random slopes and offsets extracted from the data cube, MARS-SFT could resolve the deadlocks encountered by the SO-SFT. SO-SFT and MARS-SFT deals with ideal signals, i.e., noiseless signals containing only on-grid frequencies. In Chapter 5, we propose the robust MARS-SFT, which addresses noisy signals with off-grid frequencies arising from real-world radar signal processing applications.



(a) Original Spectrum

(b) Step 1: Row recovery

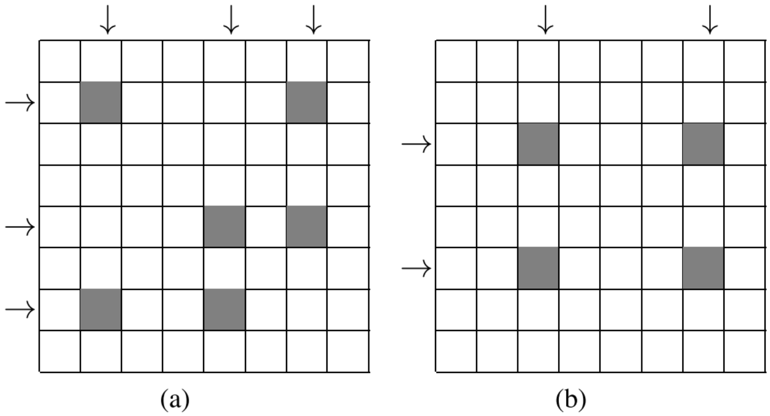
(c) Step 2: Column recovery

(d) Step 3: Row recovery

(e) Step 4: Column recovery

(f) Step 5: Row Recovery

Figure 1.11: Illustration of SO-SFT [3]. Frequencies are recovered iteratively by alternatively applying DFT along rows and columns of the data matrix.



(a)

(b)

Figure 1.12: Deadlock situations for SO-SFT

1.6 Contribution of the Dissertation

1.6.1 SFT Algorithms that Address Real-World Multidimensional Signals

We propose multidimensional SFT algorithms that address real-world signals, which lay a foundation for SFT-based radar signal processing. The proposed algorithms are summarized in the following:

- In Chapter 2, we propose the RSFT to address multidimensional signals that contain noise and off-grid frequencies. The Neyman-Pearon (NP) detection is introduced in the RSFT to achieve the optimal detection while requires less computation as compared to the FFT-based approach. The tradeoff study in the context of the RSFT provides an extra degree of freedom in system design to trade off the computation with the detection performance.

- In Chapter 4, we propose MARS-SFT. MARS-SFT iteratively recovers multidimensional sparse frequencies by randomly projecting those frequencies into a 1-D frequency domain; such projection and recovery procedure has low sample and computation complexity. The abundance of degrees of freedom in random projection enables the MARS-SFT to efficiently deal with less sparse scenario, and even cases where frequencies are clustered. MARS-SFT achieves the sample and computation complexity lower bounds of SFT algorithms, i.e., $O(K)$ and $O(K \log K)$, respectively for K-sparse ideal signals (noiseless signals that only contain on-grid frequencies).

- Based on MARS-SFT, in Chapter 5, we propose the robust MARS-SFT, which handles real-world noisy signals that contain off-grid frequencies, while enjoys low-complexity property of MARS-SFT.

Those works have been published in

- Wang, Shaogang, Vishal M. Patel, and Athina Petropulu. "The robust sparse Fourier transform (RSFT) and its application in radar signal processing." IEEE Transactions on Aerospace and Electronic Systems 53.6 (2017): 2735-2755.

- Wang, Shaogang, Vishal M. Patel, and Athina Petropulu. “Multidimensional Sparse Fourier Transform Based on the Fourier Projection-Slice Theorem.” IEEE Transactions on Signal Processing 67.1 (2019): 54-69.

- Wang, Shaogang, Vishal M. Patel, and Athina Petropulu. “A practical high-dimensional Sparse Fourier Transform.” 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017.

- Wang, Shaogang, Vishal M. Patel, and Athina Petropulu. “FPS-SFT: A multidimensional sparse Fourier transform based on the Fourier projection-slice theorem.” 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018.

1.6.2 SFT-based Radar Signal Processing Frameworks for DBF and MIMO Radars

Based on the proposed RSFT and robust MARS-SFT, in Chapters 3 and 5, we formulate SFT-based signal processing frameworks for modern digital beamforming (DBF) and MIMO radars. Especially, we consider automotive radar and near range applications for the following reasons:

- The low-complexity SFT-based radar processing is mostly beneficial to radar applications of restricted hardware; automotive radars fall into such category due to their small size and low cost.

- Compared to FFT-based processing, SFT-based processing involves tradeoff between complexity and SNR. Generally, the more savings the SFT-based processing achieves, the higher SNR the system requires. Automotive radars work at near range, which is much easier to gain high SNR as compared to long-range radars; this allows SFT-based processing to achieve high savings.

This work has been published in

- Wang, Shaogang, Vishal M. Patel, and Athina Petropulu. “RSFT: A realistic high dimensional sparse Fourier transform and its application in radar signal processing.”

MILCOM 2016-2016 IEEE Military Communications Conference. IEEE, 2016.

- Wang, Shaogang, Vishal M. Patel, and Athina Petropulu. “MIMO-RSFT radar: A reduced complexity MIMO radar based on the Sparse Fourier Transform.” 2017 IEEE Radar Conference (RadarConf). IEEE, 2017.

- Wang, Shaogang, Vishal M. Patel, and Athina Petropulu. “Robust sparse Fourier transform based on the Fourier projection-slice theorem.” 2018 IEEE Radar Conference (RadarConf18). IEEE, 2018.

1.6.3 Two-dimensional DBF Radar with Orthogonal Linear Arrays Based on SFT and Deep Learning

In Chapter 6, we propose a new automotive radar architecture that achieves high resolution in range, range rate, azimuth and elevation DOA estimation, while requires a smaller antenna aperture and fewer reception channels as compared to conventional DBF planar arrays. The receive aperture of the proposed radar is composed of two orthogonally-placed uniform linear arrays (ULAs). In sparse scenarios, such radar architecture leverages the robust MARS-SFT to implement an efficient coarser beamforming for each ULA, while computes an high resolution range-Doppler image (RDI) for each beam, where each scatterer from targets can be resolved with high probability. Next, a deep-learning based method is employed to match azimuthal and elevational radar beams that associate with the same targets; such beam matching problem is converted to a image matching problem in the RDI domain.

Part of this work has been published in

- Wang, Shaogang, Chen, Honglei, Vishal M. Patel, and Athina Petropulu. “Two-Dimensional beamforming automotive radar with orthogonal linear arrays.” 2019 IEEE Radar Conference (RadarConf19). IEEE, 2019.

1.7 Outline of the Dissertation

The dissertation is organized as follows.

In Chapter 2, based on simple and practical SFT introduced in Section 1.5.1, we propose the RSFT algorithm that addresses real-world multidimensional signal that contains noise and off-grid frequencies.

In Chapter 3, based on the proposed RSFT algorithm in Chapter 2, we formulated the signal processing framework for DBF and MIMO radar.

In Chapter 4, we propose the MARS-SFT, which extends the SO-SFT introduced in Section 1.5.2, and achieves the lower bounds of sample and computational complexity of SFT in less sparse scenarios for ideal signals.

In Chapter 5, we propose the robust MARS-SFT, which extends the MARS-SFT to handle noisy signals containing off-grid frequencies; based on that, we investigated the application of the robust MARS-SFT in automotive DBF radar signal processing.

In Chapter 6, we propose 2-D DBF automotive radar with orthogonal linear arrays; the proposed radar architecture takes advantage of the robust MARS-SFT and deep learning, and achieves high resolution in range, range rate, azimuth and elevation angular domains with reduced hardware.

Finally, Chapter 7 contains conclusions and possible future research directions.

1.8 Notation

We use lower-case (upper-case) bold letters to denote vectors (matrices). $(\cdot)^T$ and $(\cdot)^H$ respectively denote the transpose and conjugate transpose of a matrix or a vector, and x^* is the conjugate of x. $\|\cdot\|$ is Euclidean norm for a vector. $\|\mathbf{W}\|_1, \|\mathbf{W}\|_2$ are the l_1 and l_2 norm of matrix $\mathbf{W}$, respectively. $[\mathbf{a}]_i$ is the i_{th} element of vector $\mathbf{a}$, while $[\mathbf{A}]_{i,j}$ is the $(i,j)_{\text{th}}$ element of matrix $\mathbf{A}$. All operations on indices in this dissertation are taken modulo N, denoted by $[\cdot]_N$. We use $\lfloor\cdot\rceil$ to denote rounding to the nearest integer. $[S]$ refers to the set of indices $\{0, ..., S-1\}$. The cardinality of set $\mathbb{S}$ is denoted as $|\mathbb{S}|$. We use $\{0,1\}^B$ to denote the set of B-dimensional binary vectors. We use $\text{diag}(\mathbf{v})$ to denote forming a diagonal matrix from the vector $\mathbf{v}$ and use $\mathbb{E}\{\cdot\}$ to denote expectation. The DFT of signal $\mathbf{x}$ is denoted as $\hat{\mathbf{x}}$. We denote the least common multiple of N_0, N_1 as $\text{LCM}(N_0, N_1)$.

Chapter 2

Robust Sparse Fourier Transform

In this chapter, we propose the Robust Sparse Fourier Transform (RSFT), a tool that enables the application of the Sparse Fourier Transform (SFT) to a real world, noisy setting. The RSFT can accommodate off-grid frequencies in the data. Furthermore, by incorporating Neyman-Pearson (NP) detection in the SFT stages, frequency detection in the RSFT does not require knowledge of the exact sparsity of the signal and is robust to noise. We analyze the asymptotic performance of the RSFT, and study the computational complexity versus detection performance tradeoff. We show that by appropriate choice of the detection thresholds, the optimal tradeoff can be achieved.

2.1 Introduction

Simple and practical SFT algorithm of [4] achieves a substantial computational reduction of computing DFT compared with the FFT and meanwhile provides a robust frequency localization based on the property of modular inverse that is less affected by the noise compared with the SFT algorithms (e.g., [21, 5]), whose frequency localization relies on the phase information. Such property of simple and practical SFT is attractive in the radar application. However, simple and practical SFT has two main constraints:

- It assumes that the significant frequencies contained in the signal are on the grid of the N-point DFT, where N is the original input data length.
- The detection of the significant frequencies in the two detection stages of simple and practical SFT algorithm assumes the exact knowledge of the number of the significant frequencies, K.

In real-life radar applications, however, the signal frequencies that relate to the target parameters (e.g., rang, velocity and DOA) are typically off-grid; also, the number of significant frequencies, representing the number of the targets in the radar beam coverage, is typically unknown and subject to change. The consequence of off-grid frequencies is the leakage to other frequency bins, which essentially reduces the sparsity of the signal. The incomplete knowledge of the number of significant frequencies makes the frequency detection difficult. Also, the off-grid frequencies further complicate frequency detection; when the dynamic range of the signal frequencies is high, which is typical in the radar application, the leakage from strong frequencies can mask weak frequencies; thus, even if the exact sparsity of the signal was known a priori, it would not help the task of frequency detection.

We follow simple and practical SFT framework for frequency-domain sparse signal detection and propose RSFT, which addresses the aforementioned limitations of simple and practical SFT. RSFT employs a *pre-permutation window* to reduce leakage from the off-grid frequencies, and NP detection to determine the detection thresholds for the signal frequencies. We also extended the one-dimensional RSFT into multidimensional, which makes the RSFT suitable for multidimensional signal processing applications.

The contributions in the context of the proposed RSFT are summarized as follows:

- The proposed RSFT algorithm does not need the frequencies to be on-grid. Also, it does not require exact knowledge of the number of frequencies to be estimated. The leakage due to off-grid frequencies, which typically reduces the sparsity of the signal, is controlled via a simple windowing operation.

- Frequency detection is put in an NP detection framework. Based on the signal model and other design specifications, we provide (asymptotically) optimal thresholds for the two detection stages of RSFT. Since the output of the first stage of detection serves as the input of the second stage, the two stages are interconnected. The detection thresholds are jointly found by formulating and solving a constrained optimization problem. The objective function maximizes the probability of detection for the weakest frequency contained in the signal, and the constraints connect the probability of detection and false alarm rate for both two stages.

- A quantitative measure of the computational complexity and detection performance tradeoff is provided, which can serve as a concrete design reference for RSFT-based applications.

2.2 Signal Model and Overview of the RSFT Algorithm

We model a continuous-time signal as a superposition of K sinusoids in additive white noise. We sample the signal uniformly, both in I and Q channels, with a sampling frequency above the Nyquist rate. We segment the obtained samples into T consecutive equal length segments, each containing N samples. We choose N so that $K << N$. The sampled signal over the time segment $s = 0, 1, \cdots, T-1$, can be expressed in vector form as

$$\mathbf{r}_s = \sum_{i \in [K]} b_{i,s} \mathbf{v}(\omega_i) + \mathbf{n}_s, \tag{2.1}$$

where $\mathbf{v}(\omega_i)$ denotes for the i-th complex sinusoid with normalized frequency $\omega_i \in [0, 2\pi)$, i.e.,

$$\mathbf{v}(\omega_i) = [1 \quad e^{j\omega_i} \ \cdots \ e^{j(N-1)\omega_i}]^T. \tag{2.2}$$

We further assume that ω_i is unknown, deterministic and remains constant during all T segments, while the corresponding complex amplitude, i.e., $b_{i,s}$ is a random variable. Here, over the various segments, $b_{i,s}$ is taken to be independent, identically circularly symmetric Gaussian distributed, i.e., $b_{i,s} \sim \mathcal{CN}(0, \sigma_{bi}^2)$. The coefficients of different sinusoids are taken to be independent. The $\mathbf{n}_s$ represents noise, and is distributed as $\mathbf{n}_s \sim \mathcal{CN}(\mathbf{0}, \sigma_n^2 \mathbf{I})$, where $\mathbf{0}$ is an N-dimensional zero vector, and $\mathbf{I} \in \mathbb{R}^{N \times N}$ is the identity matrix. We also assume that each sinusoid's amplitude and the noise are uncorrelated. The spacing of two neighboring frequencies is assumed to be greater than $\eta_m \Delta\omega_N$, where $\Delta\omega_N \triangleq 2\pi/N$, and $\eta_m \in \mathbb{N}$ is the main-lobe broadening parameter due to a window that will be applied on $\mathbf{r}_s$. Let $SNR_i = \sigma_{bi}^2/\sigma_n^2$ be the SNR of the i-th sinusoid. The worst case SNR, best case SNR and dynamic range are respectively defined as $SNR_{min} \triangleq \min_{i \in [K]}(SNR_i), SNR_{max} \triangleq \max_{i \in [K]}(SNR_i), D \triangleq SNR_{max}/SNR_{min}$.

Estimating the frequencies in $\mathbf{r_s}$ is a classical spectral analysis problem. In non-parametric techniques, this is done by employing the DFT [39]. The DFT of length N is a sampled version of the Discrete-Time Fourier Transform of the signal, with sampling interval $\Delta\omega_N$.

A normalized frequency ω_i in $\mathbf{r_s}$ corresponds to a DFT peak at index $k = [\lfloor \frac{\omega_i N}{2\pi} \rceil]_N$. The presence of a frequency in the DFT can be determined using an NP test [40]. The DFT can be effectively computed by the FFT, whose complexity for N equal to a power of 2 is $O(N \log N)$. However, when N is large, the computation of the FFT is still demanding especially in a real-time processing scenarios.

To apply the SFT in a realistic scenario related to detection and estimation of multiple frequencies, we propose the RSFT algorithm, which employs a pre-permutation windowing procedure to suppress leakage from the off-grid frequencies and leverages the NP criterion in the two stages of detection to solve for the optimal thresholds. The RSFT algorithm is summarized in Algorithm 1.

Algorithm 1 RSFT algorithm

```
**Input:** complex signal $\mathbf{r}_s, s \in [T]$
**Output:** frequency locations of input signal, $\mathbf{o} \in \{0,1\}^N$
procedure RSFT($\mathbf{r}_s$)
    Generate a set of permutation parameters $\sigma_s, s \in [T]$ randomly
    $\bar{\mathbf{a}} \leftarrow 0$
    for $s \leftarrow 0$ to $T-1$ do
        Apply pre-permutation windowing: $\mathbf{y}_s \leftarrow \mathbf{W}\mathbf{r}_s$
        Apply permutation: $\mathbf{p}_s \leftarrow \mathbf{P}_{\sigma_\mathbf{s}}\mathbf{y}_\mathbf{s}$
        Apply flat-windowing: $\mathbf{z}_s \leftarrow \overline{\mathbf{W}}\mathbf{p}_s$
        Aliasing: $\mathbf{f}_s \leftarrow \text{Aliasing}(\mathbf{z}_s)$
        Take $B$-point FFT: $\hat{\mathbf{f}}_s \leftarrow \text{FFT}(\mathbf{f}_s)$
        First stage detection: $\mathbf{c}_{\sigma_s} \leftarrow \text{NPdet1}(|\hat{\mathbf{f}}_s|^2)$
        Reverse mapping: $\mathbf{a}_{\sigma_s} \leftarrow \text{Reverse}(\mathbf{c}_{\sigma_s})$
        Accumulation: $\bar{\mathbf{a}} \leftarrow \bar{\mathbf{a}} + \mathbf{a}_{\sigma_s}$
    end for
    Second stage detection: $\mathbf{o} \leftarrow \text{NPdet2}(\bar{\mathbf{a}})$
    return $\mathbf{o}$
end procedure
```

2.3 Leakage Suppression of Off-grid Frequencies

In real world applications, the significant frequencies ω_i are continuous and can take any value in $[0, 2\pi)$. When fitting a grid on these frequencies, leakage occurs from off-grid frequencies, which can diminish the sparsity of the signal. Sparsity in the discrete frequency domain related to N-point DFT is here defined as follows. On applying some linear operations (e.g., permutation and windowing), represented by $\mathbf{L} \in \mathbb{C}^{N\times N}$, on the signal of (2.1), the resulting spectrum equals $\mathbf{x} = \mathbb{E}\{|\widehat{\mathbf{Lr}}|^2\} \in \mathbb{R}^N$. Let $[\mathbf{x}]_i$ correspond to the peak of the sinusoid with SNR_{min}, i.e., the weakest sinusoid. Sparsity is defined as the number of entries in $\mathbf{x}$, whose values are greater than or equal to $[\mathbf{x}]_i$.

As the leakage due to strong frequency components can mask the contributions of weak frequency components, it is difficult to determine the frequency domain peaks after permutation. This is illustrated in Figs. 2.1 (a) and (b), where the amplitude of a 1024-point DFT of the signal before and after permutation, respectively, is shown. The original signal contains two significant frequencies located at $\omega_1 \approx 1.8$ and $\omega_2 \approx 4.2$, and the former is $35dB$ stronger than the latter. The permutation does not alter the sparsity of the signal, which equals to 30. However, the originally concentrated leakage frequencies are shuffled by permutation, which makes it difficult to detect weaker significant frequencies. To address this problem, we propose to multiply $\mathbf{r}$ point-wisely with a window $\mathbf{w} \in \mathbb{C}^{\mathbf{N}}$ before permutation. We will refer to this step as pre-permutation windowing. Windowing suppresses side-lobes, thus can confine the leakage within a small number of frequency bins, and thus preserving sparsity to some extent. As shown in Figs. 2.1 (c) and (d), after multiplying the signal before permutation with a Dolph-Chebyshev window[41] with $77dB$ peak to side-lobe ratio (PSR), the sparsity of the signal reduces from 30 to 6.

After applying the pre-permutation window, the side-lobes of the strongest frequency should be lower than the noise level by $\delta_{\mathbf{w}}$ (e.g., $\delta_{\mathbf{w}} = 20dB$), so that their contribution to the noise level can be neglected. Hence, the PSR of the pre-permutation window $\mathbf{w}$ should satisfy

$$p_{\mathbf{w}} = 10\log(N) + 10\log(D) + 10\log(SNR_{min}) + \delta_{\mathbf{w}}, \tag{2.3}$$

where the first term of the right side is due to the N-point DFT, and the unit of $p_{\mathbf{w}}, \delta_{\mathbf{w}}$ is dB.

For a specific window, determining the PSR also determines the main-lobe broadening parameter η_m. Here, η_m is defined as the number of points whose magnitudes are larger than the maximum side-lobe level in the magnitude of N-point DFT of the pre-permutation window, i.e., $|\hat{\mathbf{w}}|$. The larger the PSR of a window, the larger the η_m is. The DFT of a Dolph-Chebyshev window and its main-lobe broadening parameter η_m corresponding to various PSR are shown in Fig. 2.2.

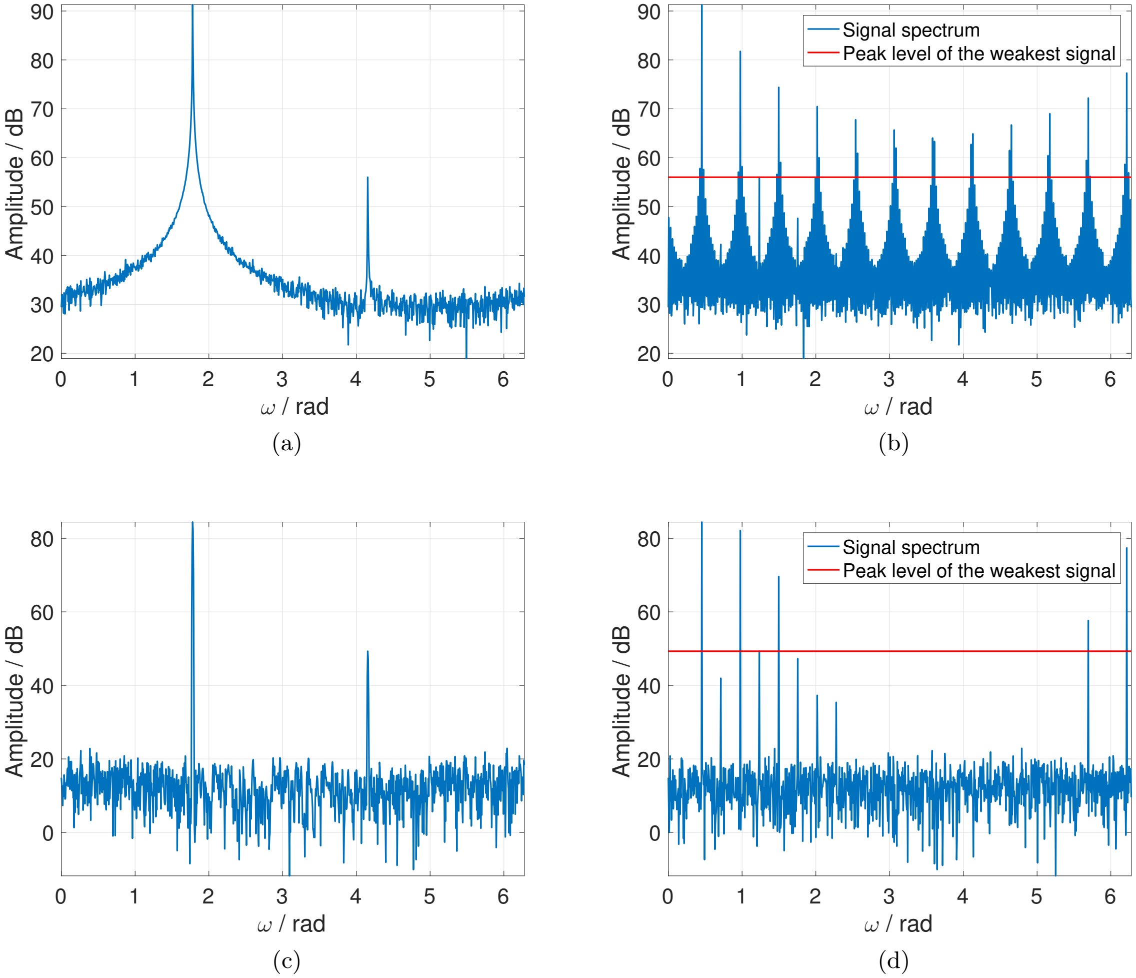
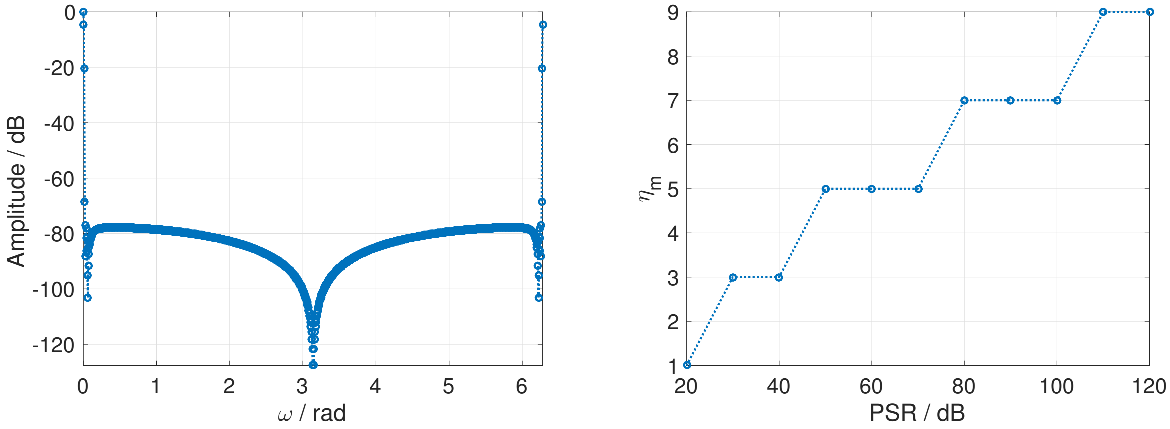


Figure 2.1: The effect of pre-permutation windowing on sparsity of the signal in the discrete frequency domain. The signal contains two significant frequency components, one of which is $35dB$ stronger than the other; $SNR_{min} = -8dB$. A Dolph-Chebyshev window with $77dB$ PSR is applied. Windowed signal after permutation appears sparser in the frequency domain as compared to the permuted signal without windowing. The spectrum is computed via 1024-point DFT. (a) Spectrum of signal without windowing. (b) Spectrum of signal without windowing after permutation with $\sigma_s = 85$. (c) Spectrum of windowed signal. (d) Spectrum of windowed signal after permutation with $\sigma_s = 85$.



(a)

(b)

Figure 2.2: Pre-permutation window. (a) 1024-point DFT of a Dolph-Chebyshev window with $77dB$ PSR. (b) Main-lobe broadening parameter η_m versus the PSR for Dolph-Chebyshev windows with 1024-points DFT.

2.4 NP Detection in the RSFT

In simple and practical SFT, detection of the significant frequencies is needed in two stages. If we know the number of the significant frequencies and all frequencies are all on-grid, the detection of the signal can be accomplished by finding the K highest spectral amplitude values. In reality, however, we do not have exact knowledge of K. Moreover, even if we knew K, due to the leakage caused by the off-grid frequencies, the K highest spectral peaks might not be correct representation of the signal frequencies. Finally, additive noise would make signal detection even more difficult.

In order to solve the detection problem, we propose to use NP detection in the two detection stages of RSFT. The proposed detection scheme does not require knowledge of K, but instead, it uses a bound of the signal sparsity K_{max}. Specifically, the NP detection problem related to the design of the RSFT parameters is described next.

We consider the signal model of (2.1). For the given data length N, number of iterations T, SNR corresponding to the weakest frequency SNR_{min}, false alarm rate of each frequency bin in the second stage detection P_{fa}, sparsity bound K_{max}, reduced data length B, noise variance σ_n^2, and signal dynamic range D, we shall design the pre-permutation window $\mathbf{w}$, flat-window $\bar{\mathbf{w}}$, the thresholds γ, μ for the first and the second stage of detection, respectively, so that the probability of detection for the weakest frequency in the second stage P_d is maximized. We should note that assuming knowledge of noise variance, minimum SNR and dynamic range is not a very restrictive assumption for real systems. For instance, in the radar case, σ_n^2 corresponds to the receiver noise; SNR_{min} (SNR_{max}) of the sinusoids can be calculated by the smallest (largest) Radar Cross Section (RCS) of targets at the maximum (minimum) detection range. The RCS and range specifications are among the most import specifications of designing radar systems, and are usually provided.

Applying NP detection in the RSFT is not a straightforward extension of simple and practical SFT, in that the two stages are inter-connected, thus need to be studied jointly. In the following, we investigate the two stages of detection separately, then summarize the solution into an optimization problem.

2.4.1 The First Stage Detection

The first stage detection is performed on each data segment. After pre-permutation windowing, permutation and flat-windowing, the input signal can be expressed as

$$\mathbf{z}_s = \overline{\mathbf{W}}\mathbf{P}_{\sigma_s}\mathbf{W}\mathbf{r}_s, \tag{2.4}$$

where $\sigma_s \in \mathbb{S}_p \triangleq \{2k+1|k \in [N/2]\}$ is the permutation parameter for the s-th data segment; we assume that σ_s has an uniform random distribution. $\mathbf{P}_{\sigma_s} \in \{0,1\}^{N\times N}$ is the permutation matrix; $\mathbf{W} = \text{diag}(\mathbf{w})$, $\overline{\mathbf{W}} = \text{diag}(\bar{\mathbf{w}})$, where $\mathbf{w}, \bar{\mathbf{w}} \in \mathbb{C}^N$ are pre-permutation window and flat-window, respectively. The flat-window $\bar{\mathbf{w}}$ is defined so that its passband has width $2\pi/B$, and its time duration is N samples.

The time domain aliasing can be described as

$$\mathbf{f}_s = \sum_{i=0}^{L-1} \overline{\mathbf{W}}_i\mathbf{P}_{\sigma_s}\mathbf{W}\mathbf{r}_s = \mathbf{V}_{\sigma_s}\mathbf{r}_s, \tag{2.5}$$

where $L = N/B$; $\overline{\mathbf{W}}_i$ is the i-th sub-matrix of $\overline{\mathbf{W}}$, which consists of the iB-th to the $((i+1)B-1)$-th rows of $\overline{\mathbf{W}}$; and $\mathbf{V}_{\sigma_s} = \sum_{i=0}^{L-1}\overline{\mathbf{W}}_i\mathbf{P}_{\sigma_s}\mathbf{W}$.

The B-point DFT operation on the aliased data $\mathbf{f}_s$ can be expressed as

$$\hat{\mathbf{f}}_s = \mathbf{D}\mathbf{V}_{\sigma_s}\mathbf{r}_s, \tag{2.6}$$

where $\mathbf{D} \in \mathbb{C}^{B\times B}$ is the DFT matrix. For the k-th entry of $\hat{\mathbf{f}}$, we have

$$[\hat{\mathbf{f}}_s]_k = \mathbf{u}_k^H \mathbf{V}_{\sigma_s} \mathbf{r}_s,\ k = 0, 1, \cdots, B-1, \tag{2.7}$$

where $\mathbf{u}_k$ is the k-th column of $\mathbf{D}$, i.e., $\mathbf{u}_k = [1 \quad e^{-jk\frac{2\pi}{B}} \ \cdots \ e^{-jk(B-1)\frac{2\pi}{B}}]^T$.

Substituting (2.1) into (2.7), we get

$$[\hat{\mathbf{f}}_s]_k = \sum_{i\in[K]} b_{i,s} \mathbf{u}_k^H \mathbf{V}_{\sigma_s} \mathbf{v}(\omega_i) + \mathbf{u}_k^H \mathbf{V}_{\sigma_s} \mathbf{n}. \tag{2.8}$$

Since $[\hat{\mathbf{f}}_s]_k$ is a linear combination of $b_{i,s}, [\mathbf{n}]_j, i \in [K], j \in [N]$, it holds that

$$[\hat{\mathbf{f}}_s]_k \sim \mathcal{CN}(0, \sigma_{fk}^2), \tag{2.9}$$

where

$$\sigma_{fk}^2 = \sum_{i\in[K]} \sigma_{bi}^2 \alpha(k, \sigma_s, \omega_i) + \sigma_n^2 \beta(\sigma_s), \tag{2.10}$$

and

$$\begin{aligned} \alpha(k, \sigma_s, \omega) &= |\mathbf{u}_k^H \mathbf{V}_{\sigma_s} \mathbf{v}(\omega)|^2 \\ \beta(\sigma_s) &= \|\overline{\mathbf{W}} \mathbf{P}_{\sigma_s} \mathbf{w}\|^2. \end{aligned} \tag{2.11}$$

It is easy to see that σ_{fk}^2 is summation of weighted variance due to each signal and noise component, with α, β the corresponding weights.

Without loss of generality, let us assume that ω_m is the frequency corresponding to the minimum SNR, i.e., $SNR_m \equiv SNR_{min}$. After pre-permutation windowing, permutation, flat-windowing, aliasing and DFT, ω_m is mapped to bin p, given in the following lemma.

Lemma 1. *For a complex sinusoid signal, i.e.,* $\mathbf{v}(\omega)$*, after pre-permutation windowing, permutation with* σ_s*, flat windowing, aliasing and* B*-point DFT, the highest amplitude of signal spectrum appears in* $[B]$ *at location*

$$p(\omega, \sigma_s) = [\lfloor \frac{B}{N} [\sigma_s [\lfloor \frac{\omega}{\Delta\omega_N} \rceil]_N]_N \rceil]_B. \tag{2.12}$$

Please see the proof in Appendix A. A visualization of this process is shown in Fig. 2.3.

Due to the signal sparsity in the frequency domain, with high probability only ω_m maps to bin p. Also, due to the two stages of windowing, the side-lobes from the strong frequencies

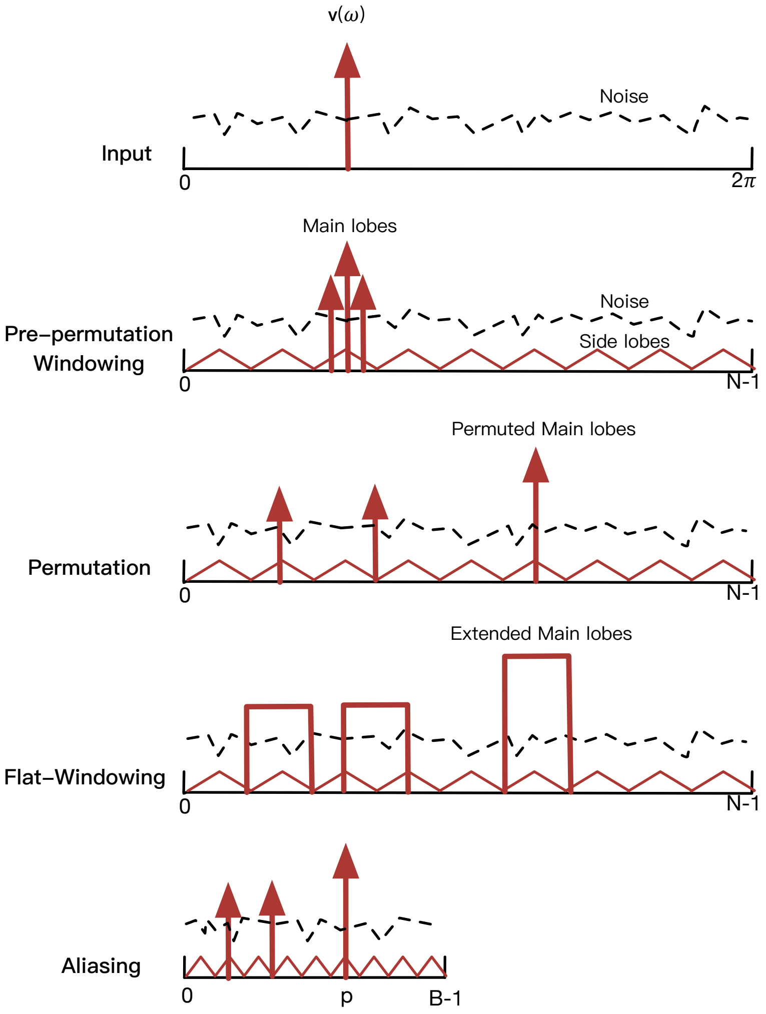


Figure 2.3: Windowing, permutation and aliasing. The frequency representation of the signal from pre-permutation windowing to aliasing is presented. Only one significant frequency is shown for conciseness.

will be below the noise level. Thus, the effect of leakage from other sinusoids can be ignored. Then we can approximate the variance of $[\hat{\mathbf{f}}]_p$ as

$$\sigma_{fp}^2 \approx \sigma_{bm}^2 \alpha(p, \sigma_s, \omega_m) + \sigma_n^2 \beta(\sigma_s). \tag{2.13}$$

In cases in which multiple frequencies are mapped to the same bin (collision), (2.13) gives a underestimate of the variance. The probability of a collision occurring reduces as $K << B$.

The bin $u \in [B]$, to which no significant frequency is mapped, contains only noise, and the corresponding variance for $[\hat{\mathbf{f}}]_u$ is

$$\sigma_{fu}^2 \approx \sigma_n^2 \beta(\sigma_s). \tag{2.14}$$

Hence, the hypothesis test for the first stage detection on $[\hat{\mathbf{f}}]_j, j = 0, 1, \cdots, B-1$ is formulated as

- $\mathcal{H}0$: no significant frequency is mapped to bin j.
- $\mathcal{H}1$: at least one significant frequency is mapped to bin j, whose SNR is at least SNR_{min}.

The log likelihood ratio test (LLRT) is

$$\log \frac{P_{f_j|H1}(x)}{P_{f_j|H0}(x)} \underset{\mathcal{H}0}{\overset{\mathcal{H}1}{\gtrless}} \gamma'. \tag{2.15}$$

where $P_{f_j|H1}(x)$ and $P_{f_j|H0}(x)$ are the probability density function (PDF) of $[\hat{\mathbf{f}}]_j$ under $\mathcal{H}1$ and $\mathcal{H}0$ respectively, and γ' is a threshold.

Substituting the PDF of $[\hat{\mathbf{f}}]_j$ under both hypothesis into (2.15), and after some manipulations we get

$$|[\hat{\mathbf{f}}_s]_j|^2 \underset{\mathcal{H}0}{\overset{\mathcal{H}1}{\gtrless}} \frac{\gamma' - \log \frac{\sigma_{fu}^2}{\sigma_{fp}^2}}{\frac{1}{\sigma_{fu}^2} - \frac{1}{\sigma_{fp}^2}}. \tag{2.16}$$

Hence, $|[\hat{\mathbf{f}}]_j|^2$ is a sufficient statistics for the first stage detection. Since $[\hat{\mathbf{f}}]_j$ has circularly symmetric Gaussian distribution, $|[\hat{\mathbf{f}}]_j|^2$ is exponentially distributed with cumulative distribution function (CDF)

$$F_{|[\hat{\mathbf{f}}_s]_j|^2}(x, \zeta^2) = \begin{cases} 1 - e^{-\frac{x}{\zeta^2}},\ x \geq 0 \\ 0,\ x < 0\ , \end{cases} \tag{2.17}$$

where ζ^2 equals to $\sigma_{f_u}^2$ under $\mathcal{H}0$, and $\sigma_{f_p}^2$ under $\mathcal{H}1$.

Based on (2.17), in the first stage of detection, the false alarm rate on each of B bins and the probability of detection of the weakest sinusoid can be derived to be equal to

$$\begin{aligned} \tilde{P}_{fa}(\sigma_s) &= e^{-\frac{\gamma}{\sigma_n^2 \beta(\sigma_s)}}, \\ \tilde{P}_d(\omega_m, \sigma_s) &= \tilde{P}_{fa}^{\frac{\beta(\sigma_s)}{\alpha(p,\omega_m,\sigma_s)SNR_{min}+\beta(\sigma_s)}}, \end{aligned} \tag{2.18}$$

where γ is the detection threshold. Both $\tilde{P}_{fa}$ and $\tilde{P}_d$ depend on the permutation σ_s.

Since $\sigma_s \in \mathbb{S}_p$ has a uniform distribution, and $|\mathbb{S}_p| = N/2$, by taking expectation with respect to σ_s of both sides of (2.18), we have

$$\begin{aligned} \bar{P}_{fa} &= \frac{2}{N} \sum_{\sigma_s \in \mathbb{S}_p} e^{-\frac{\gamma}{\sigma_n^2 \beta(\sigma_s)}}, \\ \bar{P}_d(\omega_m) &= \frac{2}{N} \sum_{\sigma_s \in \mathbb{S}_p} \tilde{P}_{fa}^{\frac{\beta(\sigma_s)}{\alpha(p,\omega_m,\sigma_s)SNR_{min}+\beta(\sigma_s)}}. \end{aligned} \tag{2.19}$$

2.4.2 The Second Stage Detection

Let $\mathbf{c}_{\sigma_s} \in \{0,1\}^B$ denote the output of the first stage detection for the s_{th} segment, with permutation parameter σ_s. Each entry in $\mathbf{c}_{\sigma_s}$ is a Bernoulli random variable, i.e., for $j = 0, 1, \cdots, B-1$,

$$[\mathbf{c}_{\sigma_s}]_j \sim \begin{cases} \text{Bernoulli}\left(\tilde{P}_{fa}(\sigma_s)\right), \textit{under } \mathcal{H}0, \\ \text{Bernoulli}\left(\tilde{P}_d(\omega_m, \sigma_s)\right), \textit{under } \mathcal{H}1. \end{cases} \tag{2.20}$$

Note that under $\mathcal{H}1$, we assume that $[\mathbf{c}_{\sigma_s}]_j$ corresponds to the weakest sinusoid. For the other $K-1$ co-existing sinusoids, since their SNR may be greater than SNR_{min}, their probability of detection may also be greater than $\tilde{P}_d(\omega_m, \sigma_s)$ (see Lemma 3).

The reverse-mapping stage hashes the $\mathbf{c}_{\sigma_s} \in \{0,1\}^B$ back to the $\mathbf{a}_{\sigma_s} \in \{0,1\}^N$. According to Definition 4, it holds that

$$[\mathbf{a}_{\sigma_s}]_i = [\mathbf{c}_{\sigma_s}]_j, \ i \in [N], j \in [B], i \in \mathcal{R}(j, \sigma_s^{-1}). \tag{2.21}$$

After accumulation of T iterations, each entry in the accumulated output is summation of T Bernoulli variables with different success rate. Define $\bar{\mathbf{a}}$ as the accumulated output, then for its i-th, $i \in [N]$ entry, we have

$$[\bar{\mathbf{a}}]_i = \sum_{s=0}^{T-1} [\mathbf{a}_{\sigma_s}]_i = \sum_{i \in \mathcal{R}(j, \sigma_s^{-1}), s \in [T]} [\mathbf{c}_{\sigma_s}]_j. \tag{2.22}$$

Note that in (2.22), each term inside the sum corresponds to a different segment, i.e., $[\mathbf{c}_{\sigma_s}]_j$ is from the s-th segment. Since σ_s is drawn randomly for each segment, j may take different values, and relates to i via a reverse-mapping. Fig. 2.4 gives a graphical illustration of the mapping and reverse-mapping.

Now, the hypothesis test for the second stage detection on $[\bar{\mathbf{a}}]_i, i \in [N]$ is formulated as

- $\overline{\mathcal{H}}0$: no significant frequency exists in bin i.
- $\overline{\mathcal{H}}1$: there exists a significant frequency in bin i, whose SNR is at least SNR_{min}.

In the following, we investigate the statistics of $[\bar{\mathbf{a}}]_i$ under both hypothesis in an asymptotic senses. Before that however, we will take a closer look at the mapping and the reverse mapping by providing the following properties.

Property 3. ***(Reversibility):*** *Let $j \in [B], i, \sigma, \sigma^{-1} \in [N]$. σ and σ^{-1} satisfy Eq. (1.46). If $j = \mathcal{M}(i, \sigma)$, then it holds that*

$$i \in \mathcal{R}(j, \sigma^{-1}). \tag{2.23}$$

Property 4. ***(Distinctiveness):*** *Let $i, j \in [B], i \neq j$. If $\sigma^{-1} \in [N]$ and satisfies Eq. (1.46), then it holds that*

$$\mathcal{R}(i, \sigma^{-1}) \cap \mathcal{R}(j, \sigma^{-1}) = \emptyset. \tag{2.24}$$

The proofs of those properties can be found in Appendices B and C. The two properties simply reveal the following facts: a mapped location can be recovered by reverse mapping (with ambiguities). Also, when applying reverse mapping to two distinct locations with the same permutation parameter, the resulting locations are also distinct.

Under $\overline{\mathcal{H}}1$, assuming that $[\bar{\mathbf{a}}]_i$ corresponds to the m-th sinusoid, i.e., the weakest sinusoid, then each term inside the sum of (2.22) has distribution $[\mathbf{c}_{\sigma_s}]_j \sim \text{Bernoulli}\left(\tilde{P}_d(\omega_m, \sigma_s)\right), s \in [T]$. Then we present the following lemma.

Lemma 2. *Under $\overline{\mathcal{H}}1$, and as $T \to \infty$,*

$$[\bar{\mathbf{a}}]_i \sim N(\mu_{a1}(\omega_m), \sigma_{a1}^2(\omega_m)), \tag{2.25}$$

where $\mu_{a1}(\omega_m) = T\bar{P}_d(\omega_m)$, $\sigma_{a1}^2(\omega_m) \leq T\bar{P}_d(\omega_m)(1 - \bar{P}_d(\omega_m))$.

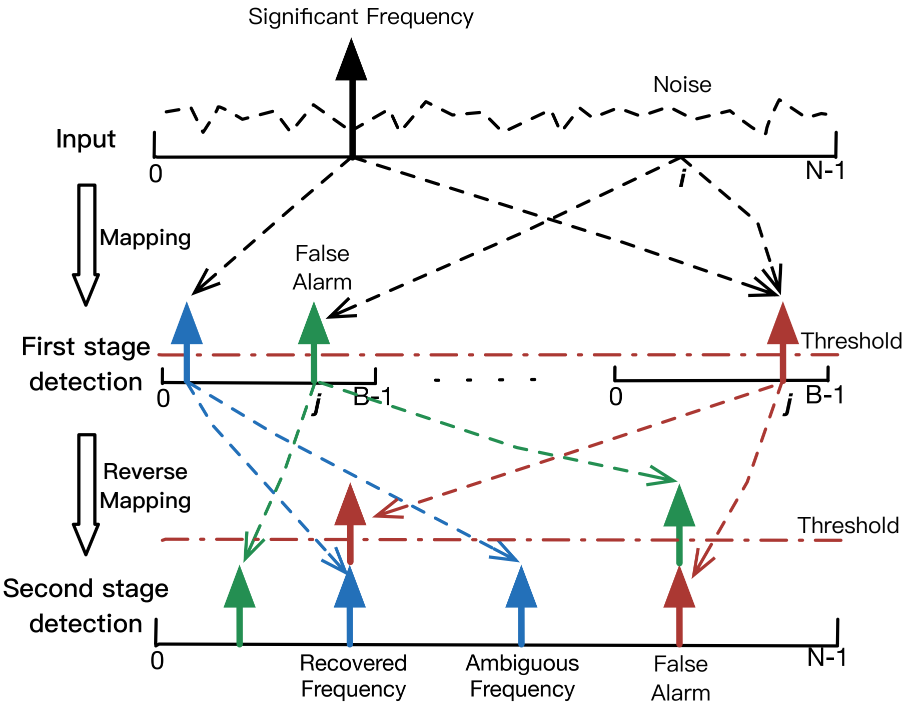


Figure 2.4: Mapping and reverse mapping. Due to the different permutations, a significant frequency may be mapped into different locations in each iteration of the first stage detection. The detected frequencies, including the false alarms in the first stage, are reverse mapped to the original discrete frequency set. The true location of the significant frequency as well as ambiguous frequencies are obtained. The occurrence on the true location grows steadily during accumulation, provided that the SNR is high enough, and thus the true location can be recovered in the second stage of detection with proper thresholding. However, false alarms may also occur in the second stage detection, due to both ambiguous frequencies and false alarms from the first stage of detection.

Please see the proof in Appendix D.

The distribution of $[\bar{\mathbf{a}}]_i$ under $\overline{\mathcal{H}}0$ is more complicated, and we have the following lemma.

Lemma 3. *Under $\overline{\mathcal{H}}0$, and as $T \to \infty$,*

$$[\bar{\mathbf{a}}]_i \sim N(\mu_{a0}(\omega_m), \sigma_{a0}^2(\omega_m)), \tag{2.26}$$

where

$$\begin{aligned} \mu_{a0}(\omega_m) &= F\eta_p \bar{P}_d(\omega_m) + (T-F)\bar{P}_{fa}, \\ \sigma_{a0}^2(\omega_m) &\leq F\eta_p \bar{P}_d(\omega_m)(1-\eta_p \bar{P}_d(\omega_m)) \\ &\quad + (T-F)\bar{P}_{fa}(1-\bar{P}_{fa}), \end{aligned} \tag{2.27}$$

and $F = \frac{TK\eta_m}{B}$; $\eta_p \in [1, \frac{1}{\bar{P}_d(\omega_m)}]$ is a calibration parameter of the probability of detection for the other $K-1$ co-existing sinusoids.

Please see the proof in Appendix E.

Remark 1. *From Lemmas 2 and 3, we notice that for the second stage detection, the LLRT is obtained based on two Normal distributions. The test statistic under $\overline{\mathcal{H}}1$ is "stable", because it only depends on $\bar{P}_d(\omega_m)$. However, under $\overline{\mathcal{H}}0$, the distribution depends on the number of co-existing sinusoids K, as well as on each sinusoid's SNR. A larger K and higher SNR will "push" the distribution under $\overline{\mathcal{H}}0$ closer to the distribution under $\overline{\mathcal{H}}1$, hence degrading the detection performance.*

A natural extension of Remark 1 is Remark 2, which gives the condition under which the RSFT will reach its limit.

Remark 2. *Assuming that $P_d \geq P_{fa}$, the RSFT will fail if $K\eta_m \geq B$ no matter how large the SNR_{min} is.*

Please see the proof in Appendix F.

2.4.3 The Optimization Problem

Based on the analysis of the two detection stages, the optimal probability of detection for the weakest frequency $P_d^*(\omega_m)$, and the optimal detection thresholds for the two stages, i.e., γ^*, μ^*, can be found as the solution of the following optimization problem:

$$
\begin{aligned}
& Maximize_{\{\gamma,\mu,\bar{P}_{fa},\bar{P}_d\}} \quad P_d(\omega_m) \\
& Subject\ to \\
& \bar{P}_{fa} = \frac{2}{N}\sum_{\sigma_s\in\mathbb{S}_p} e^{-\frac{\gamma}{\sigma_n^2\beta(\sigma_s)}} && (2.28a) \\
& \bar{P}_d(\omega_m) = \frac{2}{N}\sum_{\sigma_s\in\mathbb{S}_p} \tilde{P}_{fa}^{\frac{\beta(\sigma_s)}{\alpha(p,\omega_m,\sigma_s)SNR_{min}+\beta(\sigma_s)}} && (2.28b) \\
& P_{fa} = \int_{\mu}^{\infty} g_{a_0}(u)du && (2.28c) \\
& P_d = \int_{\mu}^{\infty} g_{a_1}(u)du && (2.28d) \\
& 0 \le \bar{P}_{fa}, \bar{P}_d, P_{fa}, P_d \le 1 && (2.28e) \\
& \gamma, \mu > 0, && (2.28f)
\end{aligned}
$$

where the first two constraints correspond to the detection for the first stage, while the second two constraints correspond to the second stage of detection. The $g_{a_0}(u), g_{a_1}(u)$ are the asymptotic PDF of $[\bar{\mathbf{a}}]_i$ under $\overline{\mathcal{H}}0$ and $\overline{\mathcal{H}}1$, respectively. We take the upper bounds of the variances in both distributions.

To solve the problem of (2.28) we can use a brute force search for the first stage threshold γ within $(0, \gamma_{max})$ with a reasonable step size, where γ_{max} corresponds to a small value of $\bar{P}_{fa}$ in (2.28a), such as $\bar{P}_{fa} = 10^{-10}$. Specifically, with γ fixed, the $\bar{P}_{fa}, \bar{P}_d$ can be solved via (2.28a) and (2.28b). Next, the second stage threshold μ can be solved by (2.28c), and then the P_d can be found by solving (2.28d). The largest value of P_d founded during the search process, i.e., P_d^*, as well as the corresponding γ^*, μ^* are the solution of (2.28).

Remark 3. *In Lemma 3, we set a parameter η_p to calibrate the distribution of $[\bar{\mathbf{a}}]_i$ under $\overline{\mathcal{H}}0$. By setting η_p as 1 or $\frac{1}{\bar{P}_d(\omega_m)}$, we can get respectively the lower and upper bound of P_d^* for the variation of SNR of other co-existing sinusoids. The upper bound of the sparsity of the signal is K_{max}, the optimal thresholds found by solving (2.28) provides the optimal thresholds for the worst case. If the actual signal sparsity were less than K_{max}, P_{fa} would be lower than the expected value, while P_d would be unchanged according to Remark 1.*

By averaging out the permutation, asymptotically, P_d^* does not depend on the permutation. However, it still depends on ω_m, as described in the following lemma.

Lemma 4. *The dependence of P_d^* on ω_m is due to the off-grid loss[41] from off-grid frequencies. P_d^* attains its maximum when ω_m is on-grid, i.e. $\omega_m = k\Delta\omega_N, k \in [N]$. When ω_m is in the middle between two grid points, i.e., $\omega_m = (k + \frac{1}{2})\Delta\omega_N$, P_d^* attains its minimum.*

Please see the proof in Appendix G.

2.5 Computational Complexity Analysis

We analyze the computational complexity of the RSFT algorithm by counting the number of operations in its main stages, which is shown in Table 2.1. The RSFT has complexity equal to

$$O\left(T(N + B + B\log B + \frac{K\eta_m N}{B\eta_p}) + N\right). \tag{2.29}$$

The FFT-based counterpart of the RSFT is the FFT-based Bartlett method followed by an NP detection [40] (see Appendix M), whose complexity is $O\left(TN(1 + \log N) + N\right)$, as shown in Table 2.2. Fig. 2.5 compares the RSFT's complexity to that of Bartlett's for various B and K. One can see that the RSFT enabled savings are remarkable when B is chosen properly. Specifically, from Fig. 2.5 one can see, the lowest complexity for K equals to $5, 50, 100$ is achieved when B equals to $32, 64, 128$, respectively. Note that the core operation in RSFT is still FFT-based, but on a reduced length. By leveraging the existing high performance FFT libraries such as FFTW [42], the implementation of the RSFT algorithm could be further improved.

Remark 4. *The complexity of RSFT is linearly dependent on $N, T, K, 1/\eta_p$ and η_m, hence it is beneficial to choose a pre-permutation window with a small η_m, provided the attenuation of the side-lobes is sufficiently large. We can also choose the optimal B from (2.29) to minimize the computation. However, there are two additional constrains for B, one is B should be a power of 2, the other is $K\eta_m \leq B$, as stated in Remark 2.*

Table 2.1: Computational Complexity of RSFT

Procedure	**Number of Operations**
Pre-permutation windowing	TN
Permutation	TN
Flat windowing	TN
Aliasing	$TB(N/B-1)$
FFT	$T\frac{B}{2}\log B$
Square	TB
First stage detection	TB
Reverse-mapping	$\frac{TK\eta_m N}{B\eta_p}$
Second stage detection	N
Complexity	$O\left(T(N+B+B\log B+\frac{K\eta_m N}{B\eta_p})+N\right)$

Table 2.2: Computational Complexity of the Bartlett Method

Procedure	**Number of Operations**
Windowing	TN
FFT	$\frac{TN}{2}\log N$
Square	TN
Detection	N
Complexity	$O\left(TN(1+\log N)+N\right)$

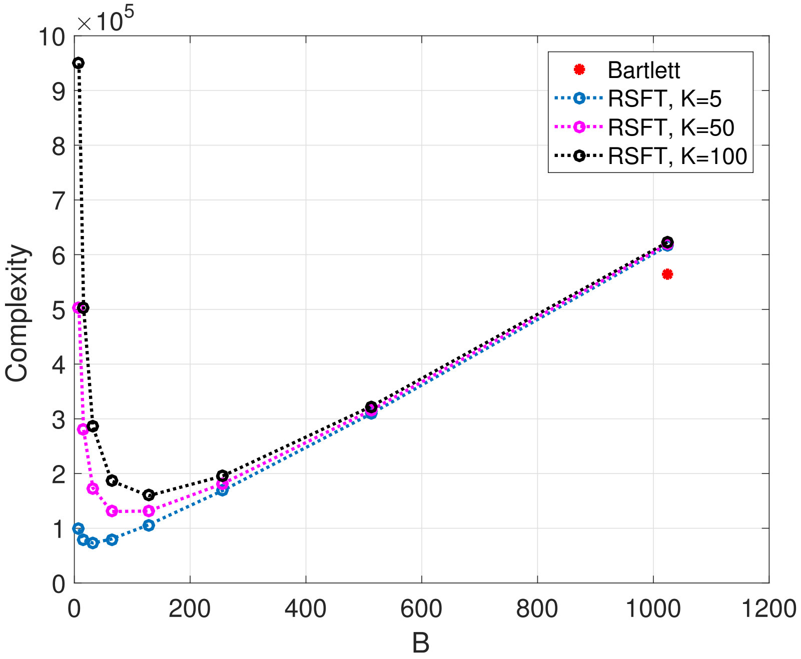


Figure 2.5: Comparison of complexity. $N = 1024, T = 50, \eta_m = 2, \eta_p = 1$, $B \in \{8, 16, 32, 64, 128, 256, 512, 1024\}$.

2.6 Multidimensional Extensions

In this section, we elaborate the extension of the RSFT into multiple dimensions; this is referred as multidimensional RSFT in the following. Compared to the multidimensional FFT, the implementation of multidimensional RSF is not straightforward, since the RSFT is not separable in each dimension due to the two stages of detection. The detection should be carried out jointly in all the dimensions, while other operations are separable in each dimension. In the following, we elaborate on multidimensional RSFT for the 2-D case.

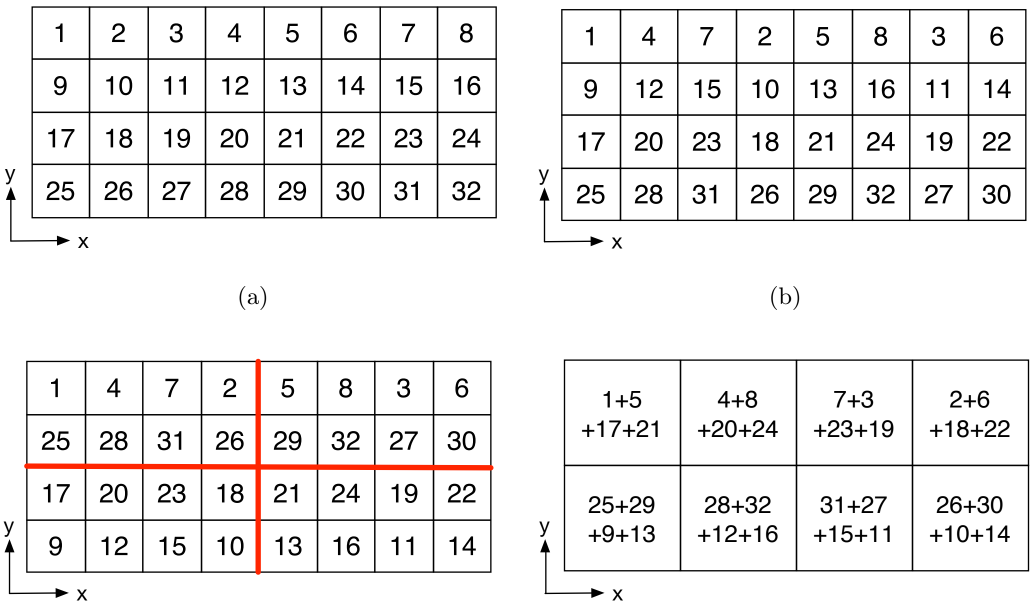
Windowing

In the pre-permutation windowing and the flat-windowing stages, the window for each dimension is designed separately. After that, the multidimensional window is generated by combining each 1-D window. For instance, in the 2-D case, assuming that $\mathbf{w}_x$ and $\mathbf{w}_y$ are the two windows in the x and y dimension, respectively, a 2-D window can be computed as

$$\mathbf{W}_{xy} = \mathbf{w}_x \mathbf{w}_y^H. \tag{2.30}$$

Permutation

The permutation parameters are generated for each dimension in a random way according to (1.46). Then, we carry the permutation on each dimension sequentially. An example for the 2-D case is illustrated in Fig. 2.6.



1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32

(a)

1	4	7	2	5	8	3	6
9	12	15	10	13	16	11	14
17	20	23	18	21	24	19	22
25	28	31	26	29	32	27	30

(b)

1	4	7	2	5	8	3	6
25	28	31	26	29	32	27	30
17	20	23	18	21	24	19	22
9	12	15	10	13	16	11	14

(c)

1+5 +17+21	4+8 +20+24	7+3 +23+19	2+6 +18+22
25+29 +9+13	28+32 +12+16	31+27 +15+11	26+30 +10+14

(d)

Figure 2.6: Permutation and aliasing in 2-D. (a) Original 2-D data forms a 4 × 8 matrix. (b) Permutation in $x-$dimension, $\sigma_x = 3$. (c) Permutation in $y-$dimension, $\sigma_y = 3$. After permutation, data is divided into four 2×4 sub-matrices. (d) Aliasing by adding sub-matrices from (c).

Aliasing

The aliasing stage compresses the high-dimensional data into much smaller size. In 2-D, as shown in Fig. 2.6, a periodic extension of the $N_x \times N_y$ data matrix is created with period B_x in the x dimension and B_y in the y dimension, with $B_x << N_x$ and $B_y << N_y$, and the basic period, i.e., $B_x \times B_y$ is extracted. Mathematically, $[\mathbf{Y}]_{\mathbf{i,j}} = \sum_{\mathbf{u=0}}^{\frac{\mathbf{N_x}}{\mathbf{B_x}}-\mathbf{1}} \sum_{\mathbf{v=0}}^{\frac{\mathbf{N_y}}{\mathbf{B_y}}-\mathbf{1}} [\mathbf{X}]_{\mathbf{i+B_xu,j+B_yv}}$, where $\mathbf{X} \in \mathbb{C}^{\mathbf{N_x \times N_y}}$, $\mathbf{Y} \in \mathbb{C}^{\mathbf{B_x \times B_y}}$ are matrices before and after aliasing, respectively.

First Stage Detection and Reverse-mapping

We carry first stage detection after taking the square of magnitude of high-dimensional DFT on the aliased data. In the 2-D case, the first stage detection is applied by comparing each data point of $|\hat{\mathbf{Y}}|^{\mathbf{2}} \in \mathbb{R}^{\mathbf{B_x \times B_y}}$ with the first stage threshold, and for those passing the thresholds, their indices, i.e., $(i, j),\ i = 0, 1, \cdots, B_x - 1,\ j = 0, 1, \cdots, B_y - 1$ are reverse-mapped to the original space, i.e., $(u, v),\ u = 0, 1, \cdots, N_x - 1,\ v = 0, 1, \cdots, N_y - 1$.

Accumulation and Second Stage Detection

As indicated in Algorithm 1, the loop from pre-permutation windowing to reverse-mapping is repeated for T iterations with random permutation. The reverse-mapped frequency locations are accumulated, which, in the 2-D case, forms $\mathbf{A} \in \mathbb{N}^{\mathbf{N_x \times N_y}}$, a matrix recording the occurrences of each frequency. Finally, the second stage is applied on $\mathbf{A}$, by comparing each of its entry with the second stage detection threshold.

2.7 Numerical Results

In this section, we verify our theoretical findings via simulations. Unless stated otherwise, the results refer to the following scenario: data length $N = 1024$, number of iterations $T = 50$, sparsity bound $K_{max} = 16$, reduced data length $B = 128$, SNR of the weakest signal $SNR_{min} = -8dB$, signal dynamic range $D = 30dB$, calibration parameter for probability of detection of co-existing frequencies $\eta_p = 1/\bar{P}_d$, false alarm rate $P_{fa} = 10^{-6}$, location of the weakest frequency $\omega_m = 64.5\Delta\omega_N \approx 0.4$, and noise variance $\sigma_n^2 = 1$. The SNR of the other three sinusoids is the same and larger than SNR_{min} by the designated dynamic range D. We adopt a Dolph-Chebyshev window with $72dB$ PSNR for pre-permutation windowing. The corresponding main-lobe broadening parameter η_m of such window is set equal to 5. The flat window is created as follows. The N-point DFT of the Dolph-Chebyshev window is convolved with a boxcar, whose width is N/B. A subsequent inverse N-point DFT is applied to extract the time domain samples of the flat window.

Fig. 2.7 shows $\alpha(p, \sigma_s, \omega_m), \beta(\sigma_s)$ (see respectively (2.11) and (2.12)), for all possible values of the permutation parameter σ_s. Recall that $\alpha(p, \sigma_s, \omega_m)$ and $\beta(\sigma_s)$ are the weights for the variance of signal and noise component in the first stage detection, respectively. Based on the figure, one can see that $\alpha(p, \sigma_s, \omega_m)$ is larger than $\beta(\sigma_s)$ for all permutations. Thus, due to the windowing, aliasing and DFT operations before the first stage of detection, the gain of the signal component is always larger than the gain of the noise component (see (2.13)).

With all the aforementioned parameters available, we are able to solve the optimal probability of detection for the weakest signal, thresholds for the two stages of detection, i.e., P_d^*, γ^*, μ^*, respectively, via (2.28). The solver executes a brute force search of the first stage threshold γ; as suggested by Fig. 2.8, each value of γ corresponds to a value of P_d, and the largest P_d corresponds to P_d^*. The obtained optimal values are $P_d^* \approx 0.52, \gamma^* \approx 6.5 \times 10^{-3}, \mu^* \approx 38.2$.

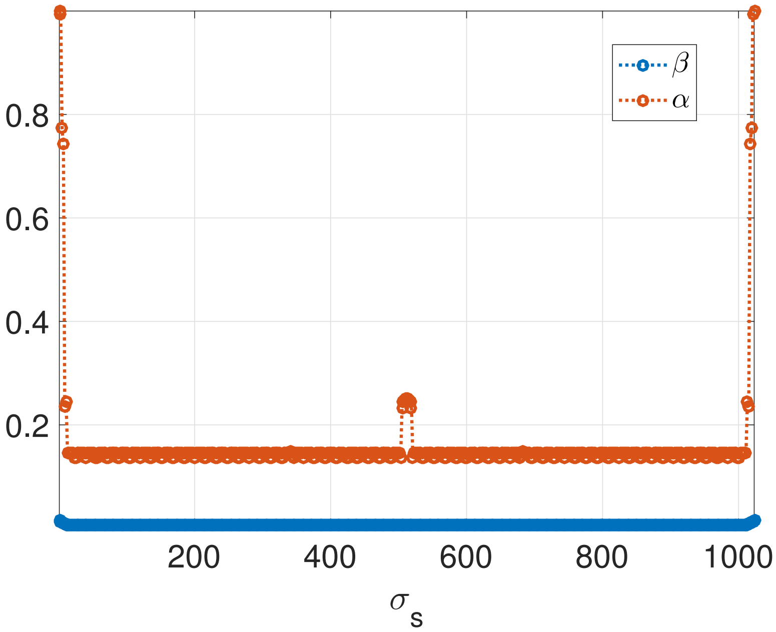


Figure 2.7: The values of $\alpha(p, \sigma_s, \omega_m), \beta(\sigma_s)$ with respect to all possible values of the permutation parameter σ_s.

To verify the effectiveness of the detection with the optimal thresholds, we simulate the specified signal and visualize the two stages of detection. The spectrum of the simulated signal after pre-permutation windowing is shown in Fig. 2.9 (a). Figs. 2.9 (b) and (c) show

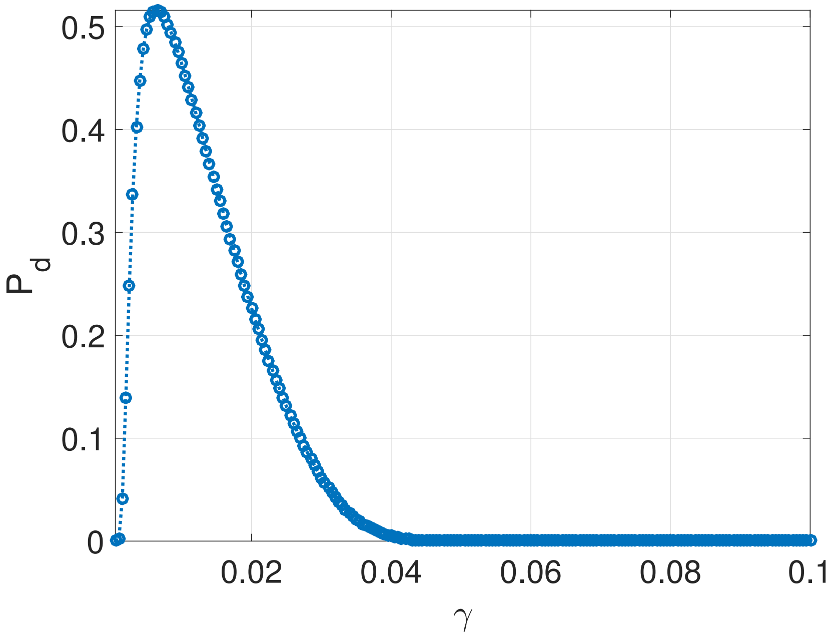
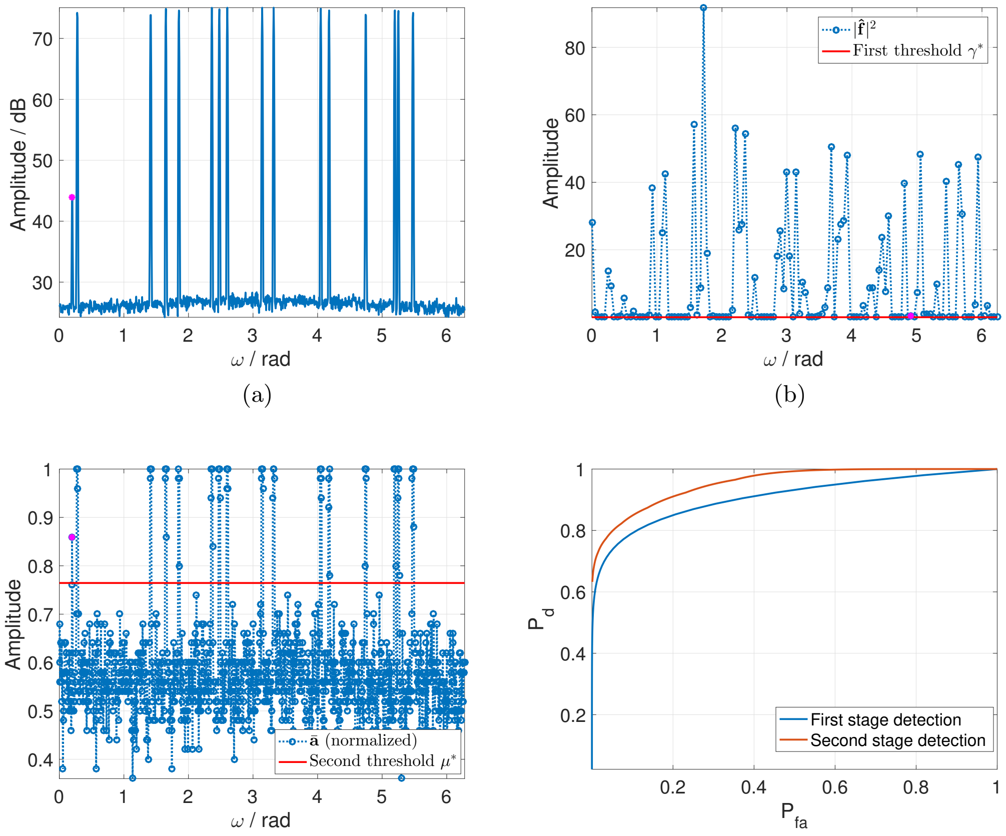


Figure 2.8: Solving for P_d^* via a brute force searching of $\gamma \in (0, 0.1)$, with a step size $\Delta\gamma = 5 \times 10^{-4}$.

that the thresholding with γ^*, μ^* respectively in the two detection stages. The locations of the detected frequencies from (c) coincide with the ground truth locations in (a), which means that the detection is effective. Fig. 2.9 (d) shows the ROC curves corresponding to the two stages of detection, respectively. Compared to the first stage, the detection performance of the second stage improves significantly, which is due to the effective accumulation stage in RSFT.

2.7.1 Improvement of Signal Sparsity by Pre-permutation Windowing

Here we verify the effectiveness of the pre-permutation window in improving the sparsity level in the discrete frequency domain. In Fig. 2.1, we visualize the improvement of the sparsity by applying a pre-permutation window on a signal that contains two significant frequencies. In Table 2.3, we show the improvement of the sparsity by pre-permutation windowing on a signal containing $K = 4$ significant frequencies with various dynamic range values D. The PSR of the pre-permutation window varies with different values of D, which



(a)

(b)

(c)

(d)

Figure 2.9: The two stages of detection. $K = K_{max} = 16$. (a) Signal spectrum after pre-permutation windowing. The magenta dot denotes the amplitude of ω_m. (b) First stage detection. (c) Second stage detection. Data and threshold are normalized by T. (d) ROC curves of the two detection stages.

is calculated by (2.3), with $\delta_{\mathbf{w}} = 20dB$. One can see that for signals with high dynamic range, the improvement of the sparsity after permutation (SAP) compared to the sparsity before permutation (SBP) is significant.

Table 2.3: Effect of Pre-permutation Window.

D (dB)	**PSR** (dB)	**SBP**	**SAP**
0	42	4	4
10	52	7	7
20	62	16	13
30	72	46	16
40	82	127	16
50	92	168	20

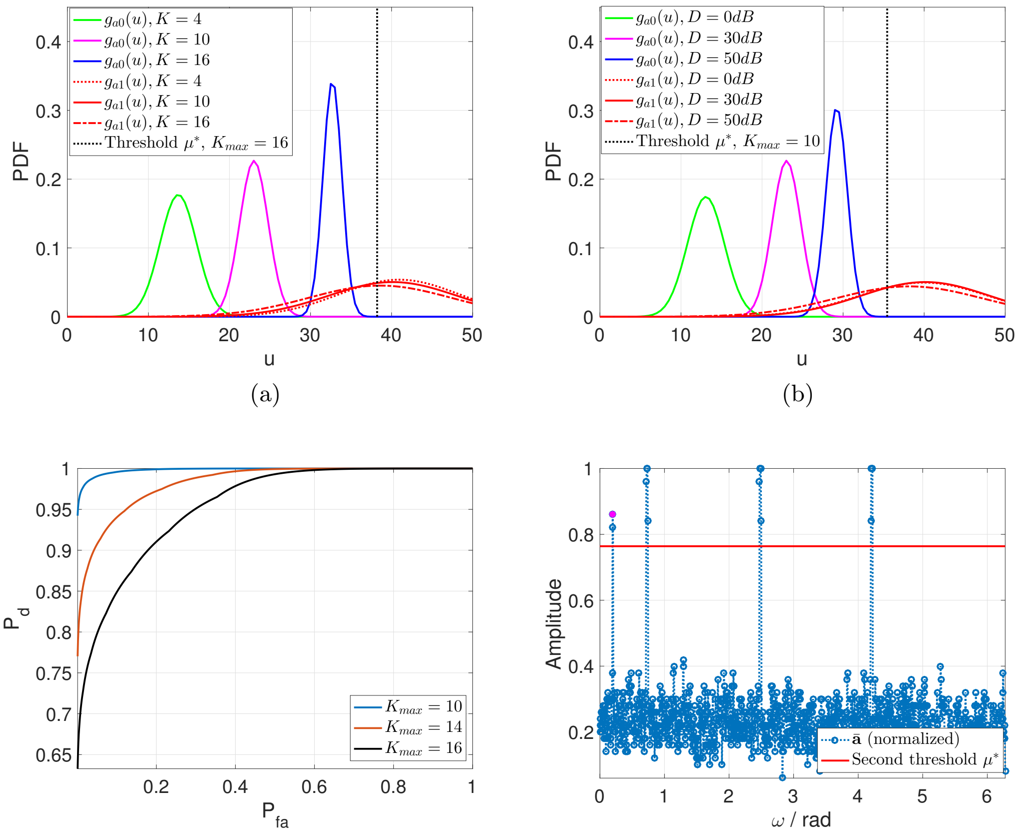
2.7.2 The Effect of Signal Sparsity and SNR of the Co-existing Frequencies on Detection Performance

According to Remarks 1 and 3, the sparsity K, the SNR of co-existing frequencies, and the dynamic range of signal D affect the detection performance for the weakest frequency. Larger values of K and D imply lower $P_d(\omega_m)$. We verify those remarks by visualizing the PDFs of two competing distributions in the second stage of detection for various values of K and D. In Fig. 2.10 (a), by fixing D, the PDF $g_{a0}(u)$ under the null hypothesis $\overline{\mathcal{H}}0$ in the second stage is evaluated for different values of K. In Fig. 2.10 (b), $g_{a0}(u)$ is calculated for different values of D with fixed K. In all cases, the change of the PDF under the alternative hypothesis $\overline{\mathcal{H}}1$, i.e., $g_{a1}(u)$, is minor, since $g_{a1}(u)$ is not directly affected by the co-existing frequencies. From these figures, one can see that large values of K and D will cause $g_{a0}(u)$ to move closer to $g_{a1}(u)$, resulting in the degradation of detection performance.

Fig. 2.10 (a) also shows that the detection is effective for unknown K, when $K \leq K_{max}$. A smaller K causes $g_{a0}(u)$ to depart from $g_{a1}(u)$. However, since the threshold μ^*, which is designed for the K_{max} case, does not change, the actual false alarm rate P_{fa} deceases, while $P_d(\omega_m)$ does not change. Using the thresholds calculated for $K_{max} = 16$ and all other parameters the same as those used in Fig. 2.9, we test a simulated signal with $K = 4$ significant frequencies to show the detection performance when $K < K_{max}$. The thresholding in the second stage detection is visualized in Fig. 2.10 (d). Compared to Fig. 2.9 (c), where

$K = K_{max} = 16$, the noise floor is much lower, hence the false alarm rate is reduced.

The detection performance related to different values of the sparsity bound K_{max} is also shown in Fig. 2.10 (c), in the form of ROC curves; the ROC curves show that, with other parameters fixed, as K_{max} increases, the detection performance degrades, which is as expected.



(a)

(b)

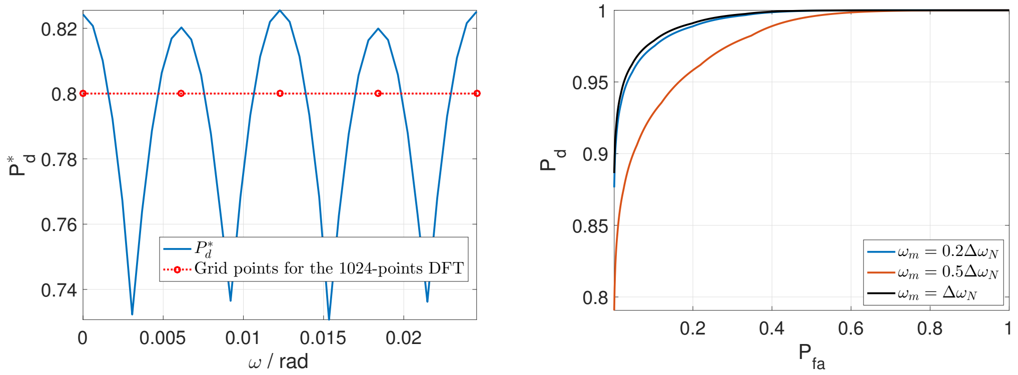
(c)

(d)

Figure 2.10: The effect of sparsity and SNR of co-existing frequencies on detection performance. (a) Changing K for $D = 30dB, \eta_p = 1/\bar{P}_d(\omega_m)$. (b) Changing D for $K = K_{max} = 10$; $\eta_p = 1$ when $D = 0dB$, $\eta_p = 1/\bar{P}_d(\omega_m)$ when $D = 30dB$ and $50dB$. (c) ROC curves for various K_{max} values. (d) The second stage detection when $K = 4, K_{max} = 16$.

2.7.3 The Effect of Frequency Locations on Detection Performance

We verify the dependence of $P_d^*(\omega_m)$ on the frequency location ω_m, and quantify the detection performance due to the off-grid loss. Fig. 2.11 (a) shows that P_d^* fluctuates periodically as ω_m varies within $[0, 4\Delta\omega_N)$. As expected from Lemma 4, P_d^* attains its maximum and minimum when ω_m is at the grid points and in the middle between the two neighboring grid points, respectively. As a result, the detection performance, as characterized by the ROC curves, also varies as ω_m varies (see Fig 2.11 (b)). According to Fig. 2.11 (a), the variation of $P_d^*(\omega_m)$ due to different locations of ω_m is within 8%. To guarantee the detection performance, a conservative design would assume that the unknown ω_m is located at the mid-point between grid points.



(a) (b)

Figure 2.11: The effect of frequency locations on detection performance. (a) The fluctuation of P_d^* due to different locations of the frequency is mainly caused by the off-grid loss. (b) ROC curves with respect to different locations of the frequency.

2.7.4 The Trade-off between Complexity and Detection Performance

In Section 2.5 we have investigated the computational complexity of RSFT, from which one can see that a smaller value of B plays a central role in reduction of complexity. However, such reduction of complexity comes at the cost of detection performance degradation. In Fig. 2.12 we show the ROC curves with respect to different choice of B, from where one can see that B affects the detection performance significantly. A large B provides better detection performance. In applying the RSFT in real-world systems, the variation of B provides a degree of freedom to trade off complexity and detection performance.

2.7.5 Comparison of RSFT and SFT

We compare RSFT and SFT in terms of their performance in localizing the significant frequencies. Clearly, since we only measure the localization error, the ℓ_2-norm metric used

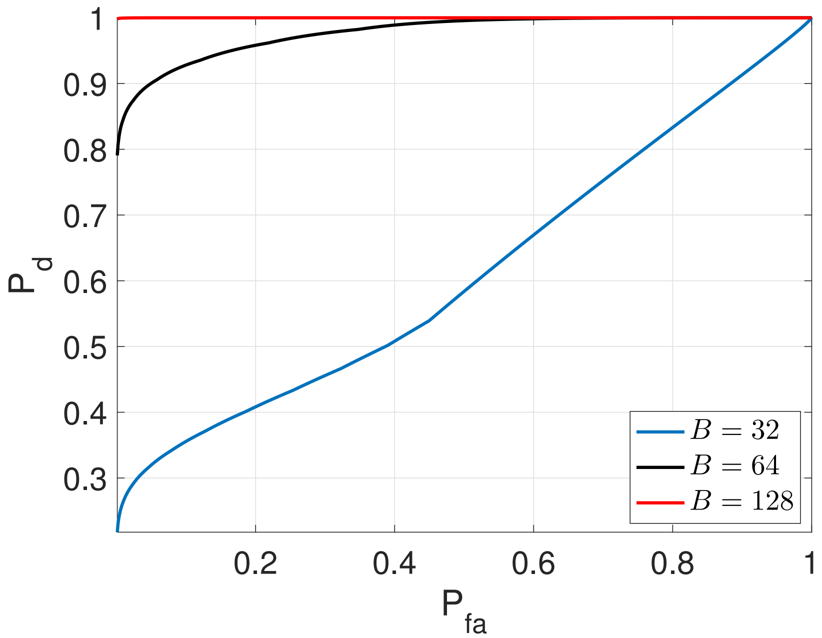


Figure 2.12: The effect of the reduced length B on detection performance when $K_{max} = 4$. A large B provides a better detection performance.

in the SFT literature for computing the approximation error between the original and the recovered signal is not a good metric. The average ℓ_1 and ℓ_∞ norm between the ground truth frequency locations and the estimated values, which are typically used in harmonic analysis literature (e.g., [29, 43]), are also not good metrics for our problem due to the following reasons: 1) the estimated frequencies typically do not exactly match the ground truth due to the grid; 2) a cluster of frequencies might appear in the vicinity of the true location of any significant frequency due to leakage and windowing; 3) there might be miss detections; and 4) there might be false alarms. In light of the above observations, we propose a metric for the localization error, defined as follows:

$$l_{err}(L) \triangleq \frac{1}{P} \sum_{\hat{\omega}_i \in \mathbb{S}_l} \frac{1}{L_i} \sum_{\tilde{\omega}_l \in \mathbb{S}_i(L)} |\hat{\omega}_i - \tilde{\omega}_l| + C_{md} n/N + C_{fa} m/N, \tag{2.31}$$

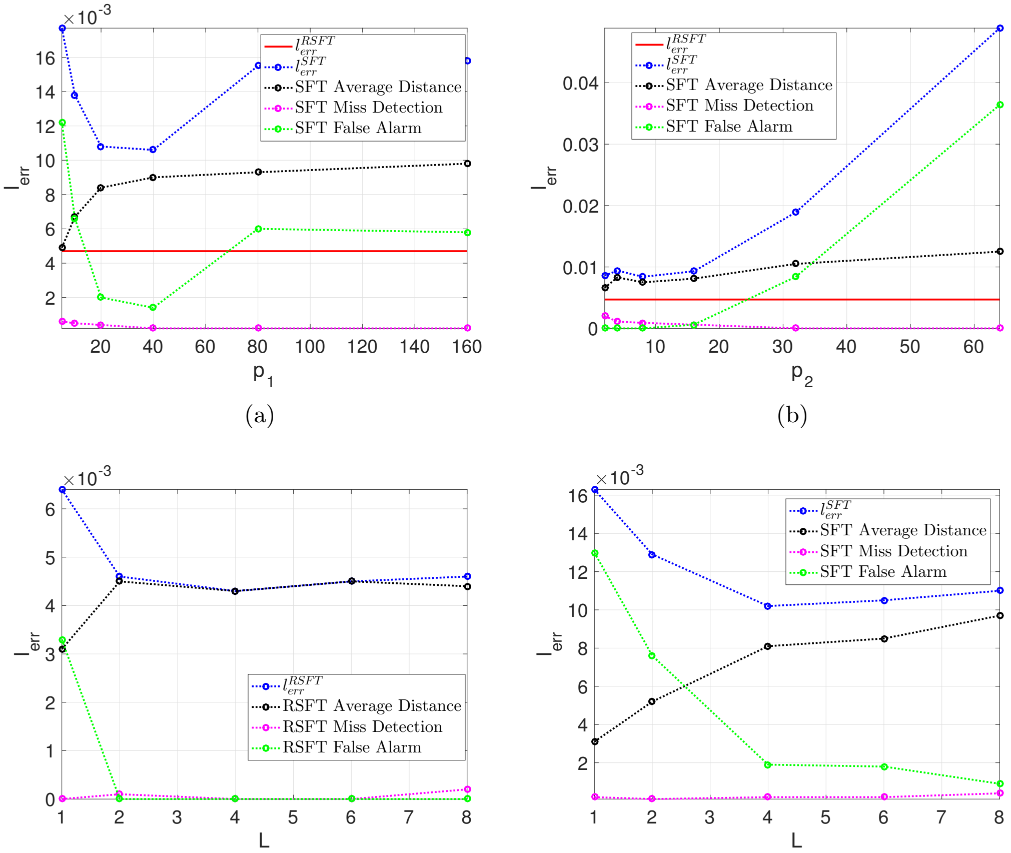
where the first, second and third terms on the right side represent the averaged ℓ_1 distance between the ground truth and the localized frequencies, the cost of miss detection, and the cost of false alarm, respectively. Specifically, for each true discretized frequency $\hat{\omega}_i = \lfloor \frac{\omega_i N}{2\pi} \rceil \in \mathbb{S}_l$, we look at the average ℓ_1 error between $\hat{\omega}_i$ and all detected frequencies $\tilde{\omega}_l \in \mathbb{S}_i(L)$ (with

$L_i = |\mathbb{S}_i(L)|$) that fall in the L-neighborhood of $\hat{\omega}_i$. P represents the number of detected true frequencies ($P \leq K$). We consider the average of all errors corresponding to all true frequencies. Missed frequencies contribute to the error via the term $C_{md}n/N$, where C_{md} is the cost parameter for miss detection and n is the number of missed true frequencies. Similarly, the false alarm frequencies contribute via the term $C_{fa}m/N$, where C_{fa} is the cost parameter for false alarm and m is the number of false frequencies. A false frequency appears outside the L-neighborhood of each $\hat{\omega}_i$.

The localization error comparison of the RSFT and the SFT is shown in Fig. 2.13, where the error is obtained by averaging 100 Monte Carlo runs. In each run, the signal is generated based on 4 randomly located significant frequencies, and the RSFT is employed to obtain the frequency set $\mathbb{S}_e^{RSFT}$ and the corresponding localization error l_{err}^{RSFT}. For the SFT-based processing, we need to determine the number of peaks to count in the two stages of detection, i.e., p_1 and p_2. Since these numbers are unknown, we test different values of p_1, p_2 and compute the corresponding l_{err}^{SFT}. The results are shown in Figs. 2.13 (a) and (b). The error components averaged ℓ_1 distance, cost of miss detection and cost of false alarm are also displayed. We observe that, while fixing L, as p_1, p_2 are increased, the cost of the averaged ℓ_1 distance increases and the cost of miss detection decreases as expected. For a fixed p_2 (p_1), there exist an optimal range for p_1 (p_2), which leads to a lower l_{err}^{SFT}. However, the lowest l_{err}^{SFT} is still larger than l_{err}^{RSFT} calculated for the same setting. The l_{err} is influenced by the half width of the vicinity window L. In Figs. 2.13 (c) and (d), we show that l_{err}^{RSFT} is stable when L exceeds 2. For l_{err}^{SFT}, as L increases, the cost from the false alarm decreases, while the averaged ℓ_1 distance increases. This is mainly due to the frequency leakage in the vicinity of a strong frequency.

2.7.6 The Variance and Its Upper Bound for $[\bar{\mathbf{a}}]_i$

The random variable under test in the second stage detection is $[\bar{\mathbf{a}}]_i, i \in [N]$. For solving (2.28), we approximate the variances of $[\bar{\mathbf{a}}]_i$ under both hypotheses by their upper bounds. In this section, we show via simulations that the actual variances are close to their upper bounds. Hence, we study the variance under $\overline{\mathcal{H}}1$, i.e., $\sigma_{a1}^2(\omega_m)$. The case for $\sigma_{a0}^2(\omega_m)$ can be similarly studied. As shown in (2), the discrepancy of $\sigma_{a1}^2(\omega_m)$ from its upper bound



(a)

(b)

(c)

(d)

Figure 2.13: Comparison of localization error for RSFT and SFT. $B = 256, C_{md} = C_{fa} = 1$. The localization error may be decomposed into averaged ℓ_1 distance, cost of miss detection and cost of false alarm. (a) The impact of p_1 on l_{err}^{SFT} when $p_2 = 20, L = 6$. (b) The impact of p_2 on l_{err}^{SFT} when $p_1 = 40, L = 6$. (c) The impact of L on l_{err}^{RSFT} when $p_1 = 40, p_2 = 20$. (d) The impact of L on l_{err}^{SFT} when $p_1 = 40, p_2 = 20$.

is due to the $\tilde{P}_d$'s dependence on σ_s, which is caused by β and α's dependence on σ_s (see Fig. 2.7). Monte Carlo simulations, presented in Fig. 2.14 indicate that the normalized approximation error, i.e., $\frac{T\bar{P}_d(\omega_m)(1-\bar{P}_d(\omega_m))-\sigma_{a1}^2(\omega_m)}{\sigma_{a1}^2(\omega_m)}$ decreases as T grows, since a larger T causes the difference of $\tilde{P}_d(\omega_m, \sigma_s)$ between different permutations to average out. Moreover, even for a small T, such as $T = 10$, one can see that the error is as small as about 1.6%, which means the approximation is reasonable.

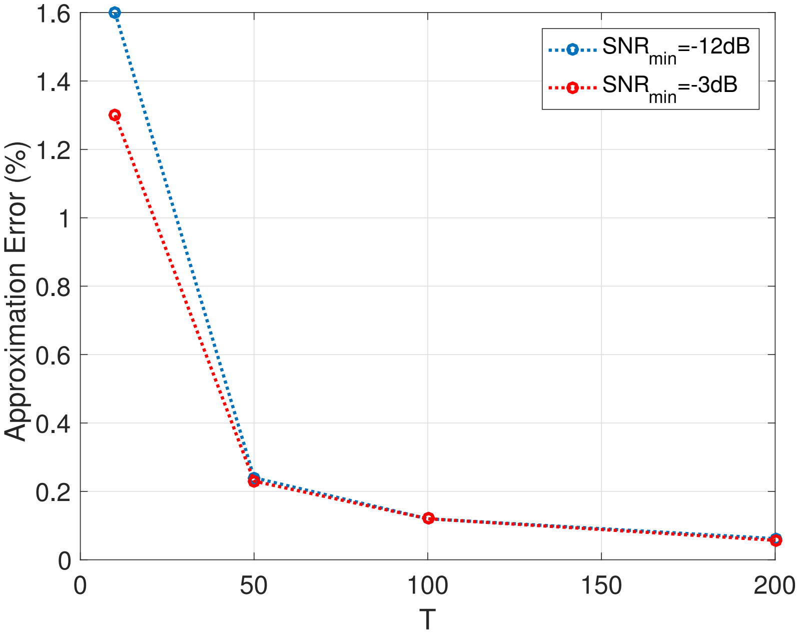


Figure 2.14: Normalized approximation error between the actual variance $\sigma_{a1}^2(\omega_m)$ and its upper bound, versus number of iterations.

2.7.7 SNR Loss due to RSFT as Compared with the FFT-based Method

The counterpart of the RSFT-based signal processing is the FFT-based Bartlett method followed by an NP detection procedure [39, 40] (see Algorithm 2 for details). To compare the two methods, in Section M we derive the relationship between the P_d and the P_{fa} for the Bartlett method using the same signal model as that of the RSFT.

As compared with the FFT-based Bartlett method, the reduction of complexity of RSFT is achieved at a cost of degradation of detection performance; such degradation can be compensated by increasing the SNR of the weakest sinusoid, i.e., SNR_{min}. The tradeoff between complexity and SNR_{min} involves the reduced data length, B. Typically, a smaller B yields lower computational complexity and requires a larger SNR_{min}; this is shown in Fig. 2.15 for different sparsity level, K. A small K requires a lower SNR_{min}. The case for the FFT-based Bartlett method to achieve the same detection performance as that of RSFT is also shown. When $B = 128, K = 5$, the computation savings of RSFT as compared with the Bartlett method is approximately $7dB$, meanwhile, the SNR_{min} required by the former is approximately $11dB$ greater than the latter.

Note that when $B = N$, the complexity of the RSFT is greater than that of the Bartlett method, also, the detection performance of former is worse than the latter. This is due to the two stages of windowing in RSFT degrade the SNR. Hence, in the case of $B = N$, one would adopt the FFT-based, rather than RSFT-based processing.

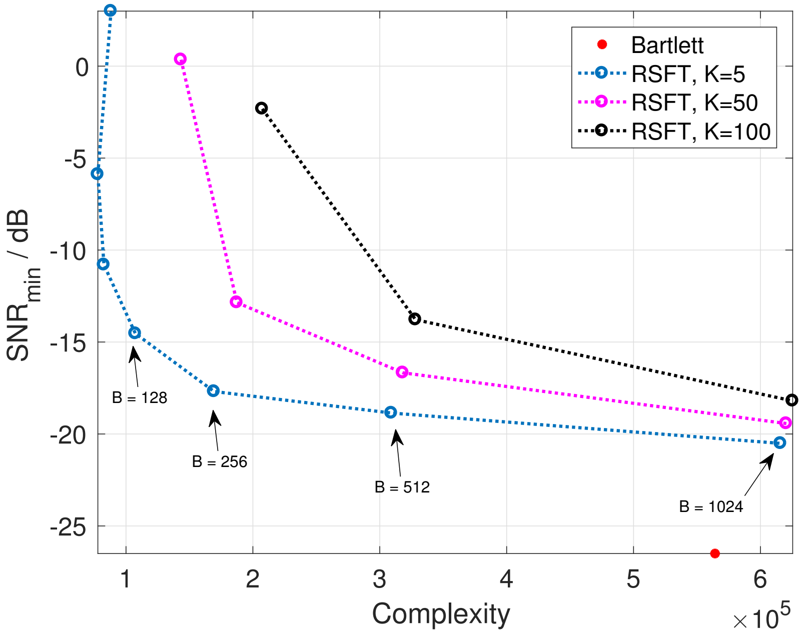


Figure 2.15: Worst case SNR and complexity tradeoff. $N = 1024, T = 50, D = 0dB, B = \{8, 16, 32, 64, 128, 256, 512, 1024\}, P_d = 0.9, P_{fa} = 10^{-6}, K = \{5, 10, 100\}, \omega_m = \Delta\omega_N/2$. The red dot shows the performance of the Bartlett method.

2.7.8 Characterize the Detection Performance for K Targets

We use the *recall* and *precision* [44] to quantify the detection performance; those metrics are widely used in the detection of multiple objects . The recall is defined as the ratio of the number of true positives to the number of frequencies existed in the signal, i.e., K. The precision is defined as the ratio of the number of true positives to the number of detected frequencies. The higher the recall and the precision, the better performance of the system.

In Fig. 2.16, we show the recall and precision versus SNR for different K. For a fixed K value, we compute the P_d for different SNR when $P_{fa} = 10^{-2}$ using (36) of [36]; here, we use a relatively large P_{fa} for the purpose of visualization. Based on this, the recall and precision can be predicted, i.e., the recall should be the same to P_d, while the precision can be computed as

$$P = \frac{P_d K}{P_d K + (N-K)P_{fa}}. \tag{2.32}$$

We compare the predicted recall and precision values with the experimental results. For the experiments, we generate signals containing K significant frequencies at various SNR values. The locations of the frequencies are random, and we guarantee that the spacing between two frequencies are large enough so that the RSFT can resolve. We use the computed optimal thresholds to detect the frequencies. After detection, due to windowing, each continuous-valued frequency may be represented by a cluster of frequencies. We extract the center location of each cluster and compare it with the ground truth location. A detection is regarded as a true positive if the center location of the cluster is within ±1 neighboring bins of the ground truth location. The center locations of clusters which do not fall into the neighborhood of any ground truth locations are regarded as false alarms. Fig. 2.16 shows that, as expected, the recall and precision become larger as SNR grows. The recall for $K = 5$ is always greater than that the cases when $K = 50$ for a same SNR due to that the P_d of the former is always greater than the latter. On the other hand, the precision of the former is worse than that of the latter, which can be explained by (2.32). While the recall calculated from experiments matches well with the predicted values, the precision calculated from experiments are slightly better than the predicted results, which suggests the number of actual false alarms are smaller than expected.

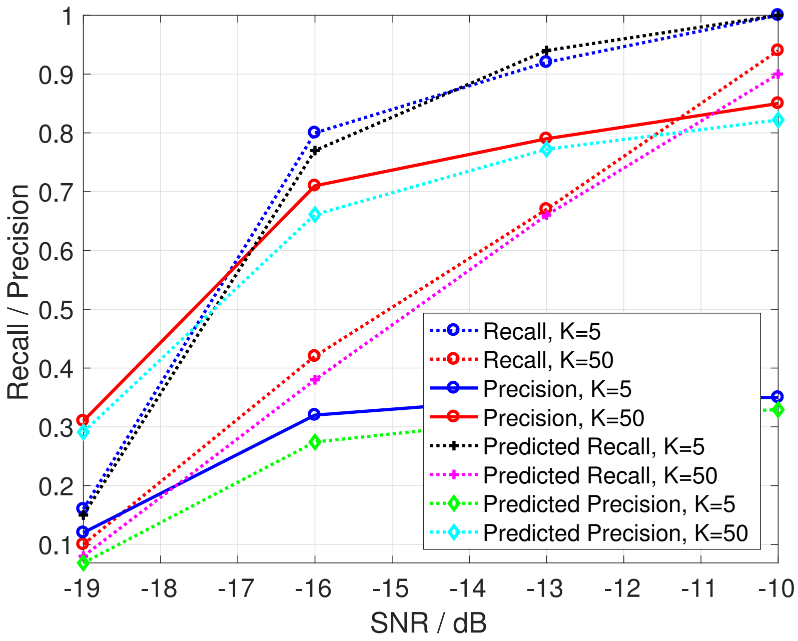


Figure 2.16: Recall and precision versus SNR for different K. $P_{fa} = 10^{-2}, D = 0dB, T = 50, N = 1024, B = 256, K = K_{max}$.

2.8 Summary

We have proposed a robust version of the simple and practical SFT algorithm, i.e., RSFT. RSFT employs a pre-permutation window and NP detection to address the off-grid frequency and frequency detection problems arising in the application of SFT in real-world situations. We have shown that the RSFT is robust in detecting frequencies when exact knowledge of signal sparsity is not available. The optimal design of parameters in RSFT have been analyzed, and the tradeoff between detection performance and computational complexity has been investigated. Such analysis has revealed that RSFT could provide an extra degree of freedom in design to trade off the system's ability to detect weak signals and complexity. Some interesting properties of the RSFT have also been revealed by our analysis. In particular, the performance of detection not only relies on the frequency under examination, but also depends on other co-existing significant frequencies. This is because the co-existing frequencies generate ambiguity locations in the reverse-mapping procedure, which raises the noise floor in the second stage detection.

Chapter 3

RSFT-based DBF and MIMO Radar Signal Processing

Conventional radar signal processing employs multidimensional Discrete Fourier Transform (DFT) to compute frequencies that are related to target parameters including range, range rate and Direction of Arrival (DOA). The DFT is usually implemented by the Fast Fourier Transform (FFT). The sample and computational complexity of the FFT is $O(N)$ and $O(N \log N)$, respectively, where N is the number of samples. For modern Digital Beamforming (DBF) and Multiple-Input Multiple-Output (MIMO) radar systems, N is large due to increasing of the dimension of processing and increased resolution in each dimension. Furthermore, some radar systems such as automotive radars have highly constrained hardware due to limited size and the low-cost requirement. Hence, the real-time signal processing of those radar systems remains a challenging problem. This motivates us to employ SFT-based radar signal processing to reduce the complexity of radar signal processing.

In Chapter 2, we proposed the Robust Sparse Fourier Rransform (RSFT), which enjoys low computational complexity as compared with the conventional FFT-based method, while addresses noisy signals containing off-grid frequencies. In this chapter, we formulate RSFT based radar signal processing frameworks for DBF radars using (linear) Frequency Modulation Continuous Waveform (FMCW), and MIMO radars using pulse-compression waveforms. To this end, the radar signal processing schemes are designed to support the application of RSFT, and the proper pre-processing is adopted to sparsify signals in cases when signals are not naturally sparse in a specific domain.

3.1 RSFT-based DBF Radar

3.1.1 Conventional Signal Processing of DBF Radar

A DBF radar can see targets everywhere at anytime. Unlike a traditional phased array radar which has to steer its beams, in DBF radars, a broad transmitting beam pattern is achieved by an omni-directional transmitter and multiple narrow beams are formed simultaneously after receiving the reflected signal. The beam pattern of an DBF radar is shown in Fig. 1.5 with a Uniform Linear Array (ULA) configuration.

Near range DBF radars are important both in military and civilian vehicular applications. In military applications, for instance, such kind of radars are well suited in active protection systems [45], allowing sensors on a vehicle to detect and locate the warheads of a closely fired rocket-propelled grenade within milliseconds. In addition to its wide angle coverage, high precision of measurement and all-weather operation render the DBF radar an ideal sensor for active protection system. In civilian applications, DBF radars play a more and more important role in Automotive Driver Assistance Systems (ADAS) and self-driving applications, where the radar offers high precision measurement in range, range rate, and DOA domains; this provides important information of surroundings to the perception systems of ADAS and self-driving systems [46, 11, 47].

In order to achieve high range resolution and cover near range, in the aforementioned applications, the DBF radar usually utilizes the FMCW waveform, as introduced in Section 1.2. Let us assume that the transmit waveform is grouped into bursts, with each burst contains M Repetition Intervals (RIs). Mathematically, the transmitted waveform can be expressed as

$$s(t, v) = A\cos(2\pi(f_c(t - vT_p) + \pi\rho(t - vT_p)^2), \tag{3.1}$$

where T_p is the RI, $v \in [M]$ denotes the v-th RI, A is amplitude of the signal, f_c is the carrier frequency and ρ is the chirp rate. Furthermore, without loss of generality, we assume that the initial phase of the signal is zero.

Upon reception, a de-chirp process is implemented by mixing the received signal with the transmitted signal, followed by a low-pass filter. The received signal is a delayed version of the transmitted one, hence by mixing the two signals, the range information of the targets is linearly encoded in the difference of the frequencies. Hence in the s-th, $s \in [T]$ burst, for the i-th, $i \in [N]$ receiving channel (corresponding to an N-element ULA of half-wavelength element-wise spacing), the de-chirped signal of the v-th RI is expressed as

$$r_{i,s}(t) = \sum_{k_0}^{K-1} a^{[k]}(s) \cos\left(2\pi((f_r^{[k]} + f_d^{[k]})(t - vT_p) + i\pi \sin\theta^{[k]}\right) + n(t), \tag{3.2}$$

which is a superposition of K sinusoids (corresponding to K targets) and additive noise $n(t)$. For the k-th sinusoid, $a^{[k]}(s)$ represents its amplitude, which can be modeled as a Gaussian random process. More specifically, the amplitude is assumed to be static within a burst, and independent between bursts. This assumption is consistent with the Swerling-I target model [9], which represents a slow fluctuation of the target RCS. $f_r^{[k]}, f_d^{[k]}$ are the frequency components respect to target's range and velocity respectively, i.e.,

$$\begin{aligned} f_r^{[k]} &= \frac{2\rho r_t^{[k]}}{c}, \\ f_d^{[k]} &= \frac{2v_t^{[k]}}{\lambda}, \end{aligned} \tag{3.3}$$

where $r_t^{[k]}, v_t^{[k]}, c$ are the k-th target's range, velocity and speed of wave propagation respectively. After analog-to-digital (AD) conversion, the received signal of each channel becomes R samples within each RI, with each sample representing a range bin.

The DOA of the k-th target, i.e., $\theta^{[k]}$ is defined as the angle between the line of sight (from the array center to the target) and the array normal. Assuming that the element wise spacing is $\lambda/2$, under the narrow-band signal assumption, $\theta^{[k]}$ will cause an increase of phase at the neighboring array element equal to $\pi \sin\theta^{[k]}$. We omit the constant phase term in each sinusoids of (3.2), since they are irrelevant to the performance of the algorithm.

After AD conversion of each receiving channel, we can use the processing scheme shown in Fig. 3.1 to detect the targets as well as estimate their range, velocity and DOA. More specifically, grid-based versions of $f_r^{[k]}, f_d^{[k]}, \pi \sin\theta^{[k]}$ can be calculated by applying a three dimensional (3-D) FFT on the windowed data cube [11], and then, after accumulation of T iterations (a different burst of signal is processed in each iteration), the NP detection procedure can be performed.

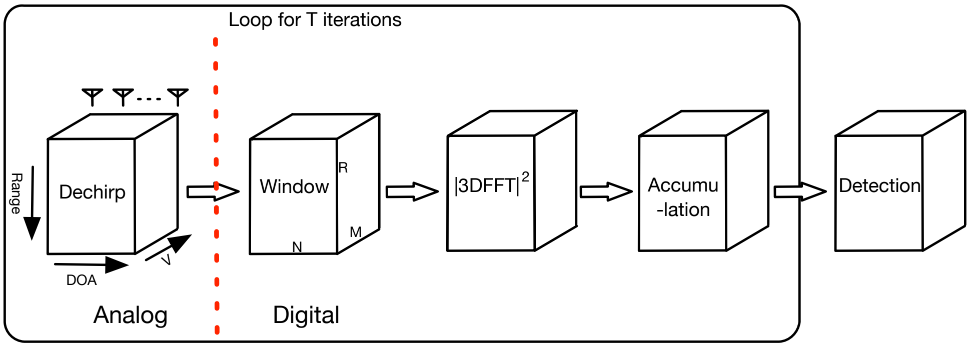


Figure 3.1: Conventional FFT-based processing scheme for FMCW DBF radars.

3.1.2 RSFT-based DBF Radar Signal Processing

The RSFT algorithm is suitable for reducing the computational complexity of the DBF radar using FMCW for the following reasons:

- The number of targets is usually much smaller than the number of resolution cells in 3-D range, Doppler and DOA space, which implies that the signal is sparse after proper translation.
- With a ULA and digitization of each received element, the signal is uniformly sampled both in spatial and temporal domain.
- The short range coverage implies that moderate high SNR is easy to achieve as compared with long range radars.

The RSFT-based DBF radar processing architecture is shown in Fig. 3.2. Compared to the conventional processing in Fig. 3.1, the 3-D FFT is replaced with a 3-D RSFT, in which the aliasing procedure reduces the data cube size from $R \times N \times M$ to $B \times C \times D$, with $B < R, C < N, D < M$. The 3-D FFT operated on the smaller data cube could reduce the computation time significantly.

Based on the radar architecture described by Fig. 1.2, we verify the feasibility of the RSFT-based DBF radar processing and compare it to simple and practical SFT based processing via simulations. The main parameters of the system are listed in Table 3.1. The

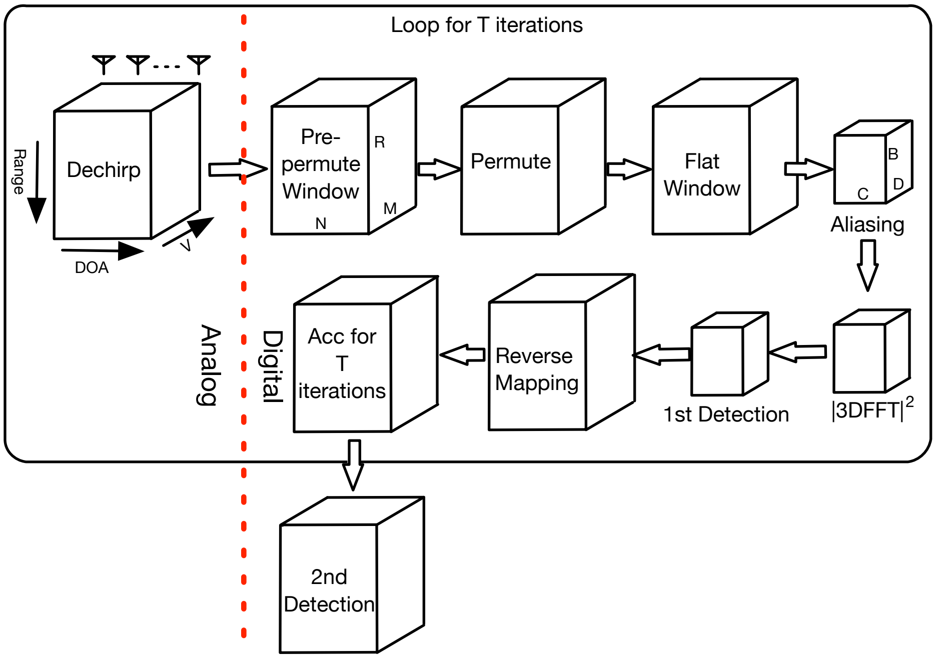


Figure 3.2: RSFT-based processing scheme for the DBF radar.

design of the system can guarantee non-ambiguous measurements of the target's range and velocity, assuming the maximum range and velocity are less than $1.5km$ and $300m/s$, respectively.

We generate signals from 4 targets according to (3.2). The range, velocity and DOA of targets can be arbitrarily chosen within the unambiguous space, which implies that the corresponding frequency components do not necessarily lie on the grid. The targets' parameters used in the simulation are listed in Table 3.2. For targets 3 and 4, we use the same range and velocity values but set their DOA to be 4° apart; this is close to the theoretical angular resolution after windowing for the Bartlett beamforming. To compare RSFT and simple and practical SFT for different scenarios, we adopt two sets of SNR for targets. For the first set, we use the same SNR, i.e., $-10dB$ for different targets, while for the second set, we assign different SNR values to different targets, which is closer to a realistic scenario.

Simple and practical SFT is a 1-D algorithm. In order to use it for reconstructing targets in the 3-D space, we extend it along the lines of the multidimensional RSFT, as described

Table 3.1: DBF radar parameters

Parameter	**Symbol**	**Value**
Number of range bins	R	2048
Number of receiving channels	N	64
Number of RI	M	32
Wave length	λ	$0.03m$
Wave propagation speed	c	$3 \times 10^8 m/s$
Bandwidth	B_w	$150MHz$
Repetition interval	T_p	$5 \times 10^{-5}s$
Maximum range	R_{max}	$1.5 \times 10^3 m$
Chirp rate	ρ	$3 \times 10^{12} Hz/s$
Sampling frequency (IQ)	f_s	$41MHz$
Reduced data length in range	B	128
Reduced data length in DOA	C	32
Reduced data length in velocity	D	16

Table 3.2: Target Parameters

Target	**Range** (m)	**Velocity** (m/s)	**DOA** ($\circ$)	**SNR (dB)**
1	1000	100	30	$-10/0$
2	500	50	0	$-10/-10$
3	350	240	-16	$-10/-20$
4	350	240	-20	$-10/-20$

in Section 2.6. For simple and practical SFT, due to existence of the large number of peaks from leakage, even if the exact number of targets is known, it is still not clear how one can determine the number of peaks to be counted. Hence, in the implementation of simple and practical SFT in the experiment, we gradually increase the number of peaks that are counted until all the targets are recovered. This is not realistic in real-world radar applications because we will never know that if all the targets within radar coverage are detected. On the other hand, the RSFT can do effective detection by knowing the SNR of the weakest frequency, dynamic range, noise variance and a sparsity bound. Typically, these parameters can be found in the design specifications or from the field test of the radar. For the case of the same SNR setting, all the targets are recovered after around 20 peaks are counted in simple and practical SFT. For the second SNR setting, simple and practical SFT needs to count nearly 200 peaks to recover the weakest targets (Targets 3 and 4). Figs. 3.3 and 3.4 show the target reconstruction results for the two settings, respectively. The former shows that both simple and practical SFT and RSFT can perfectly recover all the targets that have the same SNR. From targets 3 and 4 we can see that simple and practical SFT based method achieves a better resolution than its RSFT counterpart, which is expected since the former does not require a pre-permutation window. Such reduction in resolution is common and usually inevitable in many DFT-based applications that utilize the windowing technique to reduce frequency leakage; the reduction is usually minor and can be compensated by a proper system level design. For example, to compensate for the resolution reduction in the range domain, one can extend the bandwidth of the waveform to a desired level. In the second scenario, simple and practical SFT based method shows the side-lobes of the stronger targets, while the RSFT-based method only recovers the (extended) main-lobes of all the targets.

The simulation shows that the RSFT-based approach is better than its simple and practical SFT counterpart for a realistic scenario, within which the signal has a reasonable dynamic range. We should emphasize that in a real radar system, determining the number of peaks to be counted for simple and practical SFT based method lacks a theoretical foundation, while the thresholding approach in the RSFT is consistent with the conventional FFT-based processing, both of which are based on the Neyman-Pearson criterion.

3.2 RSFT-based MIMO Radar

As introduced in Section 1.4, MIMO radars employ multiple transmit and multiple receive antennas to simultaneously form multiple beams. Compared to DBF radars, collocated MIMO radars [14, 16] enjoys improved parameter identifiability due to he increased degree of freedom in transmission.

The MIMO radar's wide angle coverage shown in Fig.1.5 is achieved by multiple channels for transmitting and receiving. During the transmission, a set of mutually orthogonal waveforms are transmitted by each array element with an omni-directional beam pattern; after the signal is received from each digitized receiving channel, multiple narrow beams are formed in the Digital Signal Processor (DSP) using beamforming methods.

The implementation of MIMO radars usually involves high cost, which is mainly due to: 1) the large number of transmit and receive Radio Frequency (RF) channels; and 2) the high

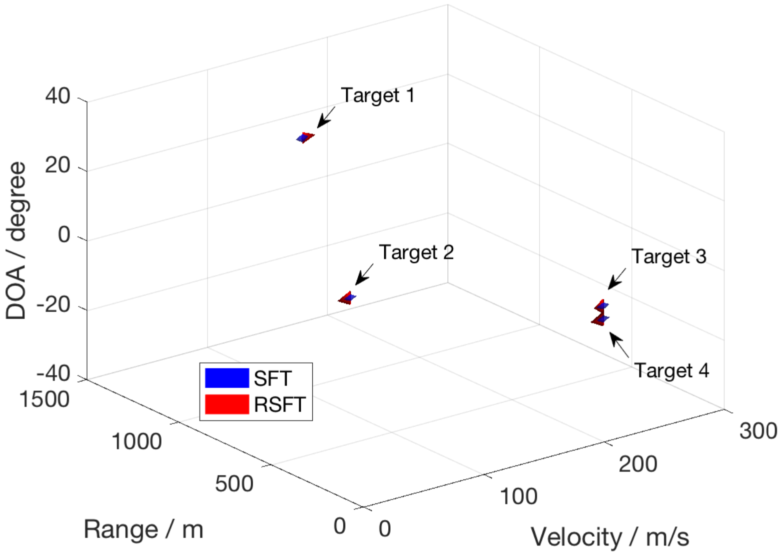


Figure 3.3: Target reconstruction via 3-D simple and practical SFT and RSFT; all targets have the same SNR. Both simple and practical SFT and RSFT based methods can reconstruct all the targets, while simple and practical SFT has better resolution.

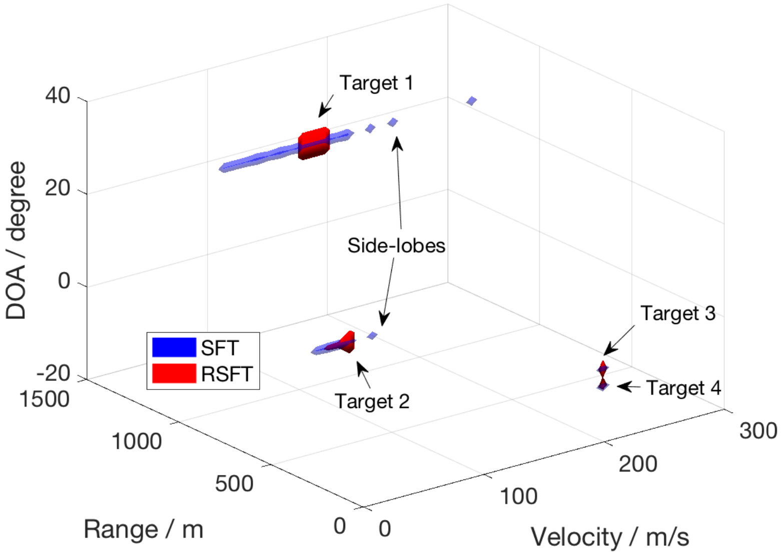


Figure 3.4: Target reconstruction via 3-D simple and practical SFT and RSFT with different SNR for the 4 Targets. Simple and practical SFT based processing recovers the side-lobes of the stronger targets, while the RSFT-based method only recovers the main-lobes of targets.

data throughput and complex processing, which results in costly DSPs. The cost of the RF channels can be reduced by sharing the transmit and receive antennas, i.e., each antenna element can be used both for transmitting and receiving in a *pulse mode* [14]. Here, we will assume such configuration and address the reduction of the cost of DSP.

Modern pulse radars usually employ the Pulse Compression (PC) technique to increase their sensitivity (ability to detect weak signals) and range resolution. To implement PC in MIMO radars, the waveforms must have good cross- and auto-correlation properties [20], which are properties needed for PC and *Waveform Decoding* (WD). WD is a process that separates the orthogonal waveforms from each transmitter, so that the so called *transmit beamforming* [14] can be applied subsequently to compensate for time delay caused during transmission. To this end, Code Division Multiple Access (CDMA) waveforms [20, 48] are usually adopted. Depending on the application, the baseband CDMA code sequences can be applied on the slow-time (time across pulses) or on the fast-time (time within a single pulse), yielding different processing schemes and computational complexities.

The high degrees of freedom in MIMO radars result in high dimensional computations. For the MIMO radar of Fig. 1.5, in order to detect targets and estimate their range and DOA, we need to do processing in a 3-D space, i.e., range, transmit DOA and receive DOA. Although parametric methods yield in general better resolution, conventional, Fourier transform-based methods are often preferable in practice due to their robustness to noise and their lower computational complexity [49]. In fact, matched filtering for range processing, and transmit and receive beamforming for DOA processing can be effectively implemented via the FFT (see e.g., [6]), which has a complexity of $O(RN^2 \log(RN^2))$ for each pulse repetition interval (PRI), where N, R are the number of array elements and the number of the range bins, respectively. When N, R are large, or, when the dimension of signal processing continuous to increase (for instance, when both azimuth and elevation DOA are considered), the computation of the FFT becomes costly. This motivates the application of the RSFT in MIMO radar signal processing. We formulate RSFT-based MIMO signal processing framework, and make the following contributions:

- We propose MIMO-RSFT radar, a reduced complexity MIMO radar that employs the RSFT to reduce the cost of the DSP.

- We explore slow-time and fast-time coded waveforms that support PC for MIMO radars. The implementation of the RSFT and the resulting computational savings are investigated for both cases.
- We provide the key steps of deriving the optimal detection thresholds for the MIMO-RSFT radar based on the signal model.

3.2.1 Signal Model

Let us consider the MIMO radar configuration of Fig. 1.5. During transmission, a set of orthogonal CDMA waveforms are transmitted by each antenna element. Let us denote the discrete baseband signal, which is transmitted by the u-th, $u \in [N]$ element in the s-th, $s \in [T]$ PRI by $\mathbf{s}_{u,s} \in \mathbb{C}^M$. Suppose that there are K targets within the radar coverage. For simplicity, we do not incorporate Doppler in our signal model, thus implicitly assuming that the targets are moving slowly and their Doppler can be neglected. Note that including the Doppler processing in the MIMO-RSFT radar is straightforward, and is briefly discussed in Section 3.2.7. The received signal of the i-th, $i \in [N]$ receiving channel (after quadrature demodulation and AD conversion) is $\mathbf{r}_{i,s} \in \mathbb{C}^R$, which is a superposition of the signals that are returned from K targets, i.e.,

$$\mathbf{r}_{i,s} = \sum_{k=0}^{K-1} \left(b_{k,s} e^{ji\pi \sin\theta_k} \sum_{u=0}^{N-1} \mathbf{a}_{u,s}(t_k) e^{ju\pi \sin\theta_k} \right) + \mathbf{n}_s, \tag{3.4}$$

where $t_k \in [R-M], \theta_k \in [-\pi/2, \pi/2]$ denote sample delay and DOA (the angle between the line-of-sight of target and the array normal) of the k_{th} target, which are unknown deterministic quantities and are assumed to be stationary within T PRIs. The phase terms $e^{ji\pi \sin\theta_k}$ and $e^{ju\pi \sin\theta_k}$ are respectively caused by the channel-wise time delay during reception and transmission, by assuming that the signal is narrow-band and the array elements are spaced apart by half wavelength. We use $\mathbf{a}_{u,s}(t_k) \in \mathbb{C}^R, R > M$ to represent the fast-time data samples within the s_{th} PRI, which contains a delayed by t_k version of $\mathbf{s}_{u,s}$, i.e., $[\mathbf{a}_{u,s}]_{v+t_k} = [\mathbf{s}_{u,s}]_v, v \in [M]$, and the other entries of $\mathbf{a}_{u,s}$ equals to zero. The $b_{k,s}$ is the complex amplitude of the k_{th} target, which is circularly symmetric Gaussian and distributed as $b_{k,s} \sim \mathcal{CN}(0, \sigma_{bk}^2)$; the noise $\mathbf{n}_s$ is temporal and spatial white, distributed as $\mathbf{n}_s \sim \mathcal{CN}(\mathbf{0}, \sigma_n^2 \mathbf{I})$, where $\mathbf{0}$ is R-dimensional zero vector, and $\mathbf{I} \in \mathbb{R}^{R \times R}$ is the identity matrix.

Let $\mathbf{R}_s = [\mathbf{r}_{0,s},\ \mathbf{r}_{1,s},\ \cdots\ \mathbf{r}_{N-1,s}]$ represent the data collected by all antennas. The data collected over T pulses, i.e., $\mathbf{R}_s$, $s = 0, 1, \cdots, T-1$, will be used to detect the targets and estimate their range and DOA. The conventional processing schemes for slow-time and fast-time coded waveforms are presented in Fig. 3.5 (a) and Fig. 3.6 (a), where WD and PC are implemented sequentially and simultaneously, respectively. Conceptually, WD separates each $\mathbf{a}_{u,s}(t_k)e^{ju\pi \sin\theta_k}$ component in $\mathbf{r}_{i,s}$, while PC convolves $\mathbf{a}_{u,s}(t_k)$ with $\mathbf{s}_{u,s}$ to achieve high range resolution. After that, the transmit and receive beamforming are implemented along the transmit and receive channel, respectively. Subsequently, after a non-coherent accumulation, a detection procedure on each resolution cell is applied.

The FFT and inverse FFT (IFFT) can be employed in various stages of the processing, however, the complexity is still high due to high dimensional data. In what follows, we explain how one can use the RSFT in both slow-time and fast-time coding schemes to save computation.

3.2.2 Slow-time and Fast-time Coded Waveforms Processing

Let us consider the transmit waveforms to be unimodal [20], i.e., $|[\mathbf{s}_{u,s}]_i| = 1, i \in [M]$. First, we discuss the orthogonality and PC requirements for such waveforms. Based on that, we compare the processing schemes for slow-time and fast-time coded waveforms and derive their computational complexities.

The correlation between $\mathbf{s}_{u,s}$ and $\mathbf{s}_{v,s}$ at lag $n, n \in [M]$ equals to $c_{u,v,s}(n) = c^*_{v,u,s}(-n) = \sum_{i=n+1}^{M} [\mathbf{s}_{u,s}]_i [\mathbf{s}_{v,s}]^*_{i-n}$. The slow-time orthogonality requires that the pulses emitted from different transmitters be uncorrelated within L consecutive PRIs, i.e.,

$$\sum_{s=0}^{L-1} c_{u,v,s}(0) = 0,\ u, v \in [N], u \neq v, \tag{3.5}$$

while the fast-time orthogonality requires that $L = 1$ in (3.5). Note that (3.5) guarantees that: 1) the transmit beam-pattern is omni-directional within L PRIs; and 2) the WD can be applied upon reception.

PC requires the auto-correlation of each pulse at different non-zero lags be below certain level, i.e., $|c_{u,u,s}(n)| < \epsilon, n \in [M] \setminus 0$, where $0 \leq \epsilon << |c_{u,u,s}(0)| = M$.

Slow-time Coding

The slow-time coding scheme [48] can employ any pulse waveform $\mathbf{s}$ that supports PC (e.g., the Barker code waveform) as its base waveform. To achieve (3.5), the antennas transmit T/L bursts, assuming that T is divisible by L; a burst is composed of L consecutive pulses, i.e., $\mathbf{s}_{u,s}, u \in [N], s \in [L]$, whose initial phases are coded by N mutually orthogonal unimodal sequences (e.g., the Hadamard sequences [48]), which is denoted as $\mathbf{h}_u \in \mathbb{C}^L, u \in [N]$. Hence $\mathbf{s}_{u,s} = [\mathbf{h}_u]_s \mathbf{s}$. Upon reception, WD is applied for each receiving channel on the burst basis by correlating $\mathbf{r}_{i,s}, s \in [L]$ with $\mathbf{h}_u$, yielding $\mathbf{w}_{i,u} \in \mathbb{C}^R, u \in [N]$, i.e.,

$$\mathbf{w}_{i,u} = [\mathbf{r}_{i,0} \ \ \mathbf{r}_{i,1} \cdots \mathbf{r}_{i,L-1}]\mathbf{h}_u = L \sum_{k \in [K]} b_k e^{j(i+u)\pi \sin\theta_k} \mathbf{a}(t_k) + \tilde{\mathbf{n}}, \tag{3.6}$$

where $\mathbf{a}(t_k)$ is t_k-delayed version of $\mathbf{s}$; $\tilde{\mathbf{n}}$ is the noise part. The WD process for each burst has a complexity of $O(N^2LR)$. Subsequently, PC is applied by matched filtering $\mathbf{w}_{i,u}$ with $\mathbf{s}$. With the matched filtering being implemented in the frequency domain for efficiency, PC for each burst has a complexity of $O(N^2R\log R)$. The subsequent transmit and receive beamforming for each burst, when implemented with FFT, has a complexity of $O(RN^2 \log N^2)$. Therefore, the complexity of processing each burst is $O(UL + U\log U)$, where $U = RN^2$. Clearly, for T PRIs (T/L bursts), the processing scheme in Fig. 3.5 (a) gives a complexity of $O(T(U + \frac{U}{L}\log U) + U)$.

Fast-time Coding

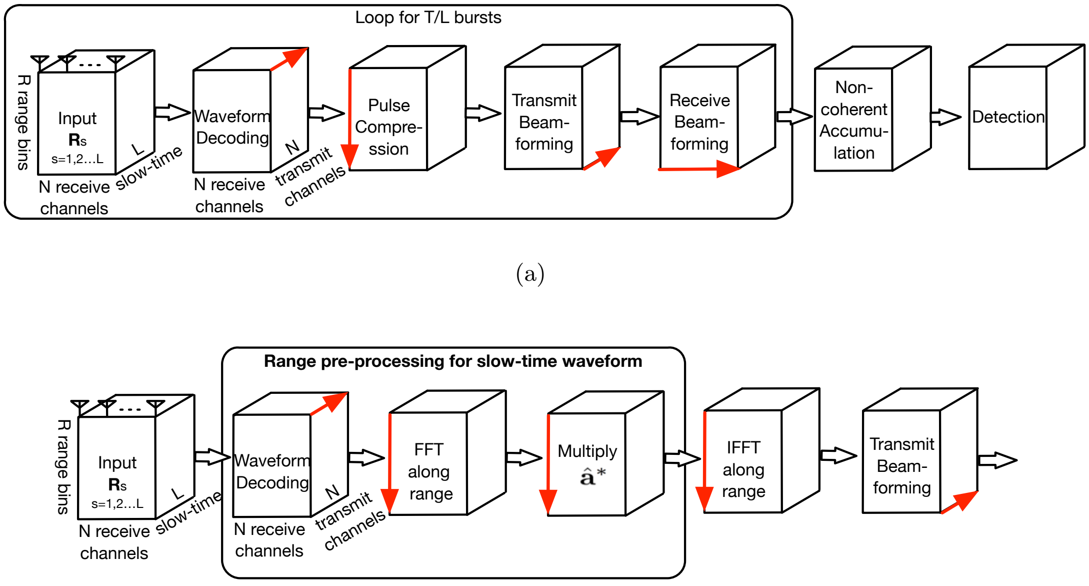
As opposed to slow-time coding that applies orthogonal coding on the inter-pulse basis, the fast-time coding implements the coding on the intra-pulse basis. However, since the ideal cross- and auto-correlation properties cannot be achieved at the same time [20], the orthogonality and the non-zero lag cross-correlation for the fast-time coded waveforms are approximate, i.e., $|c_{u,v,s}(n)| < \gamma << M$ for $n \in [M], u \neq v$. Upon reception, WD and PC can be simultaneously achieved by matched filtering $\mathbf{r}_{i,s}$ with $\mathbf{s}_{u,s}, u \in [N]$, which yields

$$\mathbf{y}_{i,u} = \mathbf{s}_{u,s} * \mathbf{r}_{i,s} \approx \sum_{k=0}^{K-1} b_k e^{j(i+u)\pi \sin\theta_k} (\mathbf{s}_{u,s} * \mathbf{a}_{u,s}(t_k)) + \bar{\mathbf{n}}, \tag{3.7}$$

where $\bar{\mathbf{n}}$ is the noise component. When the matched filtering is implemented in the frequency domain (see Fig. 3.6 (b)), the complexity is $O(U \log R)$. The subsequent transmit and receive beamforming has a complexity of $O(U \log N^2)$, therefore, the complexity of fast-time coding processing for T PRIs (Fig. 3.6 (a)) is $O(T(U \log U) + U)$.

3.2.3 RSFT-based Collocated MIMO Radar Signal Processing

The conventional processing for slow-time and fast-time coding schemes (see Fig. 3.5 (a) and Fig. 3.6 (a)) share a similar structure, except that their WD and PC processing are different. By packing some of the operations in WD and PC into a so called *range pre-processing* procedure, we are able to present a uniform RSFT-based processing structure for both coding schemes as shown in Fig. 3.7, which simplifies our discussion on MIMO-RSFT radar for both coding schemes.



(a)

(b)

Figure 3.5: Conventional MIMO radar signal processing for slow-time coded waveforms. The red arrows indicate the dimension of related processing. (a) Overview of MIMO radar processing for slow-time coded waveforms. WD, PC, transmit and receive beamforming are processed on the burst basis. For better detection performance, a non-coherent accumulation over T/L bursts is applied. (b) Range pre-processing for slow-time coded waveforms.

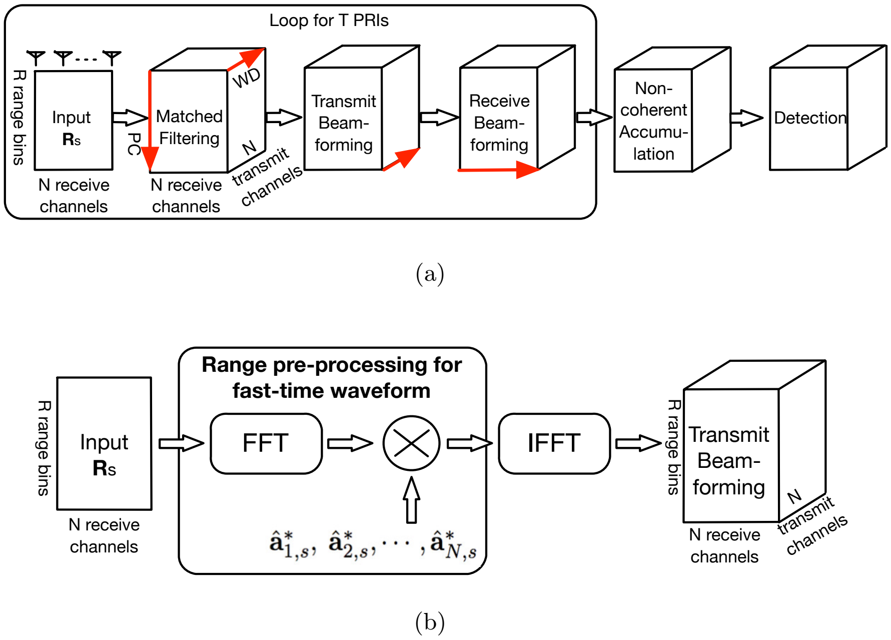


Figure 3.6: Conventional MIMO radar signal processing for fast-time coded waveforms. (a) Overview of MIMO radar processing for fast-time coded waveforms. The matched filtering, transmit and receive beamforming are applied in each pulse. A non-coherent accumulation over T pulses is applied. (b) Matched filtering and range pre-processing for fast-time coded waveforms.

3.2.4 Range Pre-processing

For slow-time coded waveforms, the range pre-processing includes WD and the front-end of PC, which contains the FFTs on the decoded waveform $\mathbf{w}_{i,u}$ and the multiplications between $\hat{\mathbf{w}}_{i,u}$ and $\hat{\mathbf{a}}^*$, as shown in Fig. 3.5 (b). For fast-time coded waveforms, the range pre-preprocessing contains the front-end of the frequency domain implementation of matched filtering, i.e., the FFTs on $\mathbf{r}_{i,s}$ and the multiplications between $\hat{\mathbf{r}}_{i,s}$ and $\hat{\mathbf{a}}^*_{u,s}$, as shown in Fig. 3.6 (b).

3.2.5 A Uniform Processing Structure for MIMO-RSFT Radar

Since the baseband signal is not sparse in the time domain nor in the frequency domain, the RSFT cannot be directly applied on the range domain. However, after PC, the signal becomes sparse in the time domain, which suggests that the backend of PC, i.e., the IFFT can be replaced by the RSFT. Moreover, since the signal is sparse in the DOA domain, both transmit and receive beamforming can be implemented with the RSFT. Hence, after the range pre-processing, we apply a 3-D RSFT on the signal to implement the detection and estimation. The processing scheme for the RSFT-based MIMO (MIMO-RSFT) radar is shown in Fig. 3.7. Note that, since we summarize the difference of the slow-time and fast-time coding processing in the range pre-processing, the processing structure of Fig. 3.7 can be applied to both coding schemes.

3.2.6 Complexity Analysis for MIMO-RSFT Radar

The complexity of the RSFT is shown in (2.29). Thus, based on the processing scheme of Fig. 3.7, the complexity of RSFT-based processing for slow-time and fast-time coding schemes are $O\left(\frac{T}{L}(UL + U\log R + \phi) + U\right)$ and $O\left(T(NR\log R + \phi) + U\right)$, respectively.

Remark 5. *The computational savings of the slow-time coding scheme are not significant, mainly due to its WD, whose most computational intensive part cannot take advantage of the RSFT. On the other hand, since the fast-time coding scheme can use the RSFT in all stages, the corresponding reduction of complexity can be significant.*

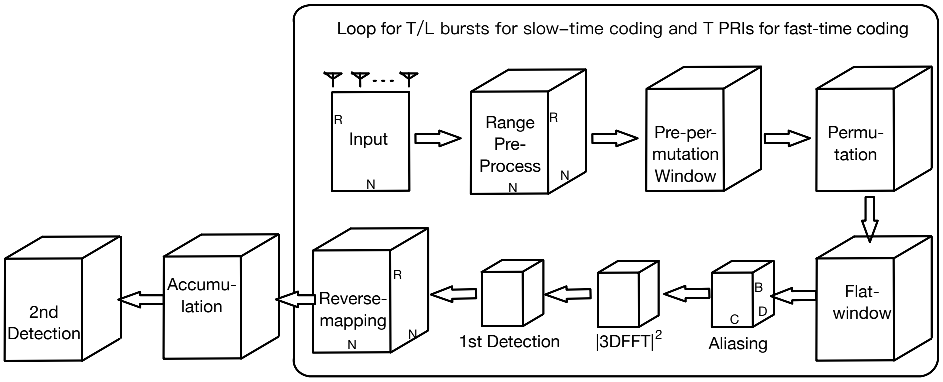


Figure 3.7: MIMO-RSFT radar signal processing for both slow-time and fast-time coding schemes. In each iteration, the input for slow-time and fast-time coding is a burst of pulses and a single pulse, respectively.

3.2.7 Doppler Processing for the Fast-time Coded Waveform

The Doppler frequency adds an additional dimensionality in the processing. For fast-time coded waveform processing, The received signal $\mathbf{R}_s, s = 0, 1, \cdots, T-1$ are partitioned into T/P Coherent Processing Intervals (CPIs), with each CPI contains P consecutive received data matrices. The Doppler processing, for example, the Moving Target Detector (MTD) are applied on the same range, transmit DOA, receive DOA resolution cell within a CPI. In conventional processing, the MTD can be effectively implemented via FFT, hence including of MTD in the fast-time coded MIMO-RSFT radar is straightforward, i.e., the range pre-processing for each CPI generates a 4-D tensor of size $R \times N \times N \times P$, then the following 4-D RSFT procedures are carried out on such tensor.

3.2.8 Simulation

Targets Reconstruction

We verify the feasibility of MIMO-RSFT radar and compare to the FFT-based and simple and practical SFT based processing via simulation. The main parameters of the system are listed in Table 3.3. We generate a signal from 4 targets according to (3.4). The targets' parameters used in the simulation are listed in Table 3.4. Note that we set targets 3 and

4 being close to each other in range and DOA to test the resolution of the MIMO-RSFT radar.

In the 3-D RSFT and SFT processing, we choose $B = 128, C = 32, D = 32$, respectively. The results are shown in Fig. 3.8. Comparing to the FFT-based processing, the 4 targets are reconstructed exactly via the RSFT-based processing, while the SFT-based method results in many false alarms due to the ineffective detection and the high side-lobes from the strongest target (Target 1) in the range and transmit DOA plane. Furthermore, compared to the conventional FFT-based processing, the resolution of the MIMO-RSFT does not degrade.

Table 3.3: DBF radar parameters

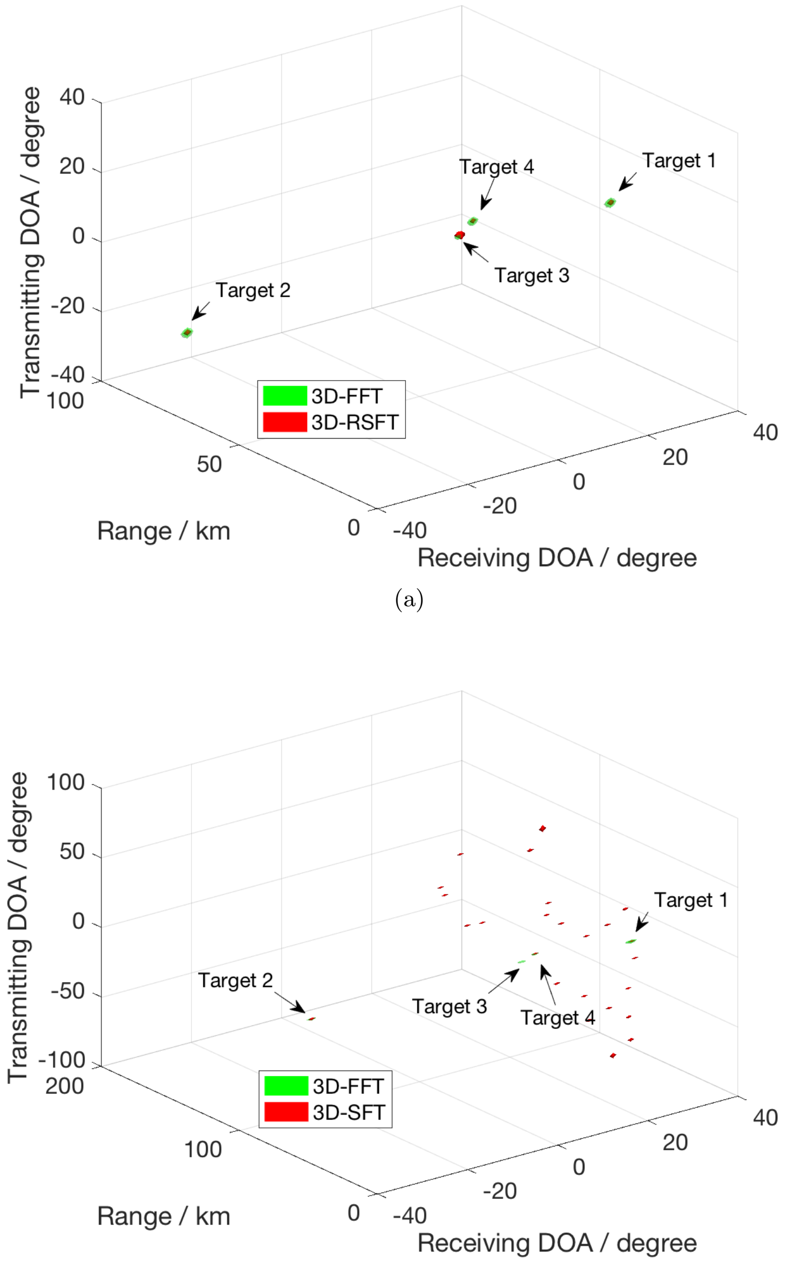
Parameter	**Symbol**	**Value**
Number of range bins	R	1024
Number of antenna elements	N	128
Length of CDMA Code	M	256
Number of PRI	T	32
Wave length	λ	$0.03m$
Wave propagation speed	c	$3 \times 10^8 m/s$
PRI	T_p	$25.6ms$
Maximum range	R_{max}	$156.6km$
Sampling frequency (IQ)	f_s	$1MHz$

Table 3.4: Target Parameters

Target	**Range** (km)	**DOA** ($\circ$)	**SNR** (dB)
1	14.8	20	0
2	90	-28	-5
3	44.8	5	-10
4	45	8	-10

Computational Savings

We compare the computational savings obtained by the MIMO-RSFT with that of the FFT-based processing both for slow-time and fast-time coded waveforms. The complexity of both slow-time and fast-time coding schemes is affected by the number of samples in the reduced space, i.e., $V = BCD$ and the signal sparsity K. A smaller V and K will lead to more computational savings of the MIMO-RSFT radar. For the slow-time coding scheme, Fig. 3.9 (a) shows that the RSFT-based processing does not save much even when V is

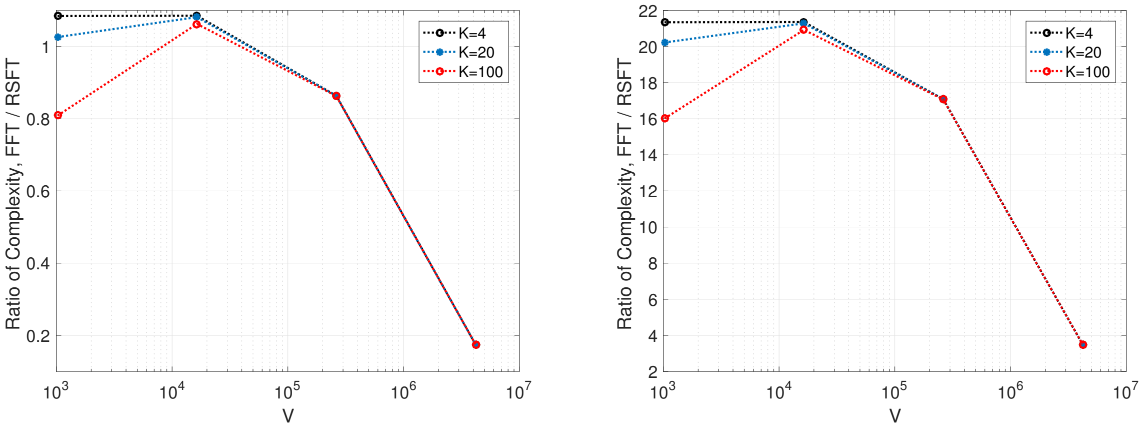


(a)

(b)

Figure 3.8: Target reconstruction via 3-D FFT, RSFT and SFT. Comparing to the 3D-FFT, 3D-RSFT can recover the targets exactly, while the recovery via the 3D-SFT results into many false alarms due to the leakage from Target 1. (a) FFT vs RSFT. (b) FFT vs SFT.

small, and when V becomes larger, its complexity is even higher than that of the FFT-based processing. On the other hand, for the fast-time coding scheme, Fig. 3.9 (b) shows that the computational savings of the RSFT-based method is significant. We shall point out that since the RSFT trades off complexity and sensitivity, more savings in computation will result in a larger degradation of sensitivity, and the MIMO-RSFT radar offers an extra degree of freedom for designing a MIMO radar by trading off complexity with sensitivity.



(a)

(b)

Figure 3.9: Complexity Ration, simple and practical FFT over RSFT. $U = 2^{24}, V = \{2^{10}, 2^{14}, 2^{18}, 2^{22}\}, \eta_m = 4, \eta_p = 1$. Other parameters are the same as in Table 3.1. (a) slow-time coding. $L = N, T = 4L$. (b) fast-time coding.

3.3 Summary

We have proposed RSFT-based signal processing frameworks for DBF radar that employs FMCW, and MIMO radars that use pulse-compression waveform. For the RSFT-based MIMO radar, the slow-time and fast-time coded waveforms are studied and a unified processing scheme based on RSFT is proposed for both waveforms. The computational savings by employing RSFT has been analyzed and the performance of the proposed processing schemes are demonstrated via simulations.

Chapter 4

Multidimensional Random Slice Based Sparse Fourier Transform

In this chapter, we propose MultidimensionAl Random Slice based Sparse Fourier Transform (MARS-SFT), a sparse Fourier transform for multidimensional, frequency-domain sparse signals, inspired by the idea of the Fourier projection-slice theorem. MARS-SFT identifies frequencies by operating on one-dimensional slices of the discrete-time domain data, taken along specially designed lines; those lines are parametrized by slopes that are randomly generated from a set at runtime. The Discrete Fourier Transform (DFT) of the data slices represent DFT projections onto the lines along which the slices were taken. On designing the lengths and slopes so that they allow for orthogonal and uniform frequency projections, the multidimensional frequencies can be recovered from their projections with low sample and computational complexity. We show theoretically that the large number of degrees of freedom of frequency projections allows for the recovery of less sparse signals. Although the theoretical results are obtained for uniformly distributed frequencies, empirical evidences suggest that MARS-SFT is also effective in recovering clustered frequencies.

4.1 Introduction

SFT is a family of low-complexity Fourier transform based algorithms that achieves significant reduction in sample and computational complexity in the implementation of DFT. The reduction of sample and computational complexity of SFT is achieved by reducing the input data samples. This is implemented via a well designed, randomized subsampling procedure, which leverages the resulting frequency domain aliasing. The significant frequencies contained in the original signal are then localized and the corresponding DFT coefficients are estimated with low-complexity operations. Such *subsampling-localization-estimation* procedure is carried out in an *iterative* manner in several SFT algorithms [21, 22, 3, 32], while in other SFT algorithms [4, 2, 1, 36], localization and estimation are implemented in *one-shot* after gathering sufficient copies of subsampled signals corresponding to different subsampling parameters, e.g., subsample rate, offset and number of samples. Generally, iterative based SFT algorithms exhibit lower complexity as compared to one-shot based SFT algorithms, since in the former, in each iteration, the contribution of the recovered frequencies are removed from the signal, which yields a sparser signal (an easier problem) in the next iteration.

Multidimensional signal processing requires multidimensional SFT algorithms. Most of the existing SFT algorithms, however, are designed for one-dimensional (1-D) signals and their extension to multidimensional signals is typically not straightforward. This is because the SFT algorithms are not separable in each dimension due to the fact that detection of significant frequencies in the subsampled signal must be considered jointly for all the dimensions [36]. Multidimensional SFT algorithms are investigated in [3, 1, 2]. The sample-optimal SFT (SO-SFT) of [3] follows the subsampling-localization-estimation iteration, while the SFT algorithms of [1, 2] are one-shot approaches. SO-SFT achieves the sample and computational complexity lower bounds of all known SFT algorithms by reducing a 2-dimensional (2-D) DFT into 1-D DFTs along a few columns and rows of a data matrix; in the frequency domain, this results into projections of 2-D frequencies onto the corresponding columns and rows of the matrix. Under the assumption that the frequencies are sparse and uniformly distributed, the 2-D frequencies are not likely to be projected to the same bin (we will refer this as collision), and thus the 2-D frequencies can be effectively recovered from their 1-D projections. The DFT along columns and rows provides two degrees of freedom; a frequency collision has low probability to occur both in the column and row direction. However, when frequencies are less sparse, or when they are clustered, there is a high probability that a set of frequencies will collide both in row and columns directions; this is referred to as ‘deadlock’ situation [3] and results in unrecoverable frequencies (see Fig. 4.1).

To reduce the probability of a deadlock, the SFT of [1, 2] introduces more degrees of freedom in projections by applying 1-D DFT to data samples extracted along some lines of predefined and deterministic slopes as well as along the axes of the data cube. However, the limited choice of line slopes in [1, 2] is still an obstacle in addressing less sparse signals. Moreover, the sample and computational complexity of [1, 2] are higher than that of SO-SFT, as the former applies the one-shot approach for frequency recovery, while the latter recovers the frequencies iteratively. In addition to the iterative approach, the low-cost frequency localization technique adopted in SO-SFT further contributes to the low-complexity of the algorithm. Specifically, SO-SFT applies the phase encoding (OFDM-trick) [5, 22], which effectively encodes the significant frequencies into the phase difference of a pair of DFTs applied on two datasets, obtained by subsampling the data with the same subsample rate but different offsets. In the exactly sparse case, the encoded frequencies can be decoded trivially with a low-complexity ($O(1)$) operation (see Section 1.5.2 for details).

In this chapter we propose MARS-SFT, which enjoys low complexity while avoiding the limitations of the aforementioned methods, i.e., it can handle less sparse data in the frequency domain and accommodate frequencies that are non-uniformly distributed. MARS-SFT uses the low-complexity frequency localization framework of SO-SFT and extends the multiple slopes idea of [1, 2] by using lines parameterized by slopes that are randomly generated from a set of sufficiently large support at runtime. This is not trivial since the line length and slope set should be carefully designed to enable an orthogonal and uniform frequency projection.

MARS-SFT can be viewed as a low-complexity, Fourier projection-slice approach for signals that are sparse in the frequency domain. In MARS-SFT, the DFT of an 1-D slice of the data is the projection of the D-dimensional (D-D) DFT of the data on that same line along which the time-domain slice was taken. The classical Fourier projection-slice based method either reconstructs the frequency domain signal via interpolation of frequency-domain slices or reconstructs the time-domain samples by solving a system of linear equations relating the DFT along projections and the time-domain samples. Different from the Fourier projection-slice based methods, the proposed MARS-SFT aims to reconstruct the signal directly based on frequency domain projections with low-complexity operations; this is achieved by leveraging the sparsity of the signal in the frequency domain.

Another body of related works, referred to as SFT based on rank-1 lattice sampling [26, 27, 50] also consider the problem of fast reconstruction of the underlying multidimensional signal based on samples along rank-1 lattices, i.e., lines. In [26, 50], the coefficients of the multidimensional DFT of the data can be efficiently calculated by applying DFT on samples along suitable lines, provided that the frequencies are known. In particular, in [26], the exact evaluation of the DFT coefficients can be accomplished by calculating the DFT along a single line; such a line is called the reconstructing rank-1 lattices and can be found for any given sparse frequency set [26]. However, finding a reconstructing rank-1 lattice is non-trivial when the frequency set is unknown. That unknown frequency case is addressed in [27], at the expense of high complexity due to the complication of finding a reconstructing rank-1 lattice.

Compared with the algorithms of [26, 50], the proposed MARS-SFT does not assume that the underlying frequency set is known. The frequency set as well as the corresponding DFT coefficients are estimated via DFT along lines progressively. Compared with the SFT of [27], MARS-SFT is based on multiple lines of randomized parameters and does not pursue to reconstruct the signal using a single line, which avoids the complication of locating a reconstructing rank-1 lattice and thus achieves a low complexity. In addition, the rank-1 lattice-based SFT algorithms assume that samples of the signal can be obtained at any arbitrary location, which is rather difficult to achieve in hardware[51]. In contrast, the MARS-SFT assumes that the samples are extracted from a predefined uniform sampling grid. Hence, MARS-SFT is less restrictive in sampling and more applicable to existing systems, which employ uniform sampling in each dimension.

The contribution of the proposed MARS-SFT algorithm is summarized as follows.

MARS-SFT is a multidimensional, low-complexity SFT algorithm that is based on the Fourier projection-slice theorem. Compared to the SFT algorithms of [3, 2, 1] that project multidimensional DFT of data onto deterministic lines, the frequency-domain projections in MARS-SFT are randomized. We show theoretically that the large number of degrees of freedom of frequency projections allows for the recovery of less sparse signals. Although the theoretical results are obtained for uniformly distributed frequencies, empirical evidences suggest that MARS-SFT is also efficient for recovery of clustered frequencies. Also, while the SFT of [3, 2, 1] requires the data to be equal-sized in each dimension, MARS-SFT applies to arbitrary-sized data, which is less restrictive.

4.2 Signal Model

For simplicity, we will present the ideas for 2-D signals. The generalization to higher dimensions, i.e., D-D cases with $D > 2$, is straightforward (see Section 4.5.1).

Let us consider the following 2-D signal model, which is a superposition of K 2-D complex sinusoids, i.e.,

$$x(\mathbf{n}) \triangleq \sum_{(a,\boldsymbol{\omega})\in\mathbb{S}} a e^{j\mathbf{n}^T\boldsymbol{\omega}},\ \mathbf{n} \triangleq [n_0, n_1]^T \in \mathcal{X} \triangleq [N_0] \times [N_1], \tag{4.1}$$

where N_0, N_1 denote the sample length of the two dimensions, respectively. $(a, \boldsymbol{\omega})$ represents a 2-D sinusoid whose amplitude is a with $a \in \mathbb{C}, a \neq 0$ and frequency is $\boldsymbol{\omega} \triangleq [\omega_0, \omega_1]^T \triangleq [2\pi m_0/N_0, 2\pi m_1/N_1]^T$ with $[m_0, m_1]^T \in \mathcal{X}$. The set $\mathbb{S}$ with $|\mathbb{S}| = K$ includes all the 2-D sinusoids. We assume that the signal is sparse in the frequency domain, i.e., $K << N \triangleq N_0N_1$, and that the frequencies are uniformly distributed.

4.3 Overview of MARS-SFT

The proposed MARS-SFT is a generalization of SO-SFT discussed in Chapter 1. SO-SFT reduces a 2-D DFT into 1-D DFTs of the columns and rows of the input data matrix. The columns and the rows can be viewed as 1-D slices taken along discrete lines with slopes ∞ and 0, respectively. In this section, by proposing MARS-SFT, we reduce the 2-D DFT into 1-D DFTs of the data slices taken along discrete lines with random slopes.

The proposed MARS-SFT is an iterative algorithm; each iteration returns a subset of recovered 2-D sinusoids. After T iterations, the MARS-SFT returns a set, $\hat{\mathbb{S}}$, which is an estimate of $\mathbb{S}$ of (4.1). As in SO-SFT, the sinusoids recovered in previous iterations are passed to the next iteration, and their contributions are removed from the signal in order to create a sparser signal.

Within each iteration, the signal of (4.1) is sampled along a L-length line, $\mathcal{E}(\boldsymbol{\alpha}, \boldsymbol{\tau}, l)$, $l = 0, 1, \cdots, L-1$, with $\boldsymbol{\alpha} \triangleq [\alpha_0, \alpha_1]^T$, $\boldsymbol{\tau} \triangleq [\tau_0, \tau_1]^T$, which satisfy the following equations

$$[\alpha_0 l + \tau_0]_{N_0} = n_0,\ [\alpha_1 l + \tau_1]_{N_1} = n_1,\ l = 0, 1, \cdots, L-1. \tag{4.2}$$

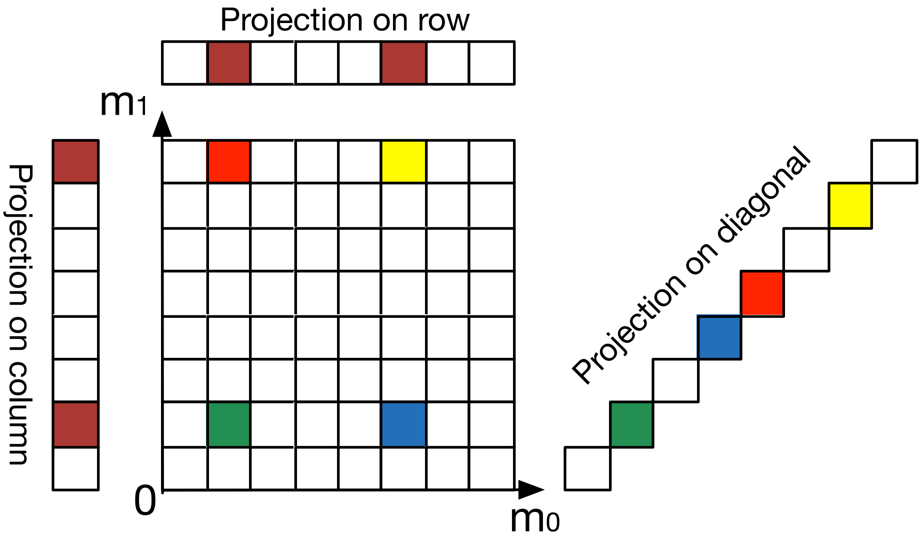


Figure 4.1: Demonstration of projection of 2-D frequencies onto 1-D. The colored blocks mark significant frequencies. The projection onto the column or the row causes collisions, while the projection onto the diagonal creates 1-sparse bins.

where

$$n_1 = \left[\frac{\alpha_1}{\alpha_0}(n_0 - \tau_0) + \tau_1\right]_{N_1}. \tag{4.3}$$

Hence, $\mathcal{E}(\boldsymbol{\alpha}, \boldsymbol{\tau}, l)$, $l = 0, 1, \cdots, L-1$ is a discrete line segment with slope α_1/α_0 and offset $\boldsymbol{\tau}$.

The sampled signal, representing a slice of the data along $\mathcal{E}(\boldsymbol{\alpha}, \boldsymbol{\tau}, l)$, $l = 0, 1, \cdots, L-1$, can be expressed as

$$\begin{aligned} s(\boldsymbol{\alpha}, \boldsymbol{\tau}, l) &\triangleq x([\alpha_0 l + \tau_0]_{N_0}, [\alpha_1 l + \tau_1]_{N_1}) \\ &= \sum_{(a,\boldsymbol{\omega})\in\mathbb{S}} a e^{j2\pi\left(\frac{m_0[\alpha_0 l + \tau_0]_{N_0}}{N_0} + \frac{m_1[\alpha_1 l + \tau_1]_{N_1}}{N_1}\right)}, \; l = 0, 1, \cdots, L-1. \end{aligned} \tag{4.4}$$

Note that a slice can 'wrap around' within $x(\mathbf{n}), \mathbf{n} \in \mathcal{X}$ due to the modulo operation, and the sampling points along the line are always on the grid of $\mathcal{X}$, since $\boldsymbol{\alpha}, \boldsymbol{\tau}$ are on grid.

Taking an L-point DFT of the data slice defined in (4.4), for all $m \in [L]$, we get

$$\begin{aligned} \hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m) &\triangleq \frac{1}{L} \sum_{l\in[L]} s(\boldsymbol{\alpha}, \boldsymbol{\tau}, l) e^{-j2\pi\frac{lm}{L}} \\ &= \frac{1}{L} \sum_{(a,\boldsymbol{\omega})\in\mathbb{S}} a e^{j2\pi\left(\frac{m_0\tau_0}{N_0} + \frac{m_1\tau_1}{N_1}\right)} \sum_{l\in[L]} e^{j2\pi l\left(\frac{m_0\alpha_0}{N_0} + \frac{m_1\alpha_1}{N_1} - \frac{m}{L}\right)}. \end{aligned} \tag{4.5}$$

Let us assume that for all $m \in [L]$ and $\boldsymbol{\alpha}, [m_0, m_1]^T \in \mathcal{X}$,

$$\hat{f}(m) \triangleq \frac{1}{L} \sum_{l \in [L]} e^{j2\pi l \left(\frac{m_0 \alpha_0}{N_0} + \frac{m_1 \alpha_1}{N_1} - \frac{m}{L} \right)} \in \{0, 1\}. \tag{4.6}$$

The above assumption holds when $\frac{m_0\alpha_0}{N_0} + \frac{m_1\alpha_1}{N_1} - \frac{m}{L}$ is multiple of $1/L$, which can be expressed as

$$\left[\frac{L}{N_0} m_0 \alpha_0 + \frac{L}{N_1} m_1 \alpha_1 \right]_L = m. \tag{4.7}$$

It is clear that $L = \text{LCM}(N_0, N_1)$ satisfies (4.7), since in that case $L/N_0, L/N_1$ are integers. Moreover, $\text{LCM}(N_0, N_1)$ is the minimum length of a line, L, that satisfies (4.7) for arbitrary $\boldsymbol{\alpha}, [m_0, m_1]^T \in \mathcal{X}$; this can be proved using contradiction as follows.

Assume that there is length L such that $L < \text{LCM}(N_0, N_1)$, then at least either L/N_0 or L/N_1 is not an integer. Without loss of generality, let us assume that $\frac{L}{N_0} \notin \mathbb{Z}$. Then, the right hand side of (4.7) equals $[L/N_0]_L \notin [L]$ for $m_0 = 1, \alpha_0 = 1, m_1 = 0$, which is contradictory to the premise that (4.7) holds for any $[m_0, m_1]^T, [\alpha_0, \alpha_1]^T \in \mathcal{X}$.

When $\hat{f}(m) = 1$, i.e.,

$$\left[\frac{m_0\alpha_0}{N_0} + \frac{m_1\alpha_1}{N_1} - \frac{m}{L} \right]_1 = 0, \; [m_0, m_1]^T \in \mathcal{X}, \tag{4.8}$$

(4.5) can be simplified as

$$\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m) = \sum_{(a, \boldsymbol{\omega}) \in \mathbb{S}} a e^{j2\pi \left(\frac{m_0 \tau_0}{N_0} + \frac{m_1 \tau_1}{N_1} \right)}, \; m = 0, 1, \cdots, L-1, \tag{4.9}$$

which can be viewed as the 1-D projection of the 2-D frequencies satisfying (4.8). The solutions of (4.8) with respect to m are equally spaced points lie on line

$$\mathcal{E}([\alpha_1 L/N_1, -\alpha_0 L/N_0]^T, [m_0', m_1']^T, l), \; l = 0, 1, \cdots, L'-1, \tag{4.10}$$

where $[m_0', m_1']^T \in \mathcal{X}$ is one of the solutions of (4.8). The line of (4.10) is orthogonal to the line of (4.2). The orthogonality is necessary for the projected 2-D frequencies to be exactly recoverable.

Moreover, for certain choices of $\boldsymbol{\alpha}$, such projection is uniform, and $L' = N/L$. The uniformity of the projection means that the DFT coefficients of N grid locations of the $N_0 \times N_1$-point DFT are uniformly projected to the L entries of the L-point DFT along a line. Compared with a non-uniform projection, the uniform projection creates more 1-sparse bins, which allows for fewer iterations of MARS-SFT to exactly reconstruct the signal. The condition for orthogonal and uniform projection is stated in the following lemma.

Lemma 5. ***(Condition for orthogonal and uniform projection):*** *Consider the slice* *of the signal of (4.1), as defined in (4.4), with* $L = \mathrm{LCM}(N_0, N_1)$, $\boldsymbol{\alpha} \in \mathcal{A} \subset \mathcal{X}$, $\boldsymbol{\tau} \in \mathcal{X}$ *where* $\mathcal{A} \triangleq \{\boldsymbol{\alpha} : \boldsymbol{\alpha} \in \mathcal{X};\ (\alpha_0, \alpha_1),\ (\alpha_0, L/N_1),\ (\alpha_1, L/N_0)$ *are co-prime pairs*$\}$*. Then each entry of (4.9) is the projection of samples of* $\hat{x}(\mathbf{m})$, $\mathbf{m} \in \mathcal{P}_m \subset \mathcal{X}$*, where* $\mathcal{P}_m$, $m \in [L]$ *contains sample locations satisfying (4.8). Moreover,* $|\mathcal{P}_m| = N/L$, $\mathcal{P}_m \cap \mathcal{P}_{m'} = \emptyset$ *for* $m \neq m'$, $m, m' \in [L]$*. Thus,* $\hat{x}(\mathbf{m})$, $\mathbf{m} \in \mathcal{X}$ *is uniformly projected to (4.9).*

Proof. Please see Appendix H. □

A slice satisfying Lemma 5 is the longest slice that does not contain any duplicated samples. Thus, the L-point DFT along such slice captures the maximum information in the frequency domain with the least number of samples. The set $\mathcal{A}$ defined in Lemma 5 contains a large number of elements, providing sufficient randomness for frequency projection. For example, when $N_0 = N_1 = 4$, $\mathcal{A} = \{[1,1]^T, [1,2]^T, [1,3]^T, [2,1]^T, [2,3]^T, [3,1]^T, [3,2]^T\}$. This means that $\frac{|\mathcal{A}|}{|\mathcal{X}|} \approx 44\%$ of all the possible values of $\boldsymbol{\alpha}$ yield uniform projections. When $N_0 = N_1 = 256$, $|\mathcal{A}| = 39636$ and $\frac{|\mathcal{A}|}{|\mathcal{X}|} \approx 60\%$.

Fig. 4.2 shows an example of a time domain line designed to allow an orthogonal and uniform projection (red circle). The corresponding frequency domain line satisfying (4.8) for $m = 1$ is also shown (black circle); the two lines are orthogonal to each other and intercept at $[11, 1]^T$. The lengths of the time domain and the frequency domain lines are $16, 8$, respectively. Each line is composed of several line segments due to the modulo operation.

Remark 6. *In the L-point DFT of samples along a time-domain line with slope* α_1/α_0*, each entry represents a projection of the* 2*-D DFT along the line with slope* $-\alpha_0 N_1/(\alpha_1 N_0)$ *in the* $N_0 \times N_1$*-point DFT domain; the DFT-domain line is orthogonal to the time-domain line. This is closely related to the Fourier projection-slice theorem, which states that the Fourier transform of a projection is a slice of the Fourier transform of the projected object. While the classical projection is in the time domain and the corresponding slice is in the frequency domain, in the MARS-SFT case, the projection is in the DFT domain and the corresponding slice is in the sample (discrete-time) domain. The important difference between the Fourier projection-slice theorem and MARS-SFT is that while the former reconstructs the frequency domain of the signal via interpolation of frequency-domain slices, or reconstructs the time-domain samples by solving a system of linear equations relating the DFT along projections*

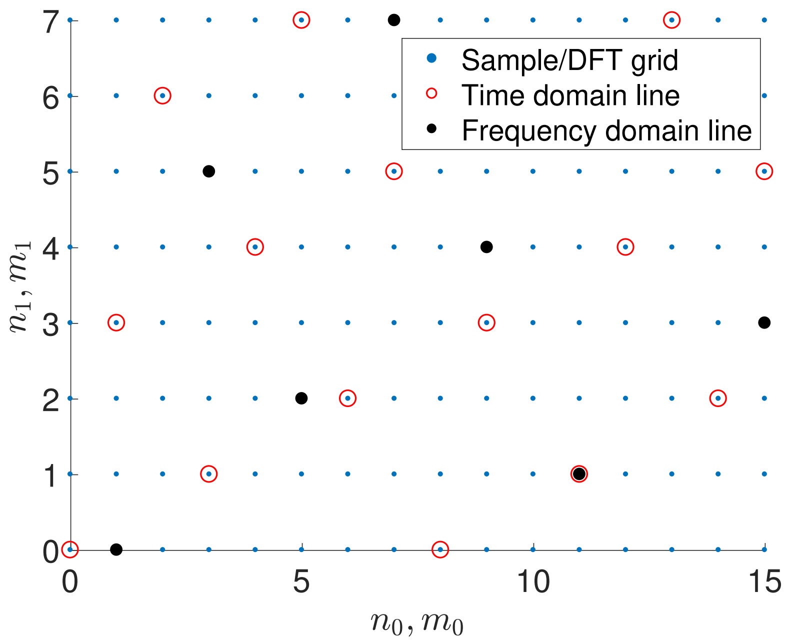


Figure 4.2: An orthogonal pair of time and frequency domain lines. $N_0 = 16, N_1 = 8, L = 16, \boldsymbol{\alpha} = [1, 3]^T, \boldsymbol{\tau} = [0, 0]^T$.

and the time-domain samples, the latter efficiently recovers the significant frequencies of the signal directly based on the DFT of time-domain 1*-D slices, i.e., samples along lines; this involves lower complexity.*

As it will be explained in the following, the efficiency of MARS-SFT is achieved by exploring sparsity in the frequency domain.

Let us assume that the signal is sparse in the frequency domain, i.e., $|\mathbb{S}| = O(L) << N$. Then, with high probability, $|\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m)| = |\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}_0, m)| = |\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}_1, m)| \neq 0$, where $\boldsymbol{\tau}_0 \triangleq [[\tau_0 + 1]_{N_0}, \tau_1]^T$, $\boldsymbol{\tau}_1 \triangleq [\tau_0, [\tau_1 + 1]_{N_1}]^T$. Thus, the m-th bin is 1-sparse, and it holds that

$$\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m) = ae^{j2\pi\left(\frac{m_0\tau_0}{N_0} + \frac{m_1\tau_1}{N_1}\right)}, \; (a, \boldsymbol{\omega}) \in \mathbb{S}. \tag{4.11}$$

In such case, the 2-D sinusoid, $(a, \boldsymbol{\omega})$, can be 'decoded' as

$$\begin{aligned} m_0 &= \left[\frac{N_0}{2\pi}\phi\left(\frac{\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}_0, m)}{\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m)}\right)\right]_{N_0}, \\ m_1 &= \left[\frac{N_1}{2\pi}\phi\left(\frac{\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}_1, m)}{\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m)}\right)\right]_{N_1}, \\ a &= \hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m)e^{-j2\pi(m_0\tau_0/N_0 + m_1\tau_1/N_1)}. \end{aligned} \tag{4.12}$$

This is the OFDM-trick adapted to MARS-SFT; such design requires sampling along three lines of the same slope but different offsets, allowing the frequency components to be decoded independently in each dimension.

In order to recover all the sinusoids in $\mathbb{S}$ efficiently, each iteration of MARS-SFT adopts a random choice of line slope from the set of $\mathcal{A}$ defined in Lemma 5. Furthermore, the contribution of the recovered sinusoids in the previous iterations is removed via a *construction-subtraction* approach so that the signal becomes sparser in subsequent iterations. Specifically, assuming that for the current iteration the line slope and offset parameters are $\boldsymbol{\alpha}, \boldsymbol{\tau}$, respectively, the recovered 2-D frequencies are projected into L frequency bins to construct the DFT of the slice taken along the line of $\mathcal{E}(\boldsymbol{\alpha}, \boldsymbol{\tau}, l)$, $l = 0, 1, \cdots, L-1$, i.e., $\hat{s}_r(\boldsymbol{\alpha}, \boldsymbol{\tau}, m) \triangleq \sum_{(a,\boldsymbol{\omega})\in\mathcal{I}_m} ae^{j2\pi\left(\frac{m_0\tau_0}{N_0}+\frac{m_1\tau_1}{N_1}\right)}$, $m = 0, 1, \cdots, L-1$, where $\mathcal{I}_m$, $m \in [L]$ represent the subsets of the recovered frequencies, i.e., $\mathcal{I}_m \triangleq \{(a, \boldsymbol{\omega}) : \boldsymbol{\omega} \text{ satisfies } (4.8)\}$, $m \in [L]$. Next, the L-point inverse DFT (IDFT), multiplied by L, is applied on $\hat{s}_r(\boldsymbol{\alpha}, \boldsymbol{\tau}, m)$, $m = 0, 1, \cdots, L-1$, from which the slice, $s_r(\boldsymbol{\alpha}, \boldsymbol{\tau}, l)$, $l = 0, 1, \cdots, L-1$, is constructed based on the previously recovered sinusoids. Subsequently, the constructed slice is subtracted from the slice of the current iteration. The pseudo-code of the MARS-SFT algorithm can be found in Appendix N.

4.4 Convergence of MARS-SFT

In this section, we investigate the convergence of MARS-SFT. First, let us look at a special case, where N_0, N_1 are co-prime.

Theorem 2. ***(One-projection theorem of MARS-SFT):*** *Consider the signal model of (4.1), where N_0, N_1 are co-prime and $0 \leq K \leq N$. The exact reconstruction of $\mathbb{S}$ via MARS-SFT only takes one iteration.*

Proof. Please see Appendix 2. □

In the Fourier projection-slice theorem, a band-limited signal of size $N_0 \times N_0$ can be exactly reconstructed by a single projection in the time domain. In the frequency domain, exact reconstruction is possible based on a single slice, provided that the slope parameters, α_0, α_1, of the line, along which the slice is evaluated are co-prime and the equality $\alpha_0 m_0 +$

$\alpha_1 m_1 = \alpha_0 m_0' + \alpha_1 m_1'$ holds only for $m_0 = m_0'$, $m_1 = m_1'$ when m_0, m_0', m_1, $m_1' \in [N_0]$; this is referred to as the one-projection theorem [52]. Theorem 2 is the one-projection theorem of MARS-SFT and provides the conditions for exact recovery of a signal with arbitrary sparsity level using only one projection in the frequency domain.

Remark 7. *The classic one-projection theorem and the one-projection theorem of MARS-SFT establish an unambiguous one-to-one mapping from a* 2*-D sequence to a* 1*-D sequence. Specifically, the former establishes the mapping of* 2*-D time-domain samples to the projected* 1*-D time-domain samples of length* N_0^2*; each entry of the DFT of the projection can be represented by a weighted summation of the* N_0^2 *samples. Hence, exact recovery of the time domain samples requires inverting a linear equation system containing at least* N_0^2 *equations. On the other hand, latter establishes the one-to-one mapping from the coefficients of the* $N_0 \times N_1$*-point DFT of the* 2*-D data to the coefficients of the* N*-point DFT of a slice of the* 2*-D data; such slice can be viewed as an ordering of the* 2*-D data into an* 1*-D sequence. The exact recovery of the* $N_0 \times N_1$*-point DFT of the data is achieved by the low-complexity OFDM-trick under the framework of MARS-SFT. We should note that if* N_0, N_1 *are co-prime, the* $N_0 \times N_1$*-point DFT can also be implement via an* N*-point DFT based on the Good-Thomas mapping [53], where the unambiguous mapping is achieved via the Chinese Remainder Theorem-based indexing.*

Next, the following theorem concludes the convergence of MARS-SFT for the general case. Compared to the special case, i.e., Theorem 2, the general case assumes $K/N << 1$ and applies for arbitrary N_0, N_1.

Theorem 3. ***(Convergence of MARS-SFT):*** *Consider the application of MARS-SFT on the signal model of (4.1). Then the average number of iterations required to recover* $\mathbb{S}$ *, i.e.,* T*, can be found by evaluating the following inequality*

$$\sum_{i \in [T]} M_i \geq K, \tag{4.13}$$

where $M_i = Q_i K_i$ *is the number of the recovered frequencies in the* i*-th iteration;* $K_i = K \prod_{k \in [i]} (1 - Q_k)$*, with* $K_0 = K$*, is the number of remaining frequencies that have not been recovered in the* i*-th iteration;* $Q_i = (1 - K_i/N)^{N/L-1}$ *is the probability of a remaining significant frequency to be projected into a* 1*-sparse bin, and thus be recovered in the* i*-th* *iteration.*

Proof. Please see Appendix J. □

Fig. 4.3 shows the relationship between T and K/L. When K/L is small, e.g., $K/L = 3$, T is small; this results in a low "big-Oh" overhead [3] of the algorithm. However, T grows super-linearly with K/L; the growth rate increases as K/N increases. In a non-sparse scenario, i.e., as K/N approaches 1, T is too large for MARS-SFT to be applicable. With the exception of the scenario in which N_0, N_1 are co-prime, MARS-SFT can fail in a non-spare scenario, in which none of the projection creates a 1-sparse bin.

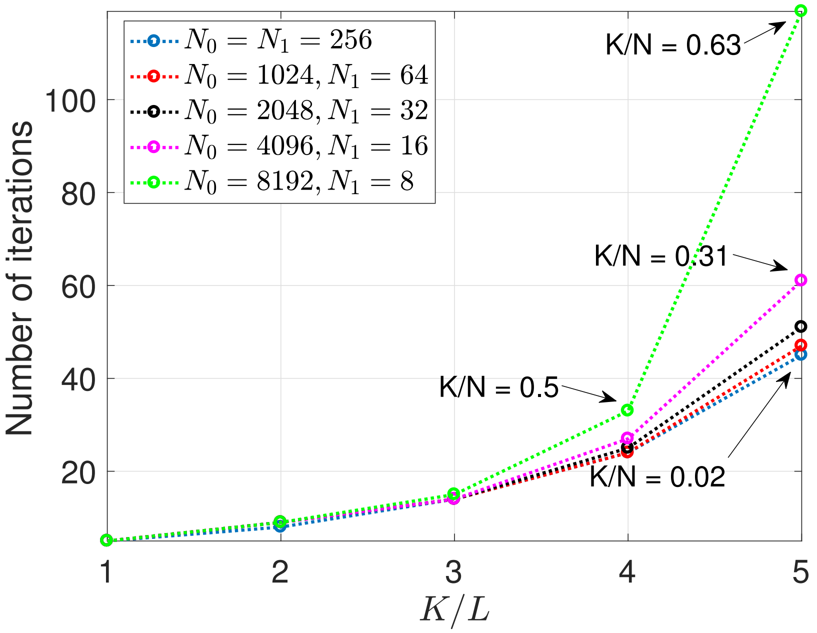


Figure 4.3: Number of iterations of MARS-SFT versus K/L.

Although in order to prove Theorem 3 we assume that the frequency distribution in the signal model of (4.1) is uniform, i.e., in the average case [3], as we will see in Section 4.6.4, numerical results show no significant difference in the convergence of MARS-SFT when the frequencies are clustered. This is because the multidimensional clustered frequencies are uniformly projected to one dimension due to the randomly generated line slopes of MARS-SFT.

4.5 Multidimensional Extension and Complexity Analysis

4.5.1 Multidimensional Extension

For the D-D ($D > 2$) case with the data cube size of $N_0 \times N_1 \times \cdots N_{D-1}$, the line length can be set as $L = \text{LCM}(N_0, \cdots, N_{D-1})$; the slope and offset parameters $[\alpha_0, \cdots, \alpha_{D-1}]^T$, $[\tau_0, \cdots, \tau_{D-1}]^T$ is randomly taken from $\mathcal{X}_D \triangleq [N_0] \times [N_1] \times \cdots [N_{D-1}]$. Each iteration extracts $D + 1$ L-length lines with a same random slope but different offsets from the D-D data cube. The 0-th line offset is set to be $[\tau_0, \cdots, \tau_{D-1}]^T$, while for the $(i + 1)$-th line with $0 \leq i \leq D - 1$, the offset for the i-th dimension is set to be $[\tau_i + 1]_{N_i}$. With such offset parameters, the frequencies can be decoded independently for each dimension.

4.5.2 Complexity analysis

MARS-SFT executes T iterations; in the 2-D case, each iteration uses $3L$ samples, since 3 L-length slices, with $L = \text{LCM}(N_0, N_1)$ are extracted in order to decode the two frequency components of a 2-D sinusoid (see (4.12)). Hence, the sample complexity is $O(3TL) = O(TL)$. The core processing is the L-point 1-D DFT, which can be implemented via an FFT with the computational complexity of $O(L \log L)$. The L-point IDFT in the construction-subtraction procedure can also be implemented via an FFT. In addition to the FFT, each iteration needs to evaluate up to L frequencies. Hence the computational complexity of MARS-SFT is $O(T(L \log L + L)) = O(TL \log L)$. If we let T equal to $T_{max} \in \mathbb{N}$, which is a sufficiently large constant to allow convergence for a given signal size and a range of K, then, the sample and computational complexity become $O(L)$ and $O(L \log L)$, respectively. For $K = O(L)$, MARS-SFT achieves the lowest sample and computational complexity, i.e., $O(K)$ and $O(K \log K)$, respectively, of all known SFT algorithms [3, 32].

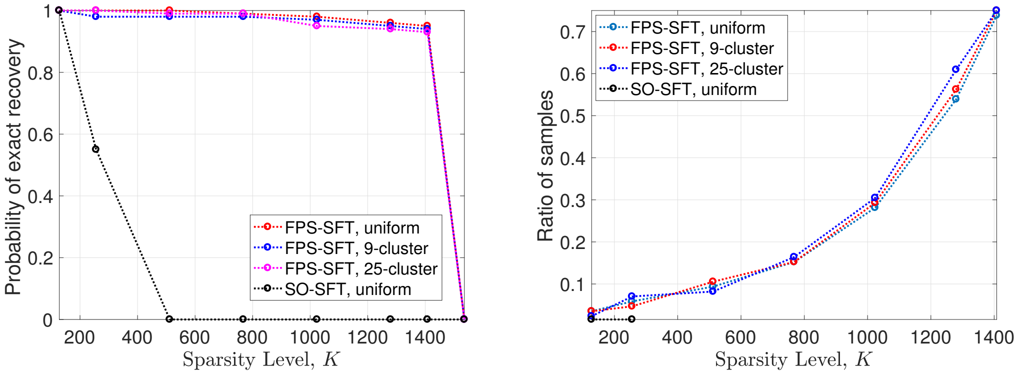
In general, in the D-D case, according to the multidimensional extension [54], it is easy to see that the sample and computational complexity of MARS-SFT are $O(DK)$ and $O(DK \log(DK))$, respectively when $K = O(L)$.

4.6 Numerical Results

In this section, we provide some numerical results to verify the theoretical findings related to the proposed MARS-SFT. Unless stated otherwise, the size of the test data is set equal to $N_0 = N_1 = 256$. We simulate cases for frequencies which are uniformly distributed and also for frequencies which are clustered; for clustered cases, we consider clusters of 9 and 25 frequencies. The experimental results are averaged over 100 iterations of Monte Carlo simulation.

4.6.1 Comparison between MARS-SFT and SO-SFT

We compare the performance of SO-SFT and the proposed MARS-SFT for the 2-D case. For MARS-SFT the line length is set to $L = N_0$, and each iteration uses $3N_0$ samples. We limit the number of iterations to $T_{max} = N/(3L) \approx 85$, which corresponds to roughly 100% samples of the input data. Fig. 4.4(a) shows the probability of exact recovery versus the level of sparsity for the two methods. When the signal is very sparse, i.e., $K < N_0/2$, SO-SFT has high probability of exact recovery, while for moderate level of sparsity, i.e., $K > N_0$, SO-SFT fails with high probability. Moreover, SO-SFT only works for the scenario in which the frequencies are distributed uniformly, while it fails when there exists even a single frequency cluster. On the contrary, MARS-SFT applies to signals with a wide range of sparsity levels. For instance, its success rate is almost 97% when $K = 5N_0$. In all cases, the success rates drop to 0 when $K = 6N_0$, since then, the exact recovery needs roughly 100 iterations, which exceeds T_{max}. Fig. 4.4 (b) shows the ratio of samples used by the MARS-SFT and SO-SFT for exact recovery over the total number of data samples, N. The figure shows that the sparser the signal, the fewer samples are required by MARS-SFT; for example, when $K = N_0$, only 5.9% of the signal samples are required. SO-SFT only needs 1.6% of signal samples in very sparse scenarios, while it fails in less sparse, or non-uniformly distributed frequency cases. The performance of MARS-SFT is similar for both uniformly-distributed and clustered frequency cases at the same sparsity level; this is due to the randomized projections that can effectively isolate the 2-D frequencies into 1-sparse bins, even when the signal is less sparse (K is large) and the frequencies are clustered. Note that the super linearity in the growth of the ratio of samples with K arises because of the super-linear of the number of iterations of MARS-SFT with K.



(a)

(b)

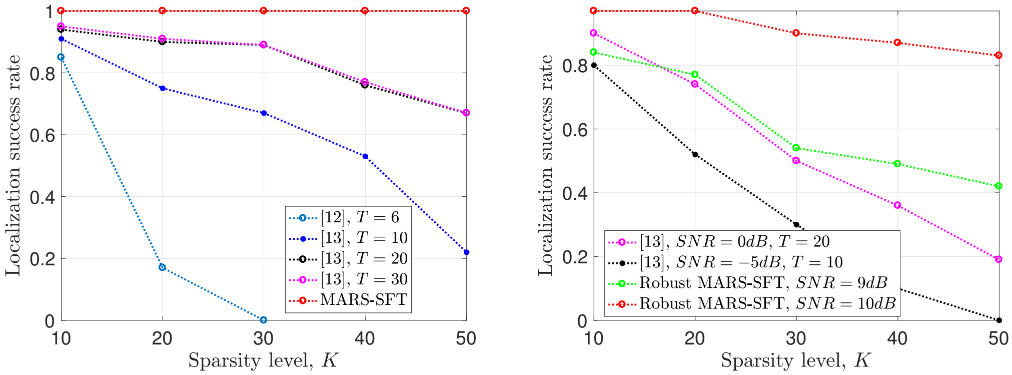
Figure 4.4: Comparison between MARS-SFT and SO-SFT. (a) Probability of exact recovery versus sparsity level, K. (b) Ratio of samples needed for exact recovery versus K.

4.6.2 Comparison between MARS-SFT and the SFT of [1, 2]

We compare MARS-SFT and its robust extension with the SFT of [1, 2] in 2-D cases. The main difference between the SFT of [2] and the SFT of [1] is that the former takes the slices only from the borders and the diagonals from the input data matrix, while the latter also takes slices along many lines with predefined slopes; this increases the degrees of freedom of projecting 2-D frequencies onto 1-D lines.

Fig. 4.5 (a) shows the frequency localization performance of SFT of [1, 2] versus K in noiseless cases. Compared to MARS-SFT, the SFT of [1, 2] only succeeds in very sparse scenarios. For instance, when $K = 50$, the best success rate that the SFT of [1] can achieve is approximately 67%, while the success rate of MARS-SFT is 100%. One way to increase the success rate of SFT of [1] is to use a larger T at the expense of increasing complexity. However, the success rate saturates when T grows beyond a certain value. For instance, the success rates corresponding to $T = 20$ and $T = 30$ are similar.

Aside from sensitivity to sparsity level, the SFT of [1, 2] is more robust to noise than the proposed robust MARS-SFT. For instance, in Fig. 4.5 (b), one can see that when $K < 30$,



(a) (b)

Figure 4.5: Comparison between MARS-SFT and SFT of [1, 2]. The robust MARS-SFT is adopted in noisy cases. (a) Localization success rate versus K in noiseless cases. (b) Localization success rate versus K in noisy cases. Frequencies are on the grid.

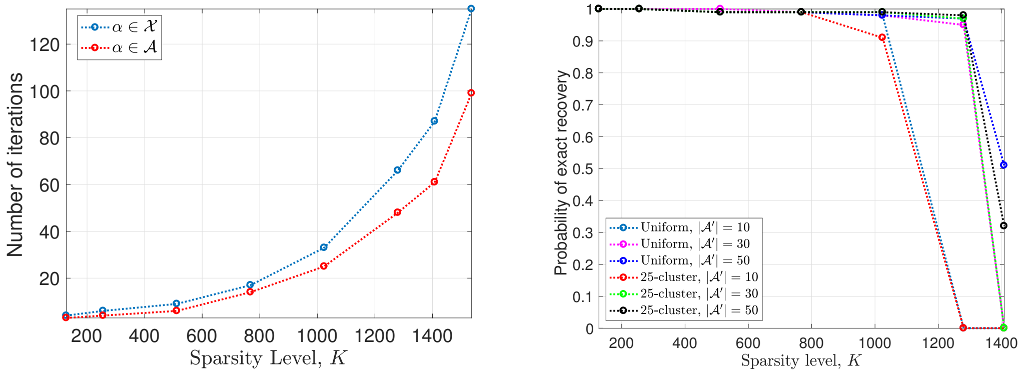
the success rate of the SFT of [1] for SNR equal to $-5dB$ is similar to that of robust MARS-SFT applied to SNR of $9dB$. However, when the SNR is greater than $11dB$, the success rate of robust MARS-SFT approaches 100%.

The computation of the SFT of [1, 2] is significantly slower as compared to that of robust MARS-SFT, as the computation complexity of the former is $O((N + K^3)\log N)$ [1] in the 2-D case; this is even greater than that of the FFT.

4.6.3 Line Slope of MARS-SFT

From Lemma 5, choosing the line slope parameters $\boldsymbol{\alpha}$ randomly from the set $\mathcal{A}$ for each iteration of MARS-SFT, as opposed to choosing them at random from the set $\mathcal{X}$, results in reduced number of iterations for exact recovery. This is because as compared to the latter case, in the former case, more 1-sparse bins are likely to be created in each iteration due to the uniformity of projections. In Fig. 4.6 (a), we compare the number of iterations of MARS-SFT when $\boldsymbol{\alpha}$ is chosen from $\mathcal{A}$ and $\mathcal{X}$. The figure confirms the expectation from Lemma 5.

The high probability of exact recovery of MARS-SFT in less sparse cases is due to the abundance of degrees of freedom in frequency projection, which requires a sufficiently large $|\mathcal{A}|$. When $N_0 = N_1 = 256$, it is easy to verify that $|\mathcal{A}| = 39639$. Fig. 4.6 (b) shows the probability of exact recovery versus sparsity when we use subsets of $\mathcal{A}$ of different support sizes. The slope parameter set, $\mathcal{A}'$, in each experiment, is created by randomly picking a subset of $\mathcal{A}$ with a specific size of support. The figure shows that the less sparse the signal (the larger the K), the larger size of $\mathcal{A}'$ is needed to achieve a high probability of exact recovery. $|\mathcal{A}'|$ should be large enough so that for each iteration of MARS-SFT, a distinct slope can be obtained from $\mathcal{A}'$ with high probability. Compared to the uniformly distributed frequency cases, the clustered frequency cases require a larger $|\mathcal{A}'|$, since the latter requires larger degrees of freedom than the former in order to isolate the clustered frequencies by randomly projecting those frequencies to distinct 1-sparse bins of the DFT along lines.



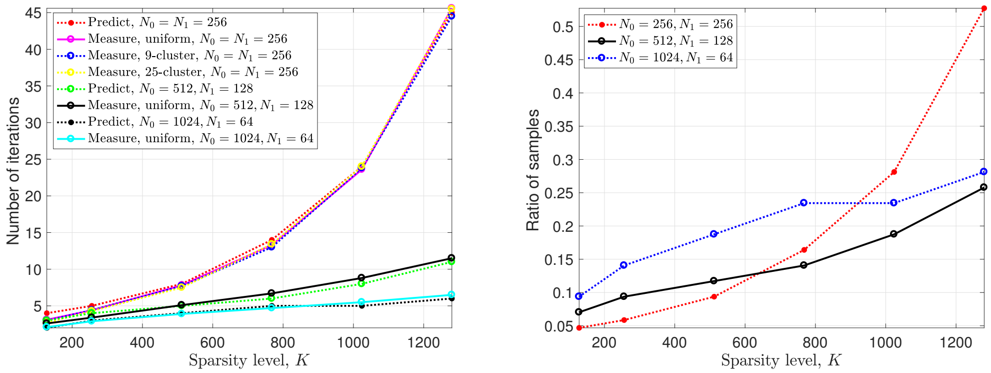
(a) (b)

Figure 4.6: Effect of the line slope to the MARS-SFT. (a) Number of iterations of exact recovery versus sparsity. (b) Effect of the size of the slope parameter set ($|\mathcal{A}'|$) to the exact recovery probability.

4.6.4 Convergence of MARS-SFT

We verify the average number of iterations of MARS-SFT in order to exactly recover the signal (see Theorem 3).

The relationship between the number of iterations, T, and sparsity level, K, for different data dimensions of the same number of samples (i.e., different N_0, N_1 but the same N) are shown in Fig. 4.7 (a); the figure shows the theoretical values based on Theorem 3 and also the results of simulations. As expected, for all cases, T increases as K increases; the rate of increase in the cases of $N_0 = N_1 = 256$ is greater than the cases when $N_0 = 512, N_1 = 128$ and $N_0 = 1024, N_1 = 64$. Also, the former case requires more iterations than the latter two cases. This is because for the three cases the the line length, L, equals $256, 512$ and 1024, respectively. A larger L leads to a higher probability of creating more 1-sparse bins in each iteration of MARS-SFT, which results in faster convergence of the algorithm. The clustered frequencies do not require larger T as compared to the uniformly distributed frequencies, which suggests that MARS-SFT is efficient in solving non-uniformly distributed frequencies. The number of samples used by MARS-SFT depends both on T and L. Fig. 4.7 (b) shows that when the signal is very sparse, i.e., $K < 640$, the equal-length case ($N_0 = N_1$) uses the least number of samples, while for less sparse cases, the number of samples required by MARS-SFT is less in the cases when $N_0 = 512, N_1 = 256$ and $N_0 = 1024, N_1 = 64$ than the case when $N_0 = 256, N_1 = 256$.



(a) (b)

Figure 4.7: Number of iterations of MARS-SFT. (a) Number of iterations versus sparsity. (b) Ratio of samples versus sparsity.

According to Theorem 2, when N_0, N_1 are co-prime, the set $\mathbb{S}$ of the signal model (4.1) can be reconstructed exactly based on only one iteration of MARS-SFT. Fig. 4.8 shows that the one-projection based reconstruction is exact even when the frequencies are not sparse, i.e., $K/N \approx 0.65$.

4.7 Summary

We have proposed MARS-SFT, a multidimensional sparse Fourier transform that is inspired by the Fourier projection-slice theorem. We have shown that MARS-SFT can handle less sparse data in the frequency domain, and enjoys low sample and computational complexity. The sample and computational complexity of MARS-SFT achieves the lowest complexity

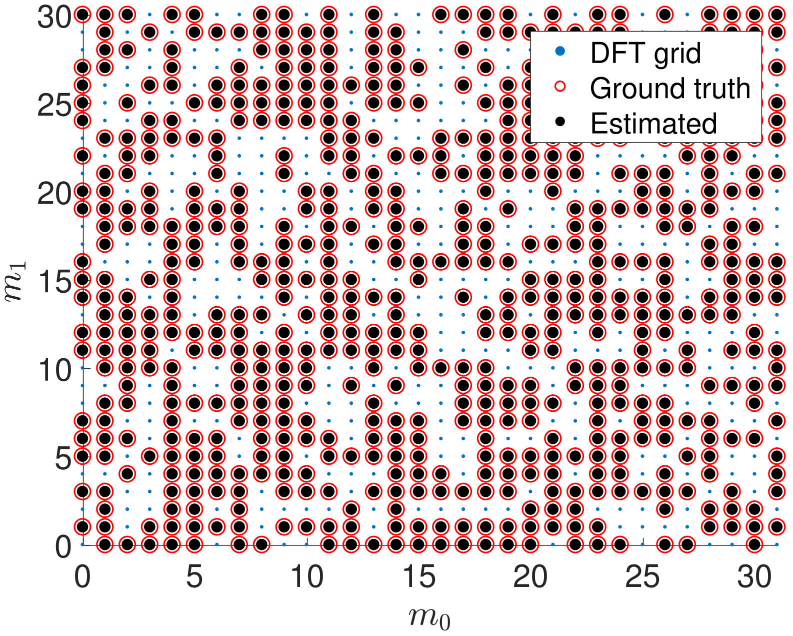


Figure 4.8: Original (red circles) and recovered frequencies (black dots) using one iteration of MARS-SFT. $N_0 = 32, N_1 = 31, K = 20N_0 = 640$.

among all know SFT algorithms when the sparsity is of the same order of the slice length. The relationship between MARS-SFT and the Fourier projection-slice theorem has been discussed. Especially, the connections between the one-projection theorems under the context of MARS-SFT and the Fourier projection-slice theorem has been exploited; the classic one-projection theorem and the one-projection theorem of MARS-SFT establish an unambiguous one-to-one mapping from a 2-D sequence to a 1-D sequence.

Chapter 5

Robust MARS-SFT and Its Application in DBF Radar

The MultidimensionAl Random Slice based Sparse Fourier Transform (MARS-SFT) is an efficient implementation of the Discrete Fourier Transform (DFT) for multidimensional signals that are sparse in the frequency domain. For a K-sparse signal, MARS-SFT achieves sample complexity of $O(K)$ and computational complexity of $O(K \log K)$. While MARS-SFT considers the ideal scenario, i.e., exactly sparse data that contain on-grid frequencies, in this chapter, we propose the robust MARS-SFT, which applies to noisy signals that contain off-grid frequencies; such signals arise in radar applications. The proposed robust MARS-SFT employs a windowing step and a voting-based frequency decoding step; the former reduces the frequency leakage of off-grid frequencies below the noise level, thus preserving the sparsity of the signal, while the latter significantly lowers the frequency localization error stemming from the noise. The performance of the proposed method is demonstrated both theoretically and numerically. We apply the robust MARS-SFT to automotive Digital Beamforming (DBF) radar signal processing, which achieves significant savings in sample and computational complexity.

5.1 Introduction

While the MARS-SFT considers the ideal scenario, i.e., noiseless frequency-domain sparse data containing frequencies on the DFT grid, in realistic applications, the data is usually noisy and contains off-grid frequencies. One example of such data is the received signal of the DBF automotive radar, which usually employs a Frequency Modulation Continuous Waveform (FMCW). After demodulation of the returned signal, each radar target can be expressed as a D-dimensional (D-D) complex sinusoid [47], whose frequency in each dimension is related to target parameters, e.g., range, Doppler and direction of arrival (DOA).

When the number of targets is much smaller than the number of samples, such signal is sparse in the D-D frequency domain. Due to the continuous nature of those parameters, the frequencies are also continuous and thus are typically off-grid. Meanwhile, the radar signal contains noise, which makes the signal approximately sparse. MARS-SFT suffers from the frequency leakage caused by the off-grid frequencies; also, the frequency localization procedure of MARS-SFT is prone to errors since the OFDM-trick is sensitive to noise [22]. We address these issues by extending MARS-SFT to a robust version. Robust MARS-SFT employs a windowing technique to reduce the frequency leakage caused by the off-grid frequencies and a voting based frequency decoding procedure to significantly reduce the localization error stemming from noise.

The off-grid frequencies are also addressed in the Robust Fourier Transform (RSFT) proposed in Chapter 2. In RSFT, the computational savings are achieved by folding the input D-D data cube into a much smaller data cube, on which a reduced sized D-D FFT is applied. Although the RSFT is more computationally efficient as compared to the FFT-based methods, its sample complexity is the same as the FFT-based algorithms. Essentially, the high sample complexity of RSFT is due to its two stages of windowing procedures, which are applied to the entire data cube to suppress the frequency leakage. In the proposed robust MARS-SFT, instead of applying the multidimensional window on the entire data, the window, while still designed for the full-sized data, is applied on samples along lines only, which does not cause overhead in sample complexity.

5.2 Signal Model

The signal model (4.1) of MARS-SFT assumes that the frequencies lie on a grid, and there is no noise. A more realistic signal model, addressing continuous-valued frequencies in $[0, 2\pi)^2$ and also noise, is the following

$$r(\mathbf{n}) = y(\mathbf{n}) + n(\mathbf{n}) = \sum_{(a,\boldsymbol{\omega})\in\mathbb{S}'} ae^{j\mathbf{n}^T\boldsymbol{\omega}} + n(\mathbf{n}),\ \mathbf{n}\in\mathcal{X} \triangleq [N_0]\times[N_1], \tag{5.1}$$

where $y(\mathbf{n}) \triangleq \sum_{(a,\boldsymbol{\omega})\in\mathbb{S}'} ae^{j\mathbf{n}^T\boldsymbol{\omega}}$ is the superposition of $K' = |\mathbb{S}'|$ continuous-frequency sinusoids; $(a, \boldsymbol{\omega})$ denotes a significant 2-D sinusoid in $\mathbb{S}'$, whose complex amplitude and frequency are $a, \boldsymbol{\omega} \triangleq [\omega_0, \omega_1]^T \in [0, 2\pi)^2$, respectively, and it holds that $0 < a_{min} \le |a| \le a_{max}$. The noise, $n(\mathbf{n})$, is assumed to be i.i.d., circularly symmetric Gaussian, i.e., $\mathcal{CN}(0, \sigma_n)$. The SNR of a sinusoid with amplitude a is defined as $SNR \triangleq (|a|/\sigma_n)^2$.

Conventionally, $\mathbb{S}'$ can be estimated via a 2-D $N_0 \times N_1$-point DFT applied on signal (5.1), after windowing the signal with a 2-D window $w(\mathbf{n})$, used to suppress frequency leakage generated by off-grid frequencies. Assuming that the peak to side-lobe ratio (PSR) of the window is large enough, such that the side-lobes of each frequency in $\mathbb{S}'$ can be neglected in the DFT domain, then the signal contribution in the DFT domain corresponds to a set of on-grid frequencies, i.e., $\mathbb{S} \triangleq \{(a, \boldsymbol{\omega}) : \boldsymbol{\omega} \triangleq [2\pi m_0/N_0,\, 2\pi m_1/N_1]^T,\, [m_0, m_1]^T \in \mathcal{X}\}$ with $K' < K = |\mathbb{S}| << N$. Hence, the sample domain signal component associated with the window $w(\mathbf{n})$ and $\mathbb{S}$ can be approximated by (4.1). Note that since the windowing degrades the frequency resolution, each continuous-valued frequency in $\mathbb{S}'$ is related to a cluster of digital frequencies in $\mathbb{S}$; $\mathbb{S}$ can be estimated from the DFT of the signal, and then lead to the frequencies in $\mathbb{S}'$ via quadratic interpolation [55].

5.3 Window Design

To address the issue of off-grid frequencies, we apply a window $w(\mathbf{n}), \mathbf{n} \in \mathcal{X}$ on the signal of (5.1). The Peak-to-Sidelobe Ratio (PSR) of the window, ρ_w, is designed such that the side-lobes of the strongest frequency are below the noise level, hence the leakage of the significant frequencies can be neglected and the sparsity of the signal in the frequency domain can be preserved to some extent. Lemma 6 reflects the relationship between ρ_w and the maximum SNR of the signal.

Lemma 6. ***(Window design for the robust MARS-SFT):*** *Consider $\hat{r}(\mathbf{m})$, which is the $N_0 \times N_1$-point DFT of the windowed signal of (5.1). Let $\mathbf{W} \in \mathbb{R}^{N_0 \times N_1}$ be the matrix generated by the window function of $w(\mathbf{n}), \mathbf{n} \in \mathcal{X}$. In order to achieve a sufficient suppression of frequency leakage, the PSR of the window, ρ_w, should satisfy*

$$\rho_w > \frac{2\|\mathbf{W}\|_1}{\sqrt{\pi}\|\mathbf{W}\|_2}\sqrt{SNR_{max}}, \tag{5.2}$$

Where $SNR_{max} \triangleq a_{max}^2/\sigma_n^2$, and a_{max}, σ_n is defined in (5.1).

Proof. Please see the proof in Appendix K. □

Note that while the window is designed for the entire data cube, the windowing is applied only to the sampled locations, which does not increase the sample complexity of the robust MARS-SFT.

5.4 Voting-Based Frequency Decoding

When the signal is approximately sparse, the frequencies decoded by (4.12) are not integers. Since we aim to recover the gridded frequencies, i.e., $\mathbb{S}$ of (4.1), the recovered frequency indices are rounded to the nearest integers. When the SNR is low, the frequency decoding could result in false frequencies; those false frequencies enter the future iterations and generate more false frequencies. To suppress the effect of false frequencies, motivated by the classical m-out-of-n radar signal detector [9], the robust MARS-SFT adopts an n_d-out-of-n_s voting procedure in each iteration. Specifically, within each iteration, n_s sub-iterations are applied; each sub-iteration adopts randomly generated line slope and offset parameters and recovers a subset of frequencies, $\mathbb{S}_i, i \in [n_s]$. Within those frequency sets, a given frequency could be recovered by n out of n_s sub-iterations. For a true significant frequency, n is typically larger than that of a false frequency, thus only those frequencies with $n \geq n_d$ are retained as the recovered frequencies of the current iteration. When (n_s, n_d) is properly designed, the false frequencies can be reduced significantly.

5.5 Lower Bound of the Probability of Correct Localization and Convergence of Robust MARS-SFT

The probability of decoding error is related to the SNR, signal sparsity and also the parameters (n_s, n_d) of the robust MARS-SFT. In the following, we provide a lower bound for the probability of correct localization of the significant frequencies for each iteration, from which one can study convergence of robust MARS-SFT, i.e., the number of iterations needed in order to recover all the significant frequencies with sufficient SNR.

As explained in Section 5.2, a 2-D continuous-valued sinusoid $(a, \boldsymbol{\omega}) \in \mathbb{S}'$ of (5.1) is associated with a cluster of 2-D on-grid sinusoids $\mathbb{S}_0 \subseteq \mathbb{S}$ of (4.1). Let us assume that the sinusoid $(a_d, 2\pi[m_0/N_0, m_1/N_1]^T) \in \mathbb{S}_0$ with $[m_0, m_1]^T \in \mathcal{X}$ has the largest absolute amplitude among all members of $\mathbb{S}_0$. In addition, let us assume that the SNR of $(a, \boldsymbol{\omega})$ is sufficiently high. Then the probability of correctly localizing $(a_d, 2\pi[m_0/N_0, m_1/N_1]^T)$ in each iteration is lower bounded by

$$P_d \triangleq \sum_{n'_d=n_d}^{n_s} \binom{n_s}{n'_d} (P_1 P_w)^{n'_d} (1 - P_1 P_w)^{n_s - n'_d}, \tag{5.3}$$

where $P_1 \triangleq (1 - |\mathbb{S}''|/N)^{N/L-1}$ with $L = \text{LCM}(N_0, N_1)$ is the probability of a frequency in $\mathbb{S}''$ being projected to a 1-sparse bin, and $\mathbb{S}''$ with $\mathbb{S}'' \subseteq \mathbb{S}$ represents the remaining frequencies to be recovered in subsequent iterations; $P_w \triangleq (1 - P_u)(1 - P_v)$ is the lower bound of the probability of correct localization for a 2-D frequency that is projected into an 1-sparse bin in one sub-iteration; P_u, P_v are the upper bounds of the probability of localization error for the two frequency components, m_0, m_1, respectively, which are defined as

$$\begin{aligned} P_u &\triangleq \left(\sigma_p(1 - f_{|a_n|}(\delta_u))\right)^2, \\ P_v &\triangleq \left(\sigma_p(1 - f_{|a_n|}(\delta_v))\right)^2, \end{aligned} \tag{5.4}$$

where $\delta_u \triangleq a\pi\|\mathbf{W}\|_1/(2NN_0)$, $\delta_v \triangleq a\pi\|\mathbf{W}\|_1/(2NN_1)$, with $\mathbf{W} \in \mathbb{R}^{N_0 \times N_1}$ denoting the window applied on the data; σ_p, with $\frac{1}{2} \leq \sigma_p \leq \frac{1}{2\pi}$ is the parameter determined by the phases of the error vectors contained in the 1-sparse bin; $f_{|a_n|}(x)$ is the cumulative distribution function of the Rayleigh distribution, which is defined as

$$f_{|a_n|}(x) \triangleq 1 - e^{-x^2/(2\sigma^2_{a'_n})},\ x > 0, \tag{5.5}$$

where $\sigma^2_{a'_n} \triangleq \sigma^2_n \|\mathbf{W}\|_2^2/(2NL)$.

The proof of (5.3) can be found in Appendix L. Essentially, (5.3) represents the complementary cumulative binomial probability resulted from the n_d-out-of-n_s voting procedure, where the success probability of each experiment, i.e., localizing $(a_d, 2\pi[m_0/N_0, m_1/N_1]^T)$ in each sub-iteration of robust MARS-SFT is $P_1 P_w$. When $K = |\mathbb{S}|$ is known, (5.3) can be applied to estimate the largest number of iterations (the upper bound) of robust MARS-SFT in order to recover all the frequencies in $\mathbb{S}$ since the least number of recovered frequencies in each iteration can be estimated by $|\mathbb{S}''|P_d$.

5.6 Numerical Results

5.6.1 Effect of Windowing on Frequency Localization of Robust MARS-SFT

For data that contains off-grid frequencies, the PSR of the window, ρ_w, should be sufficiently large in order to reduce the side-lobes of the significant frequencies (see Lemma 6). However, the larger the ρ_w, the wider the main-lobe of the window, which results in larger frequency clusters in the DFT domain, and thus larger $|\mathbb{S}|$ of the signal model of (4.1), i.e., a less sparse signal. Moreover, the larger the ρ_w, the smaller the SNR of the windowed signal, which leads to larger frequency localization error. Fig. 5.1 (a) shows the performance of windows with various ρ_w for signals of various SNR_{max} and sparsity level, $K' = |\mathbb{S}'|$ (see (5.1)). According to (5.2), for signals with SNR_{max} equal to $20dB$ and $30dB$, the ρ_w of the window should be larger than $56dB$ and $60dB$, respectively. In those two cases, the frequency localization success rate, i.e., the ratio of number of correctly localized frequencies over the number of remaining significant frequencies in each iteration of robust MARS-SFT appears to be the highest when ρ_w equal $60dB$ and $70dB$, respectively.

Fig. 5.2 demonstrates localization of off-grid 2-D frequencies of robust MARS-SFT using the Dolph-Chebyshev window, for various values of ρ_w. A window with insufficient ρ_w leads to miss detections and false alarms (see Fig. 5.2 (a)), while a window with sufficiently large ρ_w yields good performance in frequency localization, with a tradeoff of causing larger frequency cluster sizes (see Fig. 5.2 (b)).

5.6.2 Effect of Voting on Frequency Localization of Robust MARS-SFT

The n_d-out-of-n_s voting in frequency decoding procedure of robust MARS-SFT can significantly reduce the false alarm rate. For a fixed n_s, larger n_d/n_s results in smaller false alarm rate. However, the smaller the false alarm rate, the larger the number of the iterations required to recover all the significant frequencies. Figs. 5.3 and Fig. 5.2 (b) show the examples of 2-D frequency recovery using different values of (n_s, n_d). In Fig. 5.3 (a), we set $(n_s, n_d) = (1, 1)$, which reduces the frequency localization of robust MARS-SFT to that of MARS-SFT, i.e., without voting. In this case, one can see that many false frequencies are generated. Figs. 5.3 (b), (c) show the frequency localization result with (n_s, n_d) equal

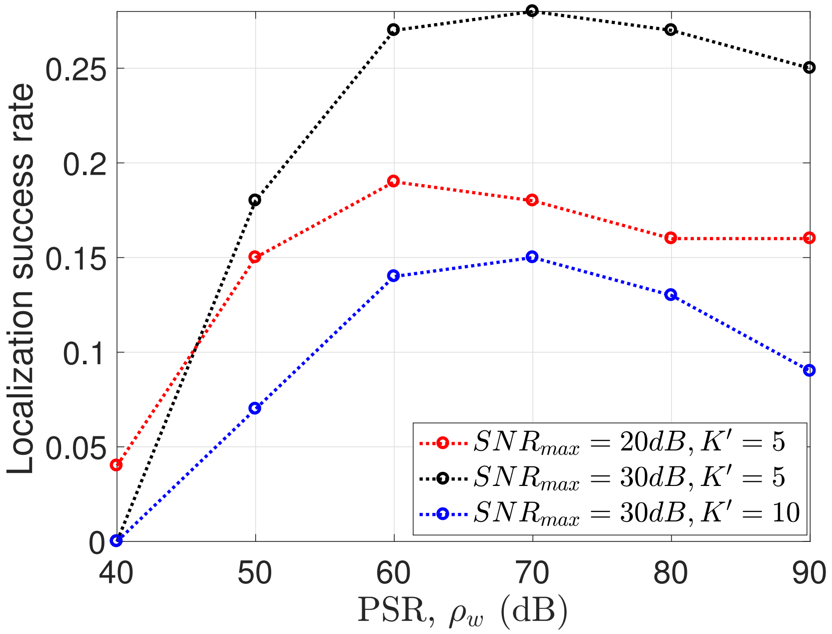
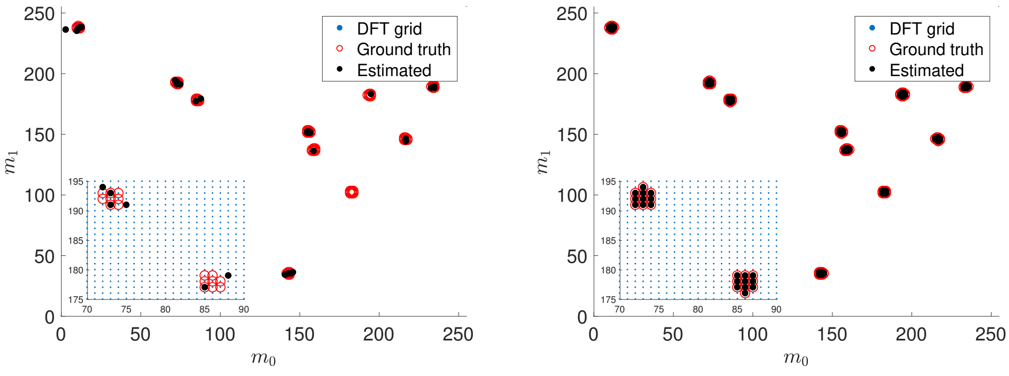


Figure 5.1: Frequency localization success rate of the first iteration of robust MARS-SFT versus window PSR. The Dolph-Chebyshev windows with various PSR is applied. $(n_s, n_d) = (3, 2)$.

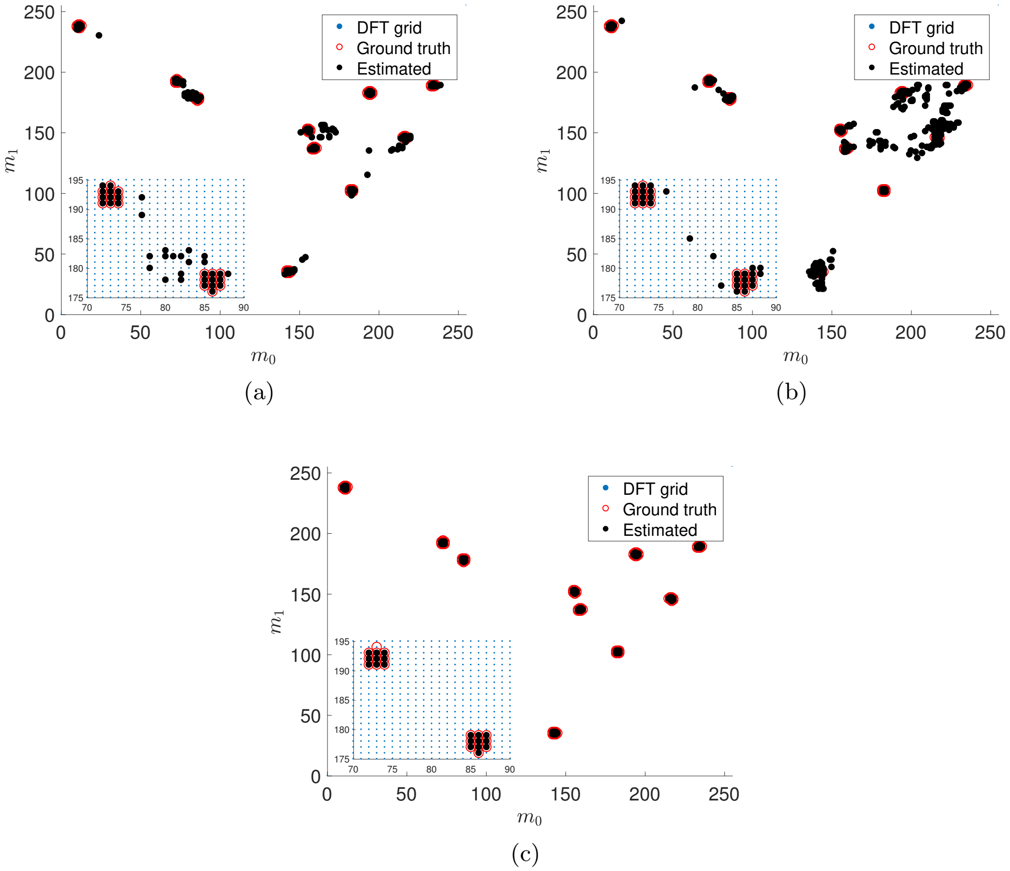


(a)

(b)

Figure 5.2: 2-D frequency recovery with different windows. $K' = 10$, $\sigma_n = 1, a_{min} = a_{max}$, $SNR_{max} = 30dB$, $(n_s, n_d) = (3, 2), T = 30$. The ground truth represents the discrete frequency clusters of $\mathbb{S}$. The PSR of the window equals $45dB$ in (a) and $70dB$ in (b).

to $(3,1),(3,3)$, respectively; while the former generates large amount of false frequencies, the latter exhibits miss detection, which implies the insufficiency of number of iterations of robust MARS-SFT used in this case. Fig. 5.2 (b) shows the ideal performance when (n_s, n_d) is designed as $(3,2)$.



(a)

(b)

(c)

Figure 5.3: Effect of voting on 2-D frequency recovery. $K' = 10$, $\sigma_n = 1, a_{min} = a_{max}$, $SNR_{max} = 30dB$. $T = 30$. Dolph-Chebyshev windows with $\rho_w = 70dB$ is applied. (a) $(n_d, n_s) = (1,1)$. (b) $(n_d, n_s) = (3,1)$. (c) $(n_d, n_s) = (3,3)$.

5.6.3 Effect of SNR and Sparsity Level on the Convergence of Robust MARS-SFT

The average number of iterations of robust MARS-SFT to recover all the significant frequencies is affected by the SNR and the sparsity level of the signal. A low SNR and less sparse signal require a large number of iterations. As discussed in Section 5.5, we are able to estimate the largest number of iterations that recover all the significant frequencies of sufficient SNR. Fig. 5.4 shows the theoretical and simulated number of iterations of robust MARS-SFT for signals with various SNR and sparsity level; the upper bounds of number of iterations are consistent with the simulated results.

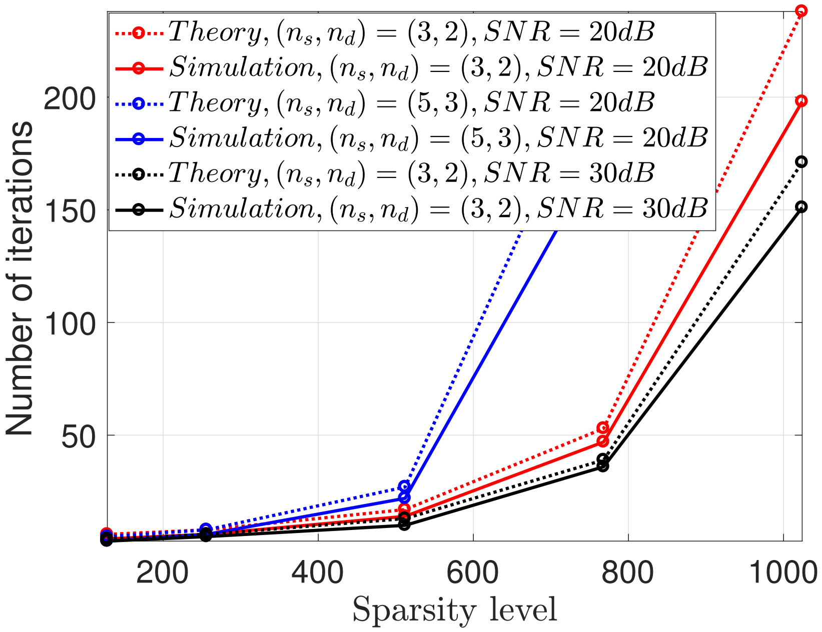


Figure 5.4: Effect of SNR and sparsity level on the convergence of robust MARS-SFT.

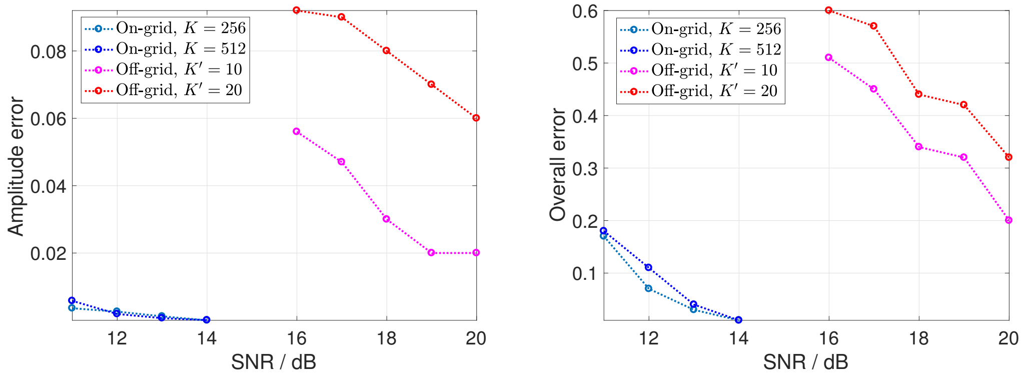
5.6.4 Amplitude and Overall Error of the Robust MARS-SFT

We investigate the amplitude and overall error between the original signal and the signal recovered by the robust MARS-SFT.

Let us define the set of $\mathcal{C}$ as the intersection of the reconstructed sinusoids $\mathbb{S}_r$ and the sinusoids contained in the windowed original signal, i.e., $\mathbb{S}'$ of (5.1), and $\mathcal{C} \triangleq \{a_r, a, \boldsymbol{\omega} : (a_r, \boldsymbol{\omega}) \in \mathbb{S}_r, (a, \boldsymbol{\omega}) \in \mathbb{S}'\}$. Then the amplitude error is defined as $\sum |a - a_r|^2 / \sum |a|^2$, where the summation is over the set of $\mathcal{C}$. The overall error is defined as $\|y(\mathbf{n}) - r_w(\mathbf{n})\|_2^2 / \|r_w(\mathbf{n})\|_2^2$, $\mathbf{n} \in \mathcal{X}$, where $y(\mathbf{n})$ and $r_w(\mathbf{n})$ are the reconstructed and windowed original signal, respectively.

Fig. 5.5 shows the amplitude error and the overall error with respect to SNR for both on-grid and off-grid cases. For all the cases, the error decrease as SNR increases. The on-grid cases yield smaller error than that of the off-grid cases, and also tolerates lower

SNR; this is because the application of the Dolph-Chebyshev windows in the off-grid cases reduces SNR. The overall error of the former is much smaller than that of the latter. For instance, when SNR equal $14dB$, the overall error of the former is less than 0.01, which is contributed by the noise energy. On the contrary, for the latter, the overall error reaches 0.2 for $SNR = 20dB$ and $K' = 10$; this is because the Dolph-Chebyshev window of large PSR smears the peaks of significant frequencies, a significant portion of signal energy spreads across the frequency spectrum. As a result, the robust MARS-SFT only recovers signal energies that are concentrated around the main lobes of significant frequencies. This is not a problem in applications such as radar signal processing, where only the frequency locations are needed in order to estimate target parameters.



(a)

(b)

Figure 5.5: Amplitude and overall error of the robust MARS-SFT. $(n_s, n_d) = (3, 2)$. Dolph-Chebyshev windows are applied for the off-grid cases, while the rectangular window is used in the on-grid cases. (a) Amplitude error. (b) Overall error.

5.7 MARS-SFT Based DBF Radar

With the rapid developments in the advanced driver-assistance systems and self-driving vehicles, the automotive radar plays an increasingly important role in providing multidimensional information of the dynamic environment to the perception system of the vehicle. A typical DBF automotive radar uses uniform linear array as the receive array (see Fig. 1.2). Let us assume that the array has N_2 half-wavelength-spaced elements. The radar transmits an FMCW waveform with a repetition interval T_p. We also assume that there exist K' targets in the radar coverage. After de-chirping, sampling (N_0 samples are obtained within an repetition interval) and analog-to-digital conversion for both I and Q channels, the received signal within N_1 repetition intervals can be expressed in (5.1) [47], where the vectors are 3-dimensional (3-D). The 3-D frequency $\boldsymbol{\omega} = [\omega_0, \omega_1, \omega_2]^T$ relates to the target parameters as

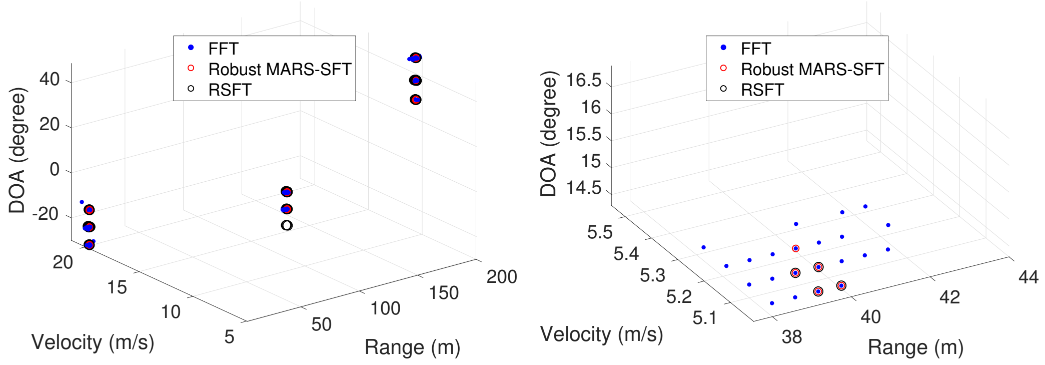
$$\begin{aligned}
&\omega_0 = 2\pi(2\rho r/c + f_d)/f_s,\\
&\omega_1 = 2\pi f_d T_p,\\
&\omega_2 = \begin{cases} \pi \sin\theta,\ \theta \in [0, 90^\circ) \\ 2\pi + \pi \sin\theta,\ \theta \in [-90^\circ, 0). \end{cases}
\end{aligned} \tag{5.6}$$

where ρ, c, f_s, f_d are chirp rate, speed of wave propagation, sampling frequency, and Doppler frequency, respectively; the chirp rate is defined as the ratio of the signal bandwidth and the repetition interval. When $K' << N$, the target parameters embedded in $\boldsymbol{\omega}$ can be estimated via the robust MARS-SFT . Conventional processing requires a multidimensional FFT, which is still computationally challenging as the increasing of the data size due to the increasing array size and also the increasing of dimensionality (e.g., the beamforming along both azimuth and elevation).

We simulate the target reconstruction for an automotive DBF radar via robust MARS-SFT, and compare against the FFT and RSFT based methods. The main radar parameters are listed in Table 5.1. Such radar configuration represents a typical long-range DBF radar [47], except that we set the number of antenna elements to be moderately large to provide a better angular resolution performance. Fig. 5.6 shows the target reconstruction of 3 radar targets via a 3-D FFT, RSFT and the robust MARS-SFT. All three algorithms are able to reconstruct all targets. Compared to the FFT and RSFT, the robust MARS-SFT requires approximately 3% of data samples, and thus exhibits low sample complexity. Also, the computation via the robust MARS-SFT is more efficient. However, we note that the robust MARS-SFT requires higher SNR than the FFT and RSFT based methods. In near range radar applications, such as automotive radar, high SNR is relatively easy to obtain.

Table 5.1: Radar Parameters

Parameter	**Symbol**	**Value**
Center frequency	f_c	$76GHz$
Pulse bandwidth	b_w	$200MHz$
Pulse repetition time	T_p	$89us$
Number of range bins	N_0	512
Number of PRI	N_1	256
Number of antenna elements	N_2	16
Maximum range	R_{max}	$300m$



(a)

(b)

Figure 5.6: Radar target reconstruction via FFT, MARS-SFT and RSFT. (a) Reconstruction of three targets. (b) Details of the frequency locations reconstructed for one of the three targets.

5.8 Summary

We have proposed the robust MARS-SFT, an extension of the SFT algorithm based on Fourier projection-slice theorem. We have shown that the robust MARS-SFT can address multidimensional data that contains off-grid frequencies and noise, while enjoys low complexity. Hence the proposed robust MARS-SFT is suitable for the low-complexity implementation of multidimensional DFT based signal processing, such as the signal processing in DBF automotive radar, provided that the scenarios are sparse.

Chapter 6

Two-Dimensional DBF Automotive Radar with Orthogonal Linear Arrays

We propose a new automotive radar architecture that achieves high resolution in range, range rate, azimuth and elevation angles, while requires a smaller antenna aperture and fewer reception channels as compared to conventional Digital Beamforming (DBF) planar arrays. This is achieved by leveraging two orthogonally-placed DBF arrays using the Frequency Modulation Continuous Waveform (FMCW). The high-resolution Range-Doppler Images (RDI) generated by azimuth and elevation beams of the two arrays isolate each physical scatterer, thus, the azimuth and elevation angles can be precisely measured. Such beamforming and RDI-generation procedure requires multidimensional Discrete Fourier Transfomr (DFT), which is ususally implemented by the Fast Fourier Transform (FFT). As the increasing of the resolution in each domain, such FFT-based method is still computational challenging. In sparse scenarios, the robust MARS-SFT proposed in Chapter 5 can be employed to replace the FFT to reduce the complexity. To match the measurements of an object from azimuth and elevation beams, a deep learning based beam matching method is proposed, which converts the beam matching problem to an image patch matching problem in the range-Doppler domain. Furthermore, a new radar resource management algorithm is proposed, which schedules radar jobs by their time urgency as well as beam locations. Jobs fall into the same beams are scheduled together to optimally use the radar time resource and also reduces the computation introduced by the beam matching procedure.

6.1 Introduction

Automotive radars play an increasingly important role in self-driving vehicle applications. As compared with other sensors, such as lidar and camera, radars are less affected by adverse weather conditions. In addition, radars provide informative 4-dimensional (4-D) measurements, i.e., range, range rate, azimuth and elevation angles, which are important in perception applications. However, while the state-of-the-art automotive radars achieve high range and range rate resolution, their angular resolution is lower as compared to lidars. This is one of the main limitations of the automotive radars [56].

The basic architecture of modern beamforming automotive radars [11] is composed of a transmit antenna, and a reception Uniform Linear Array (ULA) with digitized channels. The transmit antenna forms a broad beam pattern, which covers a large Filed-of-View (FOV), while the reception array forms multiple beams simultaneously, covering the same FOV of the transit beam pattern. The angular resolution is determined by the beamwidth of the reception array, which is fundamentally limited by the antenna aperture. The larger the aperture, the narrower the beamwidth. Also, for non-ambiguous angular measurements in a large FOV, the array elements must be closely spaced. The largest FOV that an ULA can achieve, i.e., $\pm 90^\circ$ corresponds to the half-wavelength array element-wise spacing. The higher the angular resolution and the larger the FOV, the greater the number of channels are required. For instance, to achieve 3° resolution within an FOV of $\pm 60^\circ$ requires approximately 40 channels for a $77GHz$ radar. It is even harder to have both high angular resolution in azimuth and elevation while covering a large FOV. In that case, a planar array of $N_a N_e$ channels is required, where N_a, N_e are the number of channels in azimuth and elevation, respectively; this is not realistic for the automotive radar application, where the hardware and cost are highly constrained.

One way to achieve simultaneous azimuth and elevation angle measurements and also savings in the number of channels is to leverage two linear, orthogonally placed arrays [57, 58, 59]. Compared to the planar array that uses $N_a N_e$ reception channels, the two orthogonal linear arrays only have $N_a + N_e$ channels. However, to form thin fan beams either in azimuth or in elevation for a large FOV, the number of channels is still large. To reduce the number of channels, in [58], the ESPRIT super-resolution algorithm is adopted for an orthogonal-array system which only has 4 channels in each array. However, ESPRIT requires higher SNR and is less robust to noise as compared to conventional beamforming. Moreover, [58] does not address the association problem related to the array architecture. Specifically, the measurements of the two arrays should be associated to the same scatterers to form a joint measurement. In [59], a geometric matching based association algorithm is proposed to match the measurements from azimuth and elevation beams; such method constructs two sets of 3-dimensional locations of a scatterer in the Cartesian coordinate from the range and angular measurements of the two arrays, then, the matching is determined by the Euclidean distance of the two locations. Such matching method requires that only a few scatterers existing at a same range ring, otherwise, the pairwise matching is computationally infeasible. This is not applicable for high resolution automotive radar in a dense environment. For high resolution radar, each target is represented as an extended target that is composed of a large number of scatterers. Thus, there are a large volume of scatterers existing in both azimuth and elevation beams, which makes the association challenging. Hence, reducing the number of channels and addressing the measurement association in a dense environment with low computational complexity in the orthogonal array architecture are still open problems.

In this chapter, to address the aforementioned problems, we propose a new 2-dimensional (2-D) beamforming automotive radar, which, based on a small number of reception channels achieves high resolution measurement in range, rang rate, azimuth and elevation angles in a dense environment. The proposed radar employs two orthogonally-placed, collocated ULAs with N_a and N_e channels in azimuth and elevation, respectively. Unlike the orthogonal arrays in [57] and [58] that achieve high resolution in the angular domain directly either via a large aperture or resorting to high-resolution algorithm such as ESPRIT, here, we take a two-step approach to resolve angles indirectly, which lends itself to a much smaller aperture and more robust angular measurement. The two-step approach is following. First, a coarse angular measurement is obtained by forming N_a and N_e fat beams in azimuth and elevation, respectively. Next, within each beam, a finer angular resolution is achieved by leveraging the high-resolution RDIs generated by 2-D FFT in the range and Doppler domain. In sparse scenarios, i.e., the number of scatterers is much smaller than the number of resolution cells in the range, Doppler and DOA domains, we can leverage the robust MARS-SFT to implement the DBF and RDI-generation procedure to significantly reduce the sample and computational complexity. Owning to the large-bandwidth, long-time coherent processing FMCW, each pixel of the RDIs is highly likely to represent a single scatterer from an object.

Subsequently, the azimuth and elevation angle of scatterers can be precisely measured using the monopulse technique [60], which is more robust to noise as compared with the ESPRIT algorithm applied in [58]. The high resolution RDI not only helps to resolve angles, but also leads to a novel approach of measurement association, which is described in the following.

To match the azimuth and elevation beams that cover the same objects, we propose a deep learning based approach. The main idea here is that the range-Doppler signature (pattern) of the same object at the same time, corresponding to the two beams generated by the two arrays, should be similar. Hence, the beam matching problem is converted to an image patch matching problem in the range-Doppler domain. State-of-the-art image patch matching leverages the Convolutional Neural Networks (CNN) [61]; specifically, the so-called "Siamese-net" [62] is employed to identify the similar or dissimilar image patterns. The Siamese-net is composed of a pair of identical, shared-weights CNNs. During the training, the network is fed into pairs of matched and non-matched image patches, which contain similar and dissimilar patterns, respectively. The two CNNs convert the input image patches to feature vectors; the Euclidean distance of the two feature vectors depends on the similarity of the patterns contained in the image pairs. After training, such neuron-network can generalize to unseen patterns, which is excellent for the beam matching application. Compared to the point-to-point association approach of [59] that works in a sparse scenario or within a narrow beam, the proposed pattern-based matching approach works for matching two broad beams that contain dense points.

The beam matching processing could be computationally heavy in a dense environment. In the worst case, each search frame would generate $N_a N_e P$ matching pairs, where P is the number of patches in each RDI. In order to reduce the computation and optimally utilize the radar resource, we introduce a radar resource management method for the proposed radar architecture. Radar resource management techniques are widely used in phased array radars with narrow beamwidth, whose FOV is covered by different beams in a time sharing manner. Such time-sharing scheme is scheduled by a resource manager. A popular scheduling algorithm implemented in real-life radar systems is the time-balanced algorithm [63]; such algorithm considers the urgency of each radar job and guarantees that the more urgent a job is, the higher priority it obtains. Based on the time-balanced algorithm, we propose a new scheduling algorithm, which adapts to the proposed radar architecture. Specifically, in addition to considering the urgency of jobs, the proposed algorithm also considers the beam positions of the tracking jobs; the tracking jobs which fall into the same beam are grouped and executed together; this achieves significant savings of radar time in dense environment. Moreover, since the tracking jobs are already associated with the corresponding azimuth and elevation beams, no beam matching computation is required; this enables computational savings. The advantage of the proposed radar is demonstrated by simulations.

6.2 System Model

6.2.1 Target Model

We model an extended target as a collection of independent scatterers lying on a 3-D grid of a cuboid, i.e., $\mathbf{\Omega} \triangleq \{-L/2, -L/2 + \Delta L, \cdots, L/2\} \times \{-W/2, -W/2 + \Delta W, \cdots, W/2\} \times \{-H/2, -H/2 + \Delta H, \cdots, H/2\}$, where L, W, H are the length, width and height of the target, respectively, and $\Delta L, \Delta W, \Delta H$ denotes the spacing of grid points along the length, width and height dimensions, respectively. The reflection coefficient of each scatterer is $\rho \triangleq p_s \rho_s e^{j\phi}$, where p_s, ρ_s and ϕ are random variables, which are distributed according to Bernoulli, Gaussian and uniform distributions, respectively. A target whose geometric center is located at $\mathbf{x}_t = [x_t, y_t, z_t]^T \in \mathbb{R}^3$ can be expressed as

$$T(\mathbf{x} - \mathbf{x}_t) = \sum_{\mathbf{x}_0 \in \mathbf{\Omega}} \rho(\mathbf{x}_0)\delta(\mathbf{x} - \mathbf{x}_t - \mathbf{x}_0), \tag{6.1}$$

where $\mathbf{x} = [x, y, z]^T, \mathbf{x}_0 = [x_0, y_0, z_0]^T, \mathbf{x}, \mathbf{x}_0 \in \mathbb{R}^3$; $\delta(\cdot)$ is the unit impulse function. We also assume that the velocity of the target center is $\mathbf{v} = [v_x, v_y, v_z]^T \in \mathbb{R}^3$. Assuming that the radar location is at the origin and its velocity is 0, the radial velocity of each target scatterer with respect to the radar is

$$v(\mathbf{x}_0) = \mathbf{v}^T(\mathbf{x}_t + \mathbf{x}_0)/\|\mathbf{x}_t + \mathbf{x}_0\|, \; \mathbf{x}_0 \in \mathbf{\Omega}. \tag{6.2}$$

The range, azimuth and elevation angle of each scatterer are calculated as follows

$$\begin{aligned} r(\mathbf{x}_0) &= \|\mathbf{x_t} + \mathbf{x_0}\|, \\ \phi_a(\mathbf{x}_0) &= \arctan(\frac{y_t + y_0}{x_t + x_0}), \\ \phi_e(\mathbf{x}_0) &= \arcsin\left((z_t + z_0)/r(\mathbf{x}_0)\right), \; \mathbf{x}_0 \in \mathbf{\Omega}. \end{aligned} \tag{6.3}$$

6.2.2 Radar Architecture

The proposed radar architecture is illustrated in Fig. 6.1 (a). The radar contains a transmission antenna and a pair of reception antenna arrays, which are two orthogonally placed ULAs. The two arrays are named as azimuth and elevation array, and have N_a, N_e elements, respectively. The transmit antenna forms a wide beam pattern, which covers the FOV of the radar, while the azimuth and elevation arrays form N_a, N_e beams in azimuth and elevation, respectively. Each reception channel is mixed with a coupled signal from the transmitter to de-chirp the received FMCW signal. The digitized received signal is processed by the Processing and Control Unit (PCU). The arrays of the radar are to cover a large FOV in azimuth, e.g., ±60° and a smaller FOV in elevation, e.g., 30°. The beam pattern corresponding to such design is illustrated in Fig. 6.1 (b).

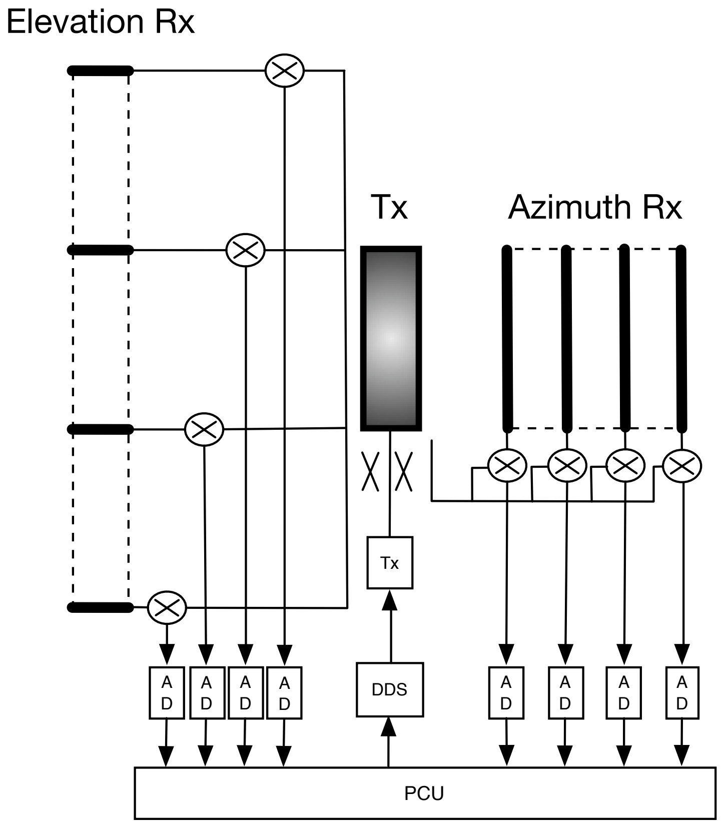
6.2.3 Signal Model

Based on the target model and the radar architecture, for a single extended target, after demodulation, the received signal for the uth, $u \in [N_a]$ azimuth channel of the pth, $p \in [P]$ pulse can be expressed as [11]

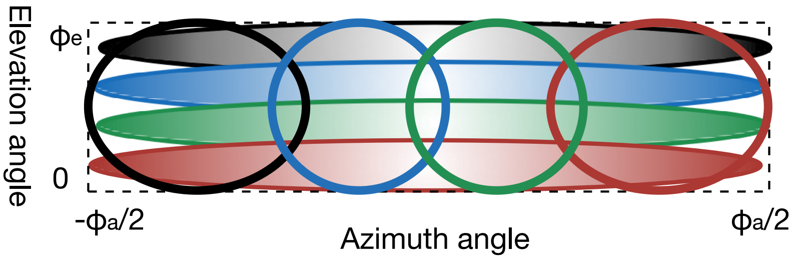
$$s_u(t) = \sum_{\mathbf{x}_0 \in \Omega} \rho(\mathbf{x}_0) e^{j2\pi(f_r(\mathbf{x}_0)t + pTf_d(\mathbf{x}_0) + id_a \sin(\phi_a)/\lambda)}, \tag{6.4}$$

where $t \in (0, T)$, and T is the pulse repetition interval. $f_r(\mathbf{x}_0)$ is the frequency related to $r(\mathbf{x}_0)$; $f_d = 2v(\mathbf{x}_0)/\lambda$ is the Doppler frequency; λ is the wavelength, and d_a is the element-wise spacing of the azimuth array.

In general, when multiple targets are present, the received signal can be expressed as the superposition of the signal generated by each target and noise. The signal received from the elevation array takes a similar form. Classic signal processing extracts target parameters, i.e., range, range rate and angle by applying DFT to the digitized signal in each dimension [11], however, the closely spaced scatterers are not resolvable in the angular domain. Also, how to associate the measurements of the two arrays is challenging for a large number of scatterers. In the next section, we propose the unique signal processing method to address those problems.



(a)



(b)

Figure 6.1: The proposed radar architecture. (a) Radar architecture. (b) Beam positions. The radar forms N_a, N_e ($N_a = N_e = 4$) beams in azimuth and elevation, respective.

6.3 Signal and Data Processing

6.3.1 Overview

The overview of the signal and data processing of the proposed radar is shown in Fig. 6.2. Upon reception of the signal from the two arrays, digital beamforming is applied to form azimuth and elevation beams as shown in Fig. 6.1 (b). Next, the RDI is computed for each beam using DFT; the beamforming and RDI-generation can be implemented by the FFT and robust MARS-SFT in dense and sparse scenarios, respectively. Subsequently, detection is applied on the RDIs generated from each azimuth beam. A successful detection triggers the beam matching procedure, which will be discussed in detail in Section 6.3.3. The matched azimuth and elevation angles are subsequently measured using monopulse. Next, the 4-D measurements of targets are forwarded to a multi-object, extended target tracker [64], which proposes and maintains tracking tasks. The resource manager is adopted to schedule searching and tracking jobs; this is discussed in detail in Section 6.3.4.

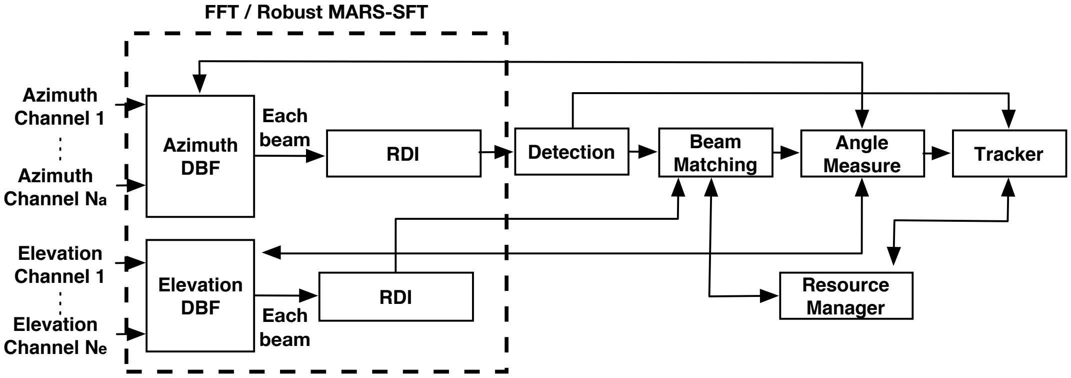


Figure 6.2: Overview of signal and data processing.

6.3.2 SFT-based Pre-processing

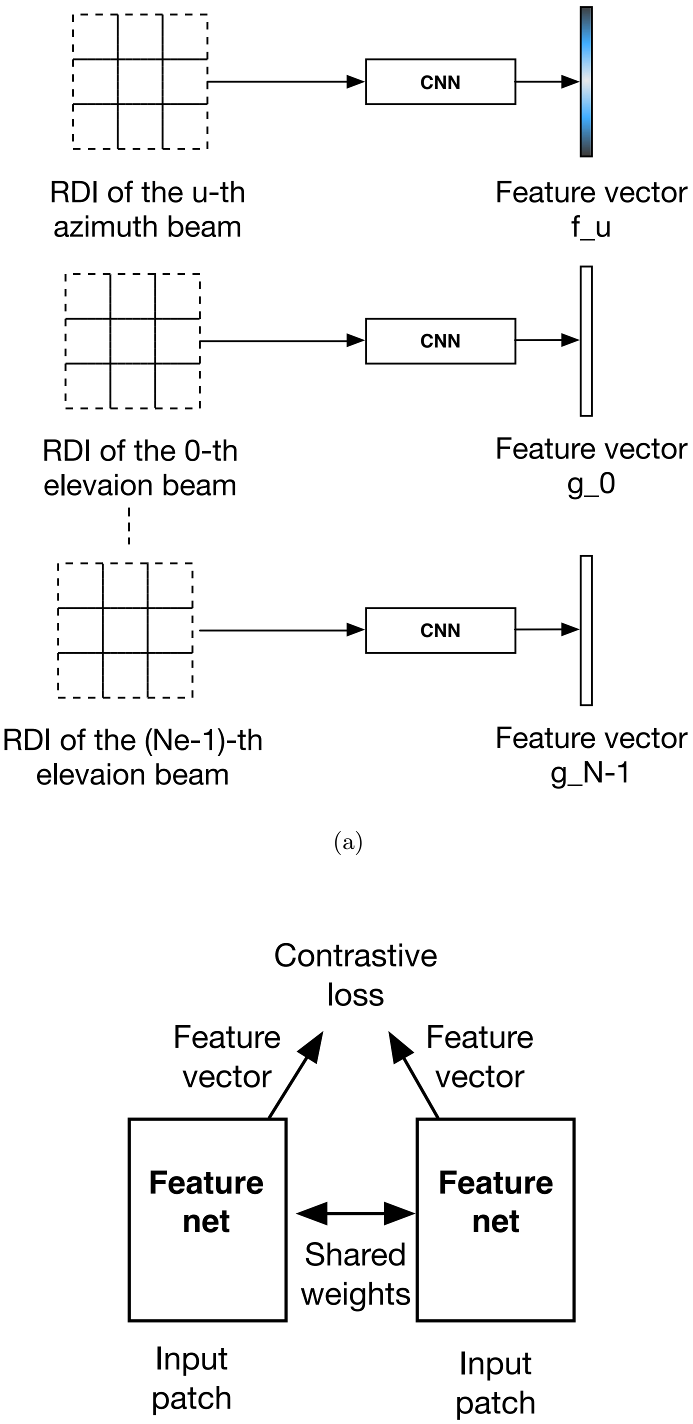
In dense scenarios, we use DFT to form azimuth and elevation beams and compute RDIs for each beam. In a sparse scenario, i.e., when the number of scatterers is much smaller than the number of resolution cells in the range, range rate, and angular domains, the robust MARS-SFT can be adopted to implement the DFT to reduce the computation. Specifically, a 3-D robust MARS-SFT is employed to simultaneously form N_a azimuth beams and compute the RDI for each beam. The output is a set of 3-D sinusoids whose frequenices represent the spatial frequency, range, and Doppler of each scatterer. The RDI images can be trivially constructed from such set. The similar processing is applied to the elevation channels.

6.3.3 Beam Matching via Deep Learning

Matching the azimuth and elevation beams associated to the same targets is not trivial especially in a dense environment. In [59], the beam matching is applied on the scatterer-basis, i.e., the association is between each range-angle pairs of measurements from the azimuth and elevation arrays. This approach is not computational feasible in scenarios where the number of measurements is large.

The same target has a similar pattern in the RDI of the corresponding azimuth and elevation beams, hence, it is possible to match the beams based on patterns rather than measurements of each scatterer. While similar, the range-Doppler patterns of a target in different beams are not exactly the same, due to different FOV of the azimuth and elevation beam, and different SNR of different channels. We take a deep learning based approach to address those challenges. Specifically, we adopt CNN to match targets from different azimuth and elevation channels based on range-Doppler patches generated from different azimuth and elevation beams. We call this as Beam Matching Net (BMN). The overall structure of the BMN is shown in Fig. 6.3 (a).

The goal of BMN is to match targets detected in any azimuth beam to those detected in the elevation beams. To achieve this, the RDI of the uth, $u \in [N_a]$ azimuth beam is divided into $p \times q$ patches; the size of each patch is $R/p \times D/q$, where $R \times D$ is the size of the RDI. Detection is applied to each patch. The detection can be implemented by comparing the energy of a patch to a predefined threshold. If targets are detected in the (i,k)th, $i \in [p], k \in [q]$ patch of the uth azimuth beam, such image patch is forwarded to a CNN to extract a feature vector denoted as $\mathbf{f}^u_{i,k}$. Meanwhile, the (i,k)th image patch of each elevation channel is forwarded to the same CNN, which outputs N_e feature vectors, i.e., $\mathbf{g}^v_{i,k}, v \in [N_e]$. Next, the Euclidean distance between $\mathbf{f}^u_{i,k}$ and $\mathbf{g}^v_{i,k}, v \in [N_e]$ is computed;



(a)

(b)

Figure 6.3: The beam matching net. (a) Beam matching net. (b) Siamese net.

the elevation channel corresponding to the minimum distance is the matched channel, i.e.,

$$v_m = \text{argmin}_{v \in [N_e]} = \|\mathbf{f}_{i,k}^u - \mathbf{g}_{i,k}^v\|. \tag{6.5}$$

The design of the CNN in BMN is based on the LeNet architecture [65], which is composed of cascade of multiple convolutional layers followed by pooling layers. Several fully connected layers are appended, which outputs a feature vector. To adapt LeNet in our application, we change the input layer as $R/p \times D/q$ and the output layer as a 64-dimensional vector.

The proposed BMN needs to identify the same and different range-Doppler patterns. In computer vision problems, such pattern matching problem is often addressed by the Siamese-net (see Fig. 6.3 (b)) [66]. The Siamese net is composed of a pair of identical weight-sharing feature extraction CNNs (feature net). The input of the feature nets are pairs of image patches, containing similar or dissimilar patterns. During the training, the contrastive loss [66] is used to enforce the feature net to output similar features for identical patterns and different features for different patterns. After training, the Siamese net can generalize to unseen patterns.

6.3.4 Resource Management

The beam matching processing could be computational heavy in a dense environment. In the worst case, each frame covering the whole FOV would generate $N_a N_e P$ matching pairs, where P is number of patches in each RDI. In order to reduce the computation of the proposed radar architecture and optimally utilize the radar resource, we introduce the radar resource management to optimally schedule radar jobs in the processing.

The proposed scheduling algorithm shown in Fig. 6.4 is an extension of the time-balanced scheduling algorithm, which is adapted to the proposed radar architecture. Time-balanced scheduling algorithm prioritizes the radar jobs based on their urgency, i.e., the more urgent a job, the higher priority it obtains. The urgency of a radar job is characterized by the time-balance, i.e., $t_b, t_b \in \mathbb{Z}$. When a job is proposed, such job is not time-ready, and t_b is set to be a negative value; as the time elapses, t_b increases. If $t_b \geq 0$, such job is time-ready and requires to execute as soon as possible. Here, in addition to considering the urgency of jobs, the track locations are also taken into account. Specifically, the tracks that fall into the same azimuth and elevation beams are grouped together; those tracks can be updated by a single tracking job, which saves radar time as compared to updating each track individually. Moreover, since tracking jobs are already associated to specific azimuth and elevation beams, no beam matching procedure is required; this results into a significant computational savings. To save radar time and computation, it is preferred to group as many tracks as possible into the smallest number of beams. Thus, the scheduler would prefer to delay the scheduling of the time-ready tasks and gather as many time-ready tasks as possible. However, the delay of scheduling of tracking tasks would cause a lower updating rate. To trade off the savings and the update time delay of tracking jobs, we propose the following cost function

$$f = \alpha \sum_{i \in [J]} t_i + B/J, \tag{6.6}$$

where $\alpha \in (0, 1)$ is the weight of the cost introduced by the time balance of J tracking jobs; the time balance of the ith job is denoted as t_i. The cost of the grouping is denoted by B/J, where B is the number of groups (beams). In the best case, all the tracking jobs can be grouped into a single group; the cost of the grouping in such scenario is $1/J$, while in the worst case, all tracking jobs fall into individual non-overlapping beams, and the cost of grouping is 1.

The resource management algorithm shown in Fig. 6.4 is executed at a fixed rate (fixed schedule interval); such rate is the same as the radar frame update rate, e.g. $20Hz$. For each iteration, the procedure is summarized as follows:

1. The t_b of each tracking task is increased by 1.

2. If there exists tracking tasks whose t_b is greater or equal to 0, then the resource manager asks the tracker to predict the jobs' beam locations at the execution time. Otherwise, the resource manager generates a search job.

3. Group all the tracking jobs that are requesting to execute into B beams; each beam is an intersection of a azimuth beam and a elevation beam.

4. Calculate the cost function based on (6.6). If the cost is greater than the threshold C, then schedule those tracking jobs. Otherwise, generate a search job and then enter

the next iteration.

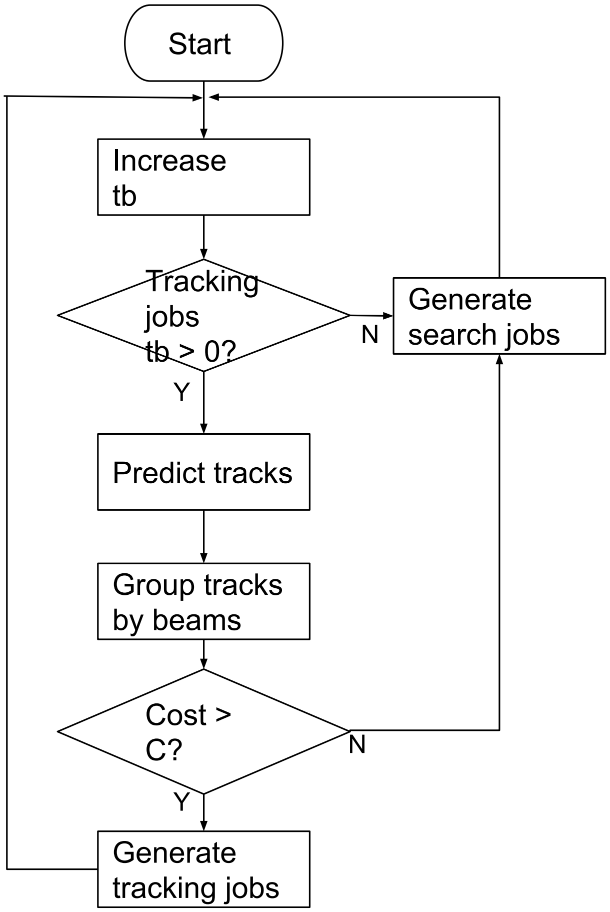


Figure 6.4: The proposed resource management algorithm.

6.4 Simulation

In this section, we verify the proposed radar architecture via simulations. The parameters of the simulated radar are show in Table 6.1. The FOVs corresponding to the azimuth and elevation arrays are ±60° and ±15°, respectively.

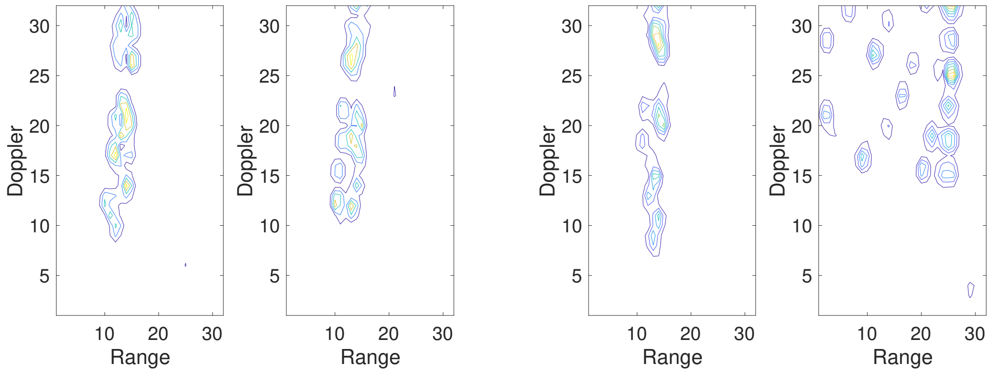
6.4.1 RDI Patch Matching

We generate RDI patches corresponding to targets of various shape, position and velocity at different SNR. The size of each patch is 32×32. Those patches are grouped into pairs of

Table 6.1: Radar Parameters

Parameter	Symbol	Value
Center frequency	f_c	$76GHz$
Pulse bandwidth	b_w	$1GHz$
Pulse repetition time	T	$90us$
Number of range bins	N_r	512
Number of PRI	P	256
Maximum range	R_{max}	$100m$
Number of azimuth antenna elements	N_a	4
Number of elevation antenna elements	N_e	4
Element-wise spacing of azimuth array	d_a	0.58λ
Element-wise spacing of elevation array	d_e	1.93λ

similar and dissimilar patterns. A pair of similar patterns corresponds to the same target of different SNR. A training sample contains a pair of patches and a label; the label is 0 and 1 for the pairs of similar and dissimilar patterns, respectively. Examples of similar and dissimilar pairs are shown in Fig. 6.5.

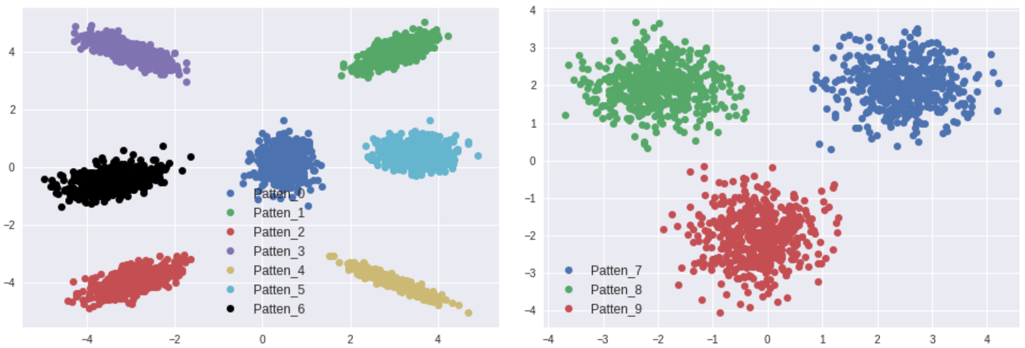


(a)

(b)

Figure 6.5: Examples of similar and dissimilar range-Doppler patches. (a) Similar patch. (b) Dissimilar patch.

We generated 5000 pairs of training samples and divided them into a training set and an evaluation set, which contains 4000 and 1000 samples, respectively. The training set contains 7 different patterns, while the testing set contains 3 different patterns which do not appear in the training set. We then use the TensorFlow to train and test the model. After training, we extract the 64-dimensional feature vectors from each RDI patch in the training and the test set and project them into 2-D vectors for visualization. Fig. 6.6 shows the visualization of the clustering of the 2-D feature vectors in the training and testing set. One can see that the feature vectors of the 7 different patterns are clustered, hence the Euclidean distance within a same pattern is small. Moreover, even though the patterns in the testing set do not appear in the training set, they are still clustered in the feature space, which shows the generalization of the CNN. Such simulation is for the proof of concept. To obtain a better performance in practical systems, a much larger training set containing much more different patterns is required.

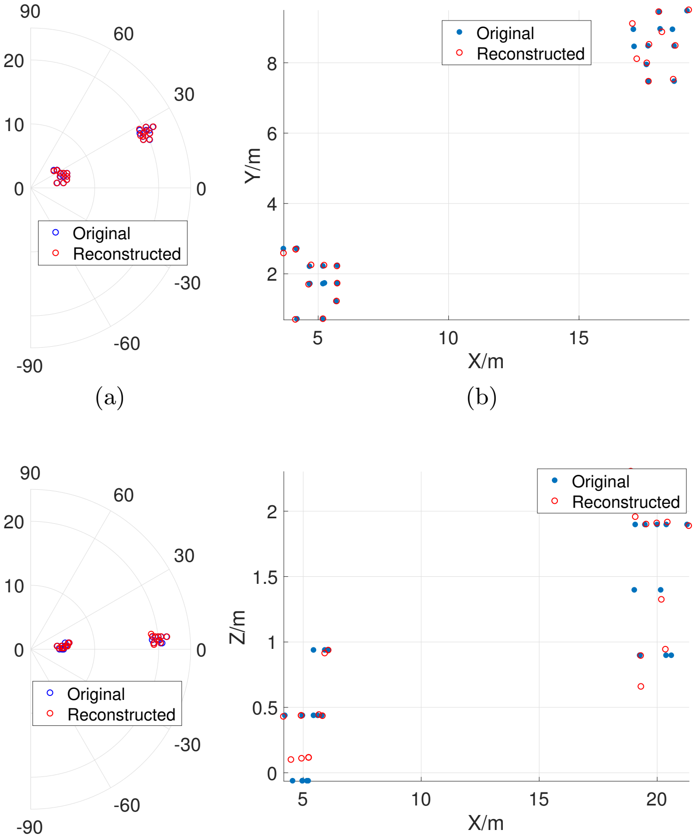


(a) (b)

Figure 6.6: Clustering of feature vectors from the train and test set. (a) Train. (b) Test.

6.4.2 Target Reconstruction

We simulate two targets whose locations in the spherical coordinate, (r, ϕ_a, ϕ_e), are $(5m, 20°, 5°)$, $(20m, 25°, 4°)$, respectively, while their velocities (v_x, v_y, v_z) are $(30, 5, 1)m/s$, $(15, 5, 1)m/s$, respectively. The size of each target is $(L, W, H) = (2, 2, 1)m$ and $\Delta L = \Delta W = \Delta H = 0.5m$. The azimuth-range and elevation-range measurements after beam matching are shown in Figs. 6.7 (a), (c), and the positions for each scatterers in the Cartesian coordinate are shown in Figs. (b), (d). Compared with that of the elevation array, the reconstructed positions in the Cartesian coordinate from the azimuth array is closer to the ground truth. This is due to that the Doppler spreading for each scatterer is more prominent in azimuth than in elevation; as a result, the azimuth angel measurement of each scatterer is more precise than that of the elevation angle. The 3-D reconstruction is shown in Fig. 6.8.



(a) (b)

(c) (d)

Figure 6.7: Measurement and reconstruction from individual array. (a) Azimuth-range measurement from the azimuth array. (b) Reconstruction in X-Y. (c) Elevation-range measurement from the elevation array. (d) Reconstruction in X-Z.

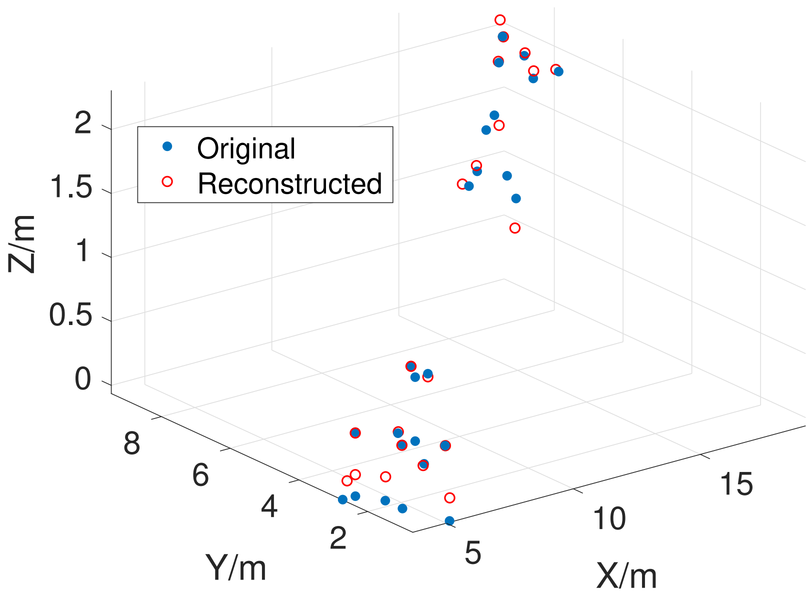
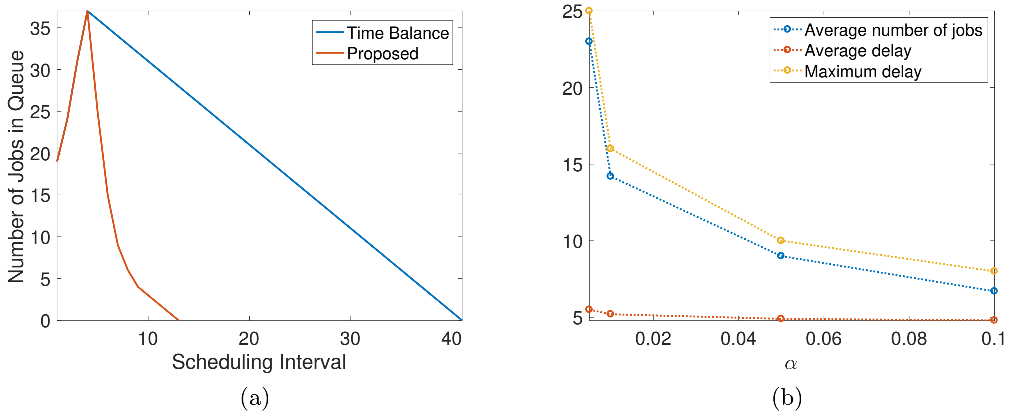


Figure 6.8: Reconstruction in the X-Y-Z coordinate.

6.4.3 Resource Management

In a dense environment, such as driving in urban areas, each search job of an automotive radar can create many tracking jobs. With the conventional time-balanced job scheduling, those jobs are stored in a queue, and been scheduled one by one according to their priorities (urgency). This means that it may take a long time to clear the queue even without any new tasks come in. On the other hand, in the proposed scheduling algorithm, since we combine the jobs that can fall in the same beam, the queue can be cleared much faster, i.e, compared to the time balance algorithm, more tracking jobs can be scheduled by the proposed algorithm in the same time frame. Fig. 6.9 (a) shows that with 37 tracking jobs in queue, the proposed algorithm can finish scheduling these tracking jobs in 9 intervals compared to the 37 intervals required by the conventional time balance algorithm. Fig. 6.9 (b) shows the performance of the proposed algorithm with respect to various values of α. The smaller the α, the smaller the penalty applied to the time-balance part of the cost function (see (6.6)), which results into more jobs than can be grouped into the same beams; this involves savings of radar time and computational at the expense of longer delays of updating of tracking jobs.



(a)

(b)

Figure 6.9: Comparison of resource management algorithms. (a) Comparison of the proposed algorithm and the time-balanced algorithm. (b) Effect of the cost function. The unit of average and maximum delay is number of schedule intervals. $C = 3$.

6.5 Summary

We have proposed a new automotive radar for perception applications in self-driving scenarios. Such radar achieves high resolution measurements in range, range rate, azimuth and elevation angles of extended targets by leveraging two orthogonally-placed digital beamforming linear arrays of a few channels. The deep learning based beam matching method and a resource management algorithm have been developed for the proposed radar architecture to address the beam association and the related computational challenges. In sparse scenarios, we have leveraged the robust MARS-SFT to reduce the sample and computational complexity in beamforming and range-Doppler image computation.

Chapter 7

Conclusions and Future Research Directions

7.1 Conclusions

The dissertation has formulated SFT-based signal processing frameworks for automotive digital beamforming (DBF) and MIMO radars that archive substantial reduction in sample and computational complexity in sparse scenarios, i.e., when the number of targets are significantly less than the number of resolution cells in the range, Doppler, and direction of arrival (DOA) domains.

Compared to the FFT, existing state-of-the-art SFT algorithms have achieved significant savings in sample and computation for signals that are sparse in the frequency domain. However, those SFT algorithms are usually one-dimensional, and suffer from noise and off-grid frequencies arising in real-world radar signals. To address those problems, we have proposed multidimensional SFT algorithms that are robust to noise and off-grid frequencies.

First, we have proposed the RSFT algorithm. RSFT employs a pre-permutation window and Neyman-Pearson (NP) detection to address the off-grid frequency and frequency detection problems arising in the application of SFT in real-world situations. We have shown that the RSFT is robust in detecting frequencies when exact knowledge of signal sparsity is not available. The optimal design of parameters in RSFT have been analyzed, and the tradeoff between detection performance and computational complexity has been investigated. Such analysis has revealed that RSFT could provide an extra degree of freedom of design in trading off the system's ability to detect weak signals and computational complexity.

We have applied the proposed RSFT to DBF radar that employs Frequency Modulation Continuous Waveform (FMCW), and MIMO radars using pulse-compression waveforms. The architecture of the radar processing has been designed to adapt the application of the proposed RSFT algorithm. For MIMO radars that employ pulse-compression waveforms, the signal is not naturally sparse in the range domain, we thus have investigated the preprocessing methods to sparsify the signal, so that the RSFT can be applied to the subsequent processing. The RSFT-based processing has achieved substantial computational reduction as compared to the FFT-based approach, however, the sample complexity of the former is the same as the latter.

Next, to reduce both sample and computational complexity of SFT, we have proposed MARS-SFT, a multidimensional sparse Fourier transform that is inspired by the Fourier projection-slice theorem. We have shown that MARS-SFT could handle less sparse and even clustered frequencies, while enjoys low sample and computational complexity. The sample and computational complexity of MARS-SFT achieves the lowest complexity among all know SFT algorithms when the sparsity is of the same order of the slice length. The relationship between MARS-SFT and the Fourier projection-slice theorem has been discussed. Especially, the connections between the one-projection theorems under the context of MARS-SFT and the Fourier projection-slice theorem has been exploited; the classic one-projection theorem and the one-projection theorem of MARS-SFT establish an unambiguous one-to-one mapping from a 2-D sequence to a 1-D sequence.

MARS-SFT is designed for ideal signals that is noiseless and contain only on-grid (digital) frequenceis. To handle real-world radar signals, we have proposed the robust MARS-SFT, which has employed a windowing step and a voting-based frequency decoding step to adress the frequency leakage of off-grid frequencies, and reduce the frequency localization error stemming from the noise. We have shown that the robust MARS-SFT can address multidimensional real-world signal, while enjoys low complexity. Hence the proposed robust MARS-SFT is suitable for the mutidimensional signal processing in DBF automotive radars that uses FMCW. We thus have demonstrated the application of the robust MARS-SFT in such radar via simulation.

Finally, we have proposed a new automotive radar for perception applications in self-driving scenarios. Such radar achieves high resolution measurements in range, range rate, azimuth and elevation angular domains of extended targets (each target is modeled by a set of scatterers) by leveraging two orthogonally-placed digital beamforming linear arrays of a few channels. The deep learning based beam matching method and a resource management algorithm have been developed for the proposed radar architecture to address the beam association and the related computational challenges. In sparse scenarios, the proposed robust MARS-SFT has been leveraged in the beamforming, and range-Doppler imaging procedures to reduce the complexity of the proposed architecture.

7.2 Future Research Directions

7.2.1 Clutter and Interference Suppression for SFT-based Radar Processing

One of the basic assumptions of the proposed SFT-based radar signal processing is that the scenarios are sparse, i.e., the number of targets is significantly smaller than the number of resolution cells in the range, Doppler, and DOA domain. In many real-life radar applications, clutters generated from stationary background objects (such as ground, buildings and vegetation), and interference produced from other radiation sources of the similar frequency band could occupy a substantial amount of resolution cells; this undermines the sparsity assumption of SFT-based algorithms. In Chapter 3, we apply proper pre-processing methods in order to sparsify signals that are not naturally sparse in the range domain for MIMO radars. It is also important and interesting to investigate low-complexity pre-processing methods for clutter and interference suppression that would sparsify radar signals, and allow SFT-based algorithms in the subsequent processing.

7.2.2 Deterministic SFT-based Method for Radar Signal Processing

The proposed RSFT and robust MARS-SFT based radar signal processing methods are all based on some randomized approach to shuffle the frequencies to different buckets for each iteration. Such randomization is necessary to avoid frequency collisions caused by projecting high dimensional frequencies to a lower dimension. Due to such randomization, those algorithms are probabilistic, whose successful rate upper-bound is a function of the sparsity level K. There is another branch of SFT algorithms that are deterministic [67, 35, 68]. For failure intolerant and mission-critical radar applications, those deterministic algorithms with zero probability of failure are highly desirable. However, the deterministic

SFT algorithms usually have much stronger assumptions, and suffer a greater complexity than randomized SFT algorithms. For instance, the deterministic SFT algorithm of [35] assumes that we know K, and each sinusoid has different amplitude, so that the algorithm can match frequencies calculated from a series of short DFTs of pre-designed length; such assumption does not hold in real-life radar signal processing. In our future research, we'd like to investigate the relaxation of the assumptions of the aforementioned deterministic SFT methods as well as reduction of their complexity to allow their applications in radar signal processing.

7.2.3 Characterization of Detection Performance for K frequencies in RSFT

In Chapter 2, we use P_{fa}, P_d to characterize the false alarm rate at each of N frequency bins and the detection performance for the weakest frequency in the signal of N samples for RSFT, respectively. However, setting the false alarm rate threshold for each individual frequency does not guarantee anything for the discovery probability of multiple targets. Additional work is required to characterize the detection performance of RSFT for K significant frequencies and the associated P_{fa} by considering all the N tests. As such, in the future work, we would like to develop theoretical results by investigating metrics such as family-wise error rate (FWER) [69] and false discovery rate (FDR) [70]. FWER and FDR are metrics for false alarms usually employed in multiple comparison problems [69], arising when one considers a set of N hypothesis testings, simultaneously. Since we also consider N tests in RSFT, those metrics could help us to better quantify the detection performance of RSFT.

Appendices

A Proof of Lemma 1

Proof. If we were applying N-point DFT to $\mathbf{v}(\omega)$, the highest amplitude of the spectrum would appear on the grid point closest to ω, i.e. $[\lfloor\frac{\omega}{\Delta\omega_N}\rceil]_N$. The pre-permutation windowing will not change the position of the highest peak, provided the window is symmetric. Then after permutation, the peak location dilates by σ_s modularly, and becomes $[\sigma_s[\lfloor\frac{\omega}{\Delta\omega_N}\rceil]_N]_N$. Finally, after flat-windowing and aliasing, the signal is ideally down-sampled in the frequency domain, and the data length changes from N to B. Then the B-point DFT exhibits the highest peak on grid point p as desired. □

B Proof of Property 3

Proof. According to Definition 3, the mapping can be split into two stages: 1) apply modular multiplication to i, i.e., $k = [\sigma i]_N \in [N]$; and, 2) convert k into $j \in [B]$ with $j = [\lfloor kB/N\rceil]_B$.

Similarly, according to Definition 4, the reverse-mapping also can be split into two stages: 1) dilate $j \in [B]$ into $\mathbb{S} \equiv \{v \in [N] \mid j\frac{N}{B} \leq v < (j+1)\frac{N}{B}]\} \subset [N]$; and, 2) apply inverse modular multiplication on $\mathbb{S}$, i.e., $\mathcal{R}(j, \sigma^{-1}) \equiv \{[u\sigma^{-1}]_N \mid u \in \mathbb{S}\}$.

The first stage of reverse-mapping is the inverse operation of the second stage of mapping, and as a result, $k \in \mathbb{S}$. Hence $i = [k\sigma^{-1}]_N \in \mathcal{R}(j, \sigma^{-1})$ as desired. □

C Proof of Property 4

Proof. We use the two stages of the reverse-mapping in the proof of Property 3. The first stage of the reverse-mapping for i and j yields $\mathbb{S}_1 \equiv \{v \in [N] \mid i\frac{N}{B} \leq v < (i+1)\frac{N}{B}]\}$ and $\mathbb{S}_2 \equiv \{v \in [N] \mid j\frac{N}{B} \leq v < (j+1)\frac{N}{B}]\}$, respectively. It is not difficult to verify that $\mathbb{S}_1 \cap \mathbb{S}_2 = \emptyset$, provided that $i \neq j$.

In what follows, we will prove that the second stage of the reverse-mapping also gives distinct results. Assume that there exists $m \in \mathbb{S}_1, n \in \mathbb{S}_2$, such that $[m\sigma^{-1}]_N = [n\sigma^{-1}]_N$. Modularly multiply both sides with σ yields that $m = n$, which is contradictory with $\mathbb{S}_1 \cap \mathbb{S}_2 = \emptyset$. Hence both stages of the reverse-mapping guarantee the results are distinct for $i \neq j$. □

D Proof of Lemma 2

Proof. Since $0 < \tilde{P}_d(\omega_m, \sigma_s)(1 - \tilde{P}_d(\omega_m, \sigma_s)) < 1$, for $\delta > 0$, the Lyapunov Condition[71] holds, i.e.,

$$\begin{aligned} &\lim_{T\to\infty} \frac{1}{\sigma_{a1}(\omega_m)^{2+\delta}} \sum_{s\in[T]} \mathbb{E}\{|[\mathbf{c}_{\sigma_s}]_j(\omega_m) - \tilde{P}_d(\omega_m, \sigma_s)|^{2+\delta}\} \\ &\leq \lim_{T\to\infty} \frac{1}{\sigma_{a1}(\omega_m)^{2+\delta}} \sum_{s\in[T]} \mathbb{E}\{|[\mathbf{c}_{\sigma_s}]_j(\omega_m) - \tilde{P}_d(\omega_m, \sigma_s)|^{2}\} \\ &= \lim_{T\to\infty} \frac{1}{\sigma_{a1}(\omega_m)^{\delta}} = 0. \end{aligned} \tag{1}$$

Therefore, according to the Lyapunov Central Limit Theory (CLT), as T goes to infinity, $\frac{1}{\sigma_{a1}(\omega_m)} \sum_{s\in[T]} \left([\mathbf{a}_{\sigma_s}]_i - \tilde{P}_d(\omega_m, \sigma_s)\right)$ converges to the standard Normal distribution. Equivalently, $[\bar{\mathbf{a}}]_i$ conforms to the Normal distribution as indicated in (2.25). It also holds that

$$\begin{aligned} \sigma_{a1}^2(\omega_m) &= \sum_{s\in[T]} \tilde{P}_d(\omega_m, \sigma_s)(1 - \tilde{P}_d(\omega_m, \sigma_s)) \\ &= T\bar{P}_d(\omega_m)(1 - \bar{P}_d(\omega_m)) \\ &- \sum_{s\in[T]} (\tilde{P}_d(\omega_m, \sigma_s) - \bar{P}_d(\omega_m))^2, \end{aligned} \tag{2}$$

from which we get that $\sigma_{a1}^2(\omega_m) \leq T\bar{P}_d(\omega_m)(1 - \bar{P}_d(\omega_m))$, with the equality holding when $\tilde{P}_d(\omega_m, \sigma_s) = \bar{P}_d(\omega_m)$. □

E Proof of Lemma 3

Proof. Under $\overline{\mathcal{H}}0$, each term in (2.22) may be distributed differently. To illustrate this, we consider a location $i \in [N]$ in the frequency domain of the input signal, which does not contain a significant frequency, as shown in Fig. 2.4. Let $j = \mathcal{M}(i, \sigma_s)$ be the mapping. There would be two cases for bin j: 1) bin j does not contain a significant frequency; or 2) bin j contains at least one significant frequency, with its SNR at least SNR_{min}. In the former case, $[\mathbf{c}_{\sigma_s}]_j \sim$ Bernoulli $\left(\tilde{P}_{fa}(\sigma_s)\right)$, i.e., $[\mathbf{c}_{\sigma_s}]_j$ is under $\mathcal{H}0$. For the latter case, $[\mathbf{c}_{\sigma_s}]_j \sim$ Bernoulli $\left(\tilde{P}_d(\omega_m, \sigma_s)\right)$, i.e., $[\mathbf{c}_{\sigma_s}]_j$ is under $\mathcal{H}1$. Due to the permutation being uniformly random, on the average, the number of $[\mathbf{c}]_j$ under $\mathcal{H}1$ is $F = \frac{TK\eta_m}{B}$, and the number of $[\mathbf{c}]_j$ under $\mathcal{H}0$ is $T - F$. The parameter η_m reflects the fact that sparsity is affected by the pre-permutation windowing. Since we assume that $\mathbf{v}(\omega_m)$ has the minimum

SNR, i.e., SNR_{min}, other sinusoids with higher SNR will have larger $\bar{P}_d$. Hence we multiply $\bar{P}_d(\omega_m)$ with η_p to calibrate the successful rate of $[\mathbf{c}_{\sigma_s}]_j$ under $\mathcal{H}1$. If all the sinusoids's SNR were equal to SNR_{min}, then $\eta_p = 1$; on the other hand, if the co-existing sinusoids' SNR were sufficient high so that their $\bar{P}_d$ approaches to 1, then $\eta_p = \frac{1}{\bar{P}_d(\omega_m)}$. Finally, the results follows immediately by applying Lyapunov CLT. □

F Proof of Remark 2

Proof. Assuming $\eta_p = 1$ and substituting $K\eta_m = B$ into F yields $F = T$, which means that the distributions under both hypothes are the same, hence the two hypothesis cannot be discriminated. If $\eta_p > 1$, the assumption of $P_d \geq P_{fa}$ will be violated as $K\eta_m$ approaching B. □

G Proof of Lemma 4

Proof. Assume $\mathbf{r} = \mathbf{v}(\omega_m)$. Since the pre-permutation window $\mathbf{w}$ is symmetric, if we applied the DFT to the pre-permuted data, the amplitude of the spectrum would attain its maximum and minimum respectively when ω_m is on-grid or in the middle between two grid points. The subsequent permutation operation would not change the amplitude of the spectrum. Also, since the flat-window is used, the down-sampling in the frequency domain, which is a result of aliasing, will not affect the amplitude either. The on-grid frequency generates highest amplitude, while the frequency in the middle of between grid points has the lowest amplitude. As a result, the detection procedure would yield the highest P_d^* for on-grid frequencies, and the lowest P_d^* for frequencies laying in the middle of between grid points. □

H Proof of Lemma 5

This proof is organized as follows. First, by exploring Bézout's lemma [72], we prove that for the specified line parameters, i.e.,

$$L = \mathrm{LCM}(N_0, N_1),\ [\alpha_0, \alpha_1]^T \in \mathcal{A},\ [\tau_0, \tau_1]^T \in \mathcal{X}, \tag{3}$$

each entry of (4.9) contains at least the projection of the DFT coefficient from one frequency location (m_0', m_1') in $\mathcal{X}$, i.e., $|\mathcal{P}_m| > 0, m \in [L]$. Next, we prove that $|\mathcal{P}_m| \geq N/L$, followed by the proof of $\mathcal{P}_m \cap \mathcal{P}_{m'} = \emptyset$ for $m \neq m', m, m' \in [L]$, and finally, we conclude that $|\mathcal{P}_m| = N/L$.

Let $\alpha_0' = \alpha_0 L/N_0$, $\alpha_1' = \alpha_1 L/N_1$. Since (α_0, α_1), $(\alpha_0, L/N_1)$, $(\alpha_1, L/N_0)$, and $(L/N_0, L/N_1)$ are co-prime pairs due to $L = \text{LCM}(N_0, N_1)$, it is obvious that α_0', α_1' are also co-prime. According to Bézout's lemma, there exist $m_0, m_1 \in \mathbb{Z}$, such that

$$\alpha_0' m_0 + \alpha_1' m_1 = 1. \tag{4}$$

By multiplying by $m \in [L]$ the two sides of (4), we get $\alpha_0' m m_0 + \alpha_1' m m_1 = m$, which, using the Euclidean division, can be written as

$$\alpha_0'(m_0' + k_0 N_0) + \alpha_1'(m_1' + k_1 N_1) = m, \tag{5}$$

where $m_0' = [m m_0]_{N_0}, m_1' = [m m_1]_{N_1}$ and $k_0, k_1 \in \mathbb{Z}$.

Since

$$[\alpha_0' k_0 N_0 + \alpha_1' k_1 N_1]_L = [L(\alpha_0 k_0 + \alpha_1 k_1)]_L = 0, \tag{6}$$

on taking modulo-L of the two sides of Eq. (5), we have that

$$[\alpha_0' m_0' + \alpha_1' m_1']_L = m, \tag{7}$$

which is equivalent to (4.7). This means that there exists a frequency location $[m_0', m_1']^T \in \mathcal{X}$, whose DFT coefficient projects to $\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m)$, i.e., $|\mathcal{P}_m| > 0, m \in [L]$.

Next, let us explore the structure of the solution of (7). It is easy to see that the frequency locations, $[m_0' + k\alpha_1', m_1' - k\alpha_0']^T$, $k \in \mathbb{Z}$, satisfy (7), i.e.,

$$[\alpha_0'(m_0' + k\alpha_1') + \alpha_1'(m_1' - k\alpha_0')]_L = m, \tag{8}$$

which can be written as

$$[\alpha_0'([m_0' + k\alpha_1']_{N_0} + k_0 N_0) + \alpha_1'([m_1' - k\alpha_0']_{N_1} + k_1 N_1)]_L = m, \tag{9}$$

where $k_0, k_1 \in \mathbb{Z}$. Again, by employing (6), we have

$$[\alpha_0'[m_0' + k\alpha_1']_{N_0} + \alpha_1'[m_1' - k\alpha_0']_{N_1}]_L = m. \tag{10}$$

Hence, the DFT coefficients at frequency locations $[[m_0' + k\alpha_1']_{N_0}, [m_1' - k\alpha_0']_{N_1}]^T \in \mathcal{P}_m \subseteq \mathcal{X}$, also project to $\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m)$; those frequencies are located along the line with slope $-\frac{\alpha_0 N_1}{\alpha_1 N_0}$ and offset $[m_0', m_1']^T$; such frequency-domain line is orthogonal to the time-domain line defined in (4.2).

Next, we prove that $|\mathcal{P}_m| \geq N/L$. Assume that for $k \neq k'$, there exits two duplicated frequency locations, i.e., $[[m_0' + k\alpha_1']_{N_0}, [m_1' - k\alpha_0']_{N_1}]^T = [[m_0' + k'\alpha_1']_{N_0}, [m_1' - k'\alpha_0']_{N_1}]^T$.

It follows that

$$[k\alpha_1']_{N_0} = [k'\alpha_1']_{N_0},\ [k\alpha_0']_{N_1} = [k'\alpha_0']_{N_1}, \tag{11}$$

which can be rewritten as

$$k\alpha_1' = k'\alpha_1' + k_0 N_0,\ k\alpha_0' = k'\alpha_0' + k_1 N_1, \tag{12}$$

where $k_0, k_1 \in \mathbb{Z}$. It is easy to conclude that $k_1/k_0 = \alpha_0/\alpha_1$. Hence we have

$$k\alpha_1' = k'\alpha_1' + i\alpha_1 N_0,\ k\alpha_0' = k'\alpha_0' + i\alpha_0 N_1, \tag{13}$$

where $i \in \mathbb{Z}, i \neq 0$. It holds that

$$k - k' = iN/L, \tag{14}$$

which means that the frequency location, $[[m_0' + k\alpha_1']_{N_0}, [m_1' - k\alpha_0']_{N_1}]^T$, repeats every N/L points. Hence, there exist at least N/L frequency locations whose DFT values are projected to $\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m)$, i.e.,

$$|\mathcal{P}_m| \geq N/L. \tag{15}$$

Next, we prove that $\mathcal{P}_m \cap \mathcal{P}_{m'} = \emptyset$ for $m \neq m', m, m' \in [L]$. On assuming that $[m_0, m_1]^T \in \mathcal{P}_m \cap \mathcal{P}_{m'}$, it can be seen that

$$[\alpha_0' m_0 + \alpha_1' m_1]_L = m = m', \tag{16}$$

which is contradict with $m \neq m'$. Hence $\mathcal{P}_m \cap \mathcal{P}_{m'} = \emptyset$.

Finally, by combing $\mathcal{P}_m \cap \mathcal{P}_{m'} = \emptyset$, $m \in [L]$, $|\mathcal{P}_m| \geq N/L$ and $|\mathcal{X}| = N$, we conclude that $|\mathcal{P}_m| = N/L$. This completes the proof.

I Proof of Theorem 2

Since N_0, N_1 are co-prime, $L = \text{LCM}(N_0, N_1) = N_0 N_1 = N$. According to Lemma 5, each entry of the L-point DFT contains exactly one distinct sample of $\hat{x}(\mathbf{m}), \mathbf{m} \in \mathcal{X}$. Hence, $\hat{x}(\mathbf{m})$ can be recovered by only one iteration of MARS-SFT.

J Proof of Theorem 3

For the i-th, $0 \leq i \leq T$, iteration of MARS-SFT, the probability of any entry of $\hat{\mathbf{x}}_r(\mathbf{m}), \mathbf{m} \in \mathcal{X}$ containing a significant frequency is K_i/N, where $\hat{\mathbf{x}}_r(\mathbf{m}), \mathbf{m} \in \mathcal{X}$ is the $N_0 \times N_1$-point DFT of (4.1) after removal of the contribution of the recovered frequencies in the previous i iterations; K_i is the average number of frequencies in $\hat{\mathbf{x}}_r(\mathbf{m}), \mathbf{m} \in \mathcal{X}$. According to

Lemma 5, each entry of the L-point DFT along the line, i.e., $\hat{\mathbf{s}}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m), m \in [L]$ contains projections of N/L distinct entries from $\hat{\mathbf{x}}_r(\mathbf{m}), \mathbf{m} \in \mathcal{X}$, where $L = \mathrm{LCM}(N_0, N_1)$. Since the significant frequencies are assumed to be randomly distributed, if the m-th entry of $\hat{\mathbf{s}}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m), m \in [L]$ is significant, i.e., $|\hat{\mathbf{s}}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m)| > 0$, then the probability of such entry being 1-sparse is $Q_i = (1 - K_i/N)^M$, with $M = N/L - 1$. Hence, due to the uniform distribution of frequencies, the average number of significant frequencies being projected into 1-sparse bin, and hence recovered in the i-th iteration is $M_i = K_i Q_i$ with $K_0 = K$.

K_i is the average number of the significant frequencies that 'survived' in the previous i iterations. The probability of non-recovery (surviving rate) for the i-th iteration is $(1 - Q_i)$. Hence $K_i = K \prod_{k \in [i]} (1 - Q_k)$.

The algorithm stops at the T-th iteration when all the K significant frequencies are recovered, i.e., $\sum_{i \in [T]} M_i \geq K$. This completes the proof.

K Proof of Lemma 6

After windowing, the maximum absolute amplitude of the strongest frequency in the $N_0 \times N_1$-DFT domain becomes

$$|\hat{a}_w| = \frac{a_{max}}{N} \sum_{\mathbf{n} \in \mathcal{X}} w(\mathbf{n}) = \frac{\|\mathbf{W}\|_1}{N} a_{max}. \tag{17}$$

The noise in the DFT domain becomes

$$\hat{n}_w(\mathbf{m}) = \frac{1}{N} \sum_{\mathbf{n} \in \mathcal{X}} w(\mathbf{n}) n(\mathbf{n}),\ \mathbf{m} \triangleq [m_0, m_1]^T \in \mathcal{X}. \tag{18}$$

Note that since $\hat{n}_w(\mathbf{m})$ is a weighted summation of i.i.d. Gaussian noises, $\hat{n}_w(\mathbf{m})$ is also i.i.d. Gaussian, i.e.,

$$\hat{n}_w(\mathbf{m}) \sim \mathcal{CN}(0, \sigma \|\mathbf{W}\|_2 / N). \tag{19}$$

The noise absolute amplitude, i.e., $|\hat{n}_w(\mathbf{m})|$ is i.i.d. Rayleigh distributed with mean equal to $\sigma_{\hat{n}'_w} \sqrt{\pi/2}$, where $\sigma_{\hat{n}'_w}$ is the standard deviation of the real or the imaginary component of $\hat{n}_w(\mathbf{m})$, and $\sigma_{\hat{n}'_w} = \sigma \|\mathbf{W}\|_2 / (\sqrt{2} N)$.

Since we need the side-lobe level of the strongest frequency to be below the noise level, i.e.,

$$\frac{\|\mathbf{W}\|_1}{N \rho_w} a_{max} < \frac{\sqrt{\pi} \sigma \|\mathbf{W}\|_2}{2N}, \tag{20}$$

it holds that

$$\rho_w > \frac{2\|\mathbf{W}\|_1}{\sqrt{\pi}\|\mathbf{W}\|_2}\sqrt{SNR_{max}}. \tag{21}$$

L Proof of (5.3)

Let us consider the decoding of the frequency location component m_0 from a 1-sparse bin. The decoding of m_1 is similar.

With noise, an 1-sparse bin contains the projection of one frequency $(a_d, \boldsymbol{\omega}) \in \mathbb{S}$ and noise component a_n, hence (4.11) becomes

$$\hat{s}(\tau_0, \tau_1) = a_d e^{j2\pi(m_0\tau_0/N_0 + m_1\tau_1/N_1)} + a_n(\tau_0, \tau_1), \tag{22}$$

where, for conciseness, we have ignored the line slope parameters $[\alpha_0, \alpha_1]^T$ and the bin number m as they are irrelevant to the decoding process.

The noise component a_n is due to the noise frequencies that are projected to $\hat{s}(\tau_0, \tau_1)$. According to Lemma 5, $a_n(\tau_0, \tau_1)$ is the summation of N/L samples of the $N_0 \times N_1$-point DFT of the i.i.d noise samples, which can be expressed as

$$a_n(\tau_0, \tau_1) = \sum_{\mathbf{m} \in \mathcal{P}_m} \hat{n}_w(\mathbf{m}) e^{j2\pi\left(\frac{m_0\tau_0}{N_0} + \frac{m_1\tau_1}{N_1}\right)}, \tag{23}$$

where $\mathbf{m} \triangleq [m_0, m_1]^T$; $\mathcal{P}_m \triangleq \{[m_0, m_1]^T : m_0, m_1 \text{ satisfy (4.8)}\}$ represents the frequency set that projects to the m-th bin of the DFT of the slice defined in (4.4) (see Lemma 5).

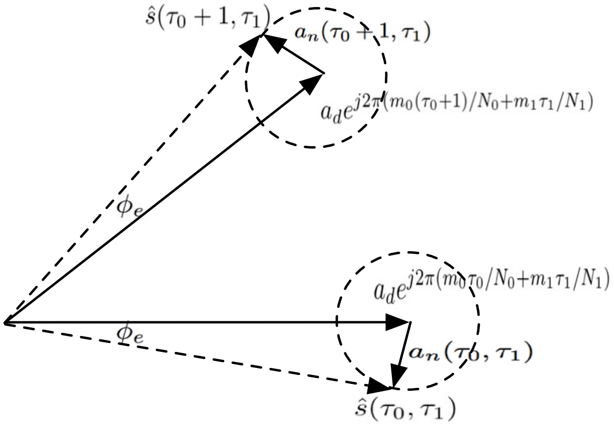
The same entry (the m-th bin) of the DFT along the other line with delay $[[\tau_0+1]_{N_0}, \tau_1]^T$ can be decomposed as

$$\hat{s}(\tau_0 + 1, \tau_1) = a_d e^{j2\pi(m_0(\tau_0+1)/N_0 + m_1\tau_1/N_1)} + a_n(\tau_0 + 1, \tau_1). \tag{24}$$

The frequency location m_0 is decoded as in (4.12). A graphical representation of the components of (22) and (24) is shown in the Figure below, from where one can see that the angular error ϕ_e changes with the rotation of the noise components $a_n(\tau_0, \tau_1), a_n(\tau_0 + 1, \tau_1)$. The angular error due to decoding, i.e., $\Delta\phi = |\phi\left(\frac{\hat{s}(\tau_0+1,\tau_1)}{\hat{s}(\tau_0,\tau_1)}\right) - \phi(e^{j2\pi m_0/N_0})|$ reaches maximum when $\hat{s}(\tau_0, \tau_1), \hat{s}(\tau_0+1, \tau_1)$ are out of phase and perpendicular to $a_n(\tau_0, \tau_1), a_n(\tau_0+1, \tau_1)$, respectively, as shown in the Figure. In such case, assuming that $|a_d| >> |a_n|$, we can use the following approximation:

$$\Delta\phi = 2|\phi_e| = 2\text{asin}(|\mathrm{a_n}|/|\mathrm{a_d}|) \approx 2|\mathrm{a_n}|/|\mathrm{a_d}|. \tag{25}$$

Since the localization error Δu due to $\Delta\phi$ has to be less than 1/2, i.e., $\Delta u = \frac{N_0}{2\pi}\Delta\phi < \frac{1}{2}$,


$\hat{s}(\tau_0+1,\tau_1)$
$a_n(\tau_0+1,\tau_1)$
$a_d e^{j2\pi(m_0(\tau_0+1)/N_0+m_1\tau_1/N_1)}$
ϕ_e
$a_d e^{j2\pi(m_0\tau_0/N_0+m_1\tau_1/N_1)}$
ϕ_e
$a_n(\tau_0,\tau_1)$
$\hat{s}(\tau_0,\tau_1)$

OFDM-trick for the 1-sparse bin with noise.

we need that

$$\frac{|a_d|}{|a_n|} > \frac{2N_0}{\pi}. \tag{26}$$

In the following, we derive the distribution of $|a_n|$ in order to derive the decoding error probability.

From (19) and (23), a_n also follows a circularly symmetric Gaussian distribution, i.e., $a_n \sim \mathcal{CN}(0, \sigma_{a_n})$, where $\sigma_{a_n} = \sigma\|\mathbf{W}\|_2/\sqrt{NL}$. Hence $|a_n|$ follows a Rayleigh distribution whose cumulative distribution function can be expressed as $f_{|a_n|}(x) = 1 - e^{-x^2/(2\sigma^2_{a'_n})}$, $x > 0$, where $\sigma^2_{a'_n} = \sigma^2_{a_n}/2$.

After windowing, in the DFT domain the highest peak of a frequency with the amplitude of a becomes $|a_d| = \|\mathbf{W}\|_1 a/N$. By substituting into (26), the correct decoding needs that $|a_n| < \delta_u \triangleq a\pi\|\mathbf{W}\|_1/(2NN_0)$. Hence, the decoding error for m_0 is upper bounded by

$$P_u = \left(\sigma_p(1 - f_{|a_n|}(\delta_u))\right)^2, \tag{27}$$

where σ_p with $1/2 \leq \sigma_p \leq 1/(2\pi)$ represents the probability of $a_n(\tau_0,\tau_1), a_n(\tau_0+1,\tau_1)$ being out-of-phase; the lower bound of σ_p represents to the probability of $a_n(\tau_0,\tau_1), a_n(\tau_0+1,\tau_1)$ pointing to the opposite direction and the upper bound of σ_p represents $a_n(\tau_0,\tau_1), a_n(\tau_0+1,\tau_1)$ pointing to the opposite direction and perpendicular to $\hat{s}(\tau_0,\tau_1), \hat{s}(\tau_0+1,\tau_1)$, respectively.

Similarly, the decoding error for m_1 is upper bounded as

$$P_v = \left(\sigma_p(1 - f_{|a_n|}(\delta_v))\right)^2, \tag{28}$$

where $\delta_v \triangleq a\pi\|\mathbf{W}\|_1/(2NN_1)$. Hence the lower bound of the probability of correct decoding of $[m_0, m_1]^T$ that projects to a 1-sparse bin is

$$P_w = (1 - P_u)(1 - P_v). \tag{29}$$

Next, the probability of a frequency being project to a 1-sparse bin is determined by

$$P_1 \triangleq (1 - |\mathbb{S}''|/N)^{N/L-1}, \tag{30}$$

where $L = \mathrm{LCM}(N_0, N_1)$; $\mathbb{S}''$ is the set of remaining frequencies in the signal in each iteration of robust MARS-SFT.

The success probability of the n_d-out-of-n_s voting decoding procedure can be expressed as the complementary cumulative binomial probability of (5.3), whose success probability of each experiment, i.e., localizing $(a_d, \boldsymbol{\omega})$ in each sub-iteration is $P_1 P_w$. This completes the proof.

M Detection Performance of the FFT-based Bartlett Method

The detection performance of the RSFT-based signal processing is characterized by (36) of [36], which establishes the relationship between the probability of detection, P_d, of the weakest sinusoid contained in the signal and the probability of false alarm, P_{fa}, of each frequency bin.

The counterpart of the RSFT-based signal processing is the FFT-based Bartlett method followed by an NP detection procedure [39, 40] (see Algorithm 2 for details). To compare the two methods, first of all, we derive the relationship between the P_d and the P_{fa} for the Bartlett method using the same signal model (see (2.1)) as that of the RSFT.

The analysis of the Bartlett method follows Algorithm 2. After windowing and FFT, the signal becomes

$$\hat{\mathbf{u}}_s = \mathbf{D}_N \mathbf{W} \mathbf{r}_s,\ s = 0, 1, \cdots, T-1, \tag{31}$$

where $\mathbf{D}_N \in \mathbb{C}^{N \times N}$ is the DFT matrix.

Substituting (2.1) into (31), for the kth, $k \in [N]$ entry of $\hat{\mathbf{u}}_s$, we get

$$[\hat{\mathbf{u}}_s]_k = \sum_{j \in [K]} b_{j,s} \mathbf{d}_k^H \mathbf{W} \mathbf{v}(\omega_j) + \mathbf{d}_k^H \mathbf{W} \mathbf{n}_s, \tag{32}$$

Algorithm 2 Bartlett Method algorithm

Input: complex signal $\mathbf{r}_s, s \in [T]$ in any fixed dimension

Output: $\mathbf{o}$, frequency domain representation of input signal

```
procedure BARTLETT($\mathbf{r}_s$)
    $\mathbf{x} \leftarrow 0$
    for $s = 0 \rightarrow T$ do
        Windowing: $\mathbf{u}_s \leftarrow \mathbf{W}\mathbf{r}_s$
        FFT: $\hat{\mathbf{u}}_s \leftarrow \mathrm{FFT}(\mathbf{u}_s)$
        Accumulation: $\mathbf{x} \leftarrow \mathbf{x} + |\hat{\mathbf{u}}_s|^2$
    end for
    Detection: $\mathbf{o} \leftarrow \mathrm{NPdet}(\mathbf{x})$
    return $\mathbf{o}$
end procedure
```

where $\mathbf{d}_k$ is the kth column of $\mathbf{D}_N$, i.e., $\mathbf{d}_k = [1 \quad e^{jk\Delta\omega_N} \ \cdots \ e^{jk(N-1)\Delta\omega_N}]^T$, and $\Delta\omega_N = 2\pi/N$.

Since $[\hat{\mathbf{u}}_s]_k$ is a linear combination of $b_{i,s}, [\mathbf{n}_s]_j, i \in [K], j \in [N]$, it is a circularly symmetry Gaussian scalar with distribution

$$[\hat{\mathbf{u}}_s]_k \sim \mathcal{CN}(0, \sigma_{uk}^2). \tag{33}$$

The hypothesis for each of N frequency bins are

- $\mathcal{H}'0$: no significant frequency exists.
- $\mathcal{H}'1$: there exists a significant frequency, with its SNR at least equals to SNR_{min}.

We assume the side-lobes of the significant frequencies are far below the noise level due to windowing, then under $\mathcal{H}'1$ and $\mathcal{H}'0$, respectively, we have the following approximation for σ_{uk}^2

$$\begin{aligned} \sigma_{uk|H1}^2 &\approx \sigma_{bm}^2\alpha' + \sigma_n^2\beta', \\ \sigma_{uk|H0}^2 &\approx \sigma_n^2\beta'. \end{aligned} \tag{34}$$

where $\alpha' = |\mathbf{d}_k^H \mathbf{W}\mathbf{v}(\omega_m)|^2$ and $\beta' = \|\mathbf{w}\|^2$.

The likelihood ratio test yields the sufficient statistics

$$l_k = \frac{1}{T}\sum_{s\in[T]} |[\hat{\mathbf{u}}_s]_k|^2 \underset{\mathcal{H}'0}{\overset{\mathcal{H}'1}{\gtrless}} \gamma. \tag{35}$$

We study its asymptotic performance. Assume that T is moderately large, then after applying central limit theory, the test statistic are Normal distributions under both hypothesis, i.e.,

$$\begin{aligned} l_{k|H0} &\sim \mathcal{N}(\sigma^2_{uk|H0}, \frac{\sigma^4_{uk|H0}}{T}), \\ l_{k|H1} &\sim \mathcal{N}(\sigma^2_{uk|H1}, \frac{\sigma^4_{uk|H1}}{T}). \end{aligned} \tag{36}$$

Finally, we can relate P_d and P_{fa} as

$$P_d = 1 - \Phi\left(\frac{\beta'\Phi^{-1}(1-P_{fa}) - \sqrt{T}\alpha' SNR_{min}}{\alpha' SNR_{min} + \beta'}\right), \tag{37}$$

where $\Phi(\cdot)$ is the cumulative distribution function (CDF) of standard normal distribution.

N Pseudo-code of robust MARS-SFT

The pseudo-code of robust MARS-SFT is shown in Algorithm 3. The input and output of the algorithm are as follows.

Input: Input signal function $r(\mathbf{n}), \mathbf{n} \in \mathcal{X}$; window function $w(\mathbf{n})$; data length for the two dimensions, N_0, N_1; number of iterations T; threshold of detecting significant frequencies in a slice ϵ; threshold for 1-sparsity detection γ; and parameters of n_d-out-of-n_s detection.

Output: the set $\mathbb{S}$ containing all the significant frequencies.

Note that the line length L and the set of slope parameters, $\mathcal{A}$, can be precomputed for efficiency. MARS-SFT can be viewed as a special case of the robust MARS-SFT, where the input signal is given in (4.1); the window is the rectangular window; and $\epsilon = 0, \gamma = 0, n_d = n_s = 1$.

Algorithm 3 (Robust) MARS-SFT algorithm

```
procedure MARS-SFT($r, w, N_0, N_1, T, \epsilon, \gamma, n_d, n_s$)
    $L \leftarrow \mathrm{LCM}(\mathrm{N_0}, \mathrm{N_1})$
    Compute the set of $\mathcal{A}$ defined in Lemma 5
    $\mathbb{S} \leftarrow \emptyset$
    for $t \leftarrow 1$ to $T$ do
        $\mathbb{S}' \leftarrow$ SFT-INNER ($r, w, N_0, N_1, \mathbb{S}, L, \epsilon, \gamma, n_d, n_s$)
        $\mathbb{S} \leftarrow \mathbb{S} \cup \mathbb{S}'$
    end for
    return $\mathbb{S}$
end procedure
```

```
procedure SFT-INNER($r, w, N_0, N_1, \mathcal{I}_m, L, \epsilon, \gamma, n_d, n_s$)
    $\mathbb{S}, \mathbb{S}' \leftarrow \emptyset$
    $O(\boldsymbol{\omega}) \leftarrow 0, \boldsymbol{\omega} \in \mathcal{X}$        ▷ Number of detection for a same location
    for $i \leftarrow 1$ to $n_s$ do
        Choose $\boldsymbol{\alpha}$ uniformly random from $\mathcal{A}$
        Choose $\boldsymbol{\tau}$ uniformly random from $\mathcal{X}$
        $\mathbb{S}_0 \leftarrow$ SLICING $(y, w, N_0, N_1, \mathcal{I}_m, L, \epsilon, \boldsymbol{\alpha}, \boldsymbol{\tau})$
        $\mathbb{S}_1 \leftarrow$ SLICING $(y, w, N_0, N_1, \mathcal{I}_m, L, \epsilon, \boldsymbol{\alpha}, \boldsymbol{\tau}_0)$
        $\mathbb{S}_2 \leftarrow$ SLICING $(y, w, N_0, N_1, \mathcal{I}_m, L, \epsilon, \boldsymbol{\alpha}, \boldsymbol{\tau}_1)$
        for $(m, \hat{s}_0) \in \mathbb{S}_0, (m, \hat{s}_1) \in \mathbb{S}_1, (m, \hat{s}_2) \in \mathbb{S}_2$ do
            if $\mathrm{Var}(|\hat{\mathrm{s}}_0|, |\hat{\mathrm{s}}_1|, |\hat{\mathrm{s}}_2|) < \gamma$ then        ▷ 1-sparse
                Decoding $(a, \boldsymbol{\omega})$ according to (4.12)
                $\mathbb{S}' \leftarrow \mathbb{S}' \cup (a, \boldsymbol{\omega})$, $O(\boldsymbol{\omega}) \leftarrow O(\boldsymbol{\omega}) + 1$
            end if
        end for
    end for
    for $(a, \boldsymbol{\omega}) \in \mathbb{S}'$ do
        if $O(\boldsymbol{\omega}) \geq n_d$ then
            $\mathbb{S} \leftarrow \mathbb{S} \cup (a, \boldsymbol{\omega})$
        end if
    end for
    return $\mathbb{S}$
end procedure
```

```
procedure SLICING($y, w, N_0, N_1, \mathcal{I}_m, L, \epsilon, \boldsymbol{\alpha}, \boldsymbol{\tau}$)
    $\mathbb{S} \leftarrow \emptyset$
    $s(l) \leftarrow y([\alpha_0 l + \tau_0]_{N_0}, [\alpha_1 l + \tau_1]_{N_1}), l \in [L]$
    $w(l) \leftarrow w([\alpha_0 l + \tau_0]_{N_0}, [\alpha_1 l + \tau_1]_{N_1}), l \in [L]$
    $s_w(l) \leftarrow s(l) * w(l), l \in [L]$
    $s_r(l) \leftarrow \mathrm{CONSTRUCTION}(\mathcal{I}_\mathrm{m}, \mathrm{N}_0, \mathrm{N}_1, \boldsymbol{\alpha}, \boldsymbol{\tau}, \mathrm{L})$
    $d(l) \leftarrow s_w(l) - s_r(l)$
    $\hat{d}(l) \leftarrow \frac{1}{L}\mathrm{DFT}(\mathrm{d(l)})$
    $\mathbb{S} \leftarrow \{(l, \hat{d}(l)) : |\hat{d}(l)| > \epsilon\}$
    return $\mathbb{S}$
end procedure
```

```
procedure CONSTRUCTION($\mathcal{I}_m, N_0, N_1, \boldsymbol{\alpha}, \boldsymbol{\tau}, L$)
    $\hat{s}_r(m) \leftarrow 0, m \in [L]$
    for $(a, \boldsymbol{\omega}) \in \mathcal{I}_m$ do
        $a' \leftarrow a e^{j2\pi(u\tau_0/N_0 + v\tau_1/N_1)}$
        $m = [\alpha_0 uL/N_0 + \alpha_1 vL/N_1]_L$
        $\hat{s}_r(m) = \hat{s}_r(m) + a'$
    end for
    $s_r(l) \leftarrow L\,\mathrm{IDFT}(\hat{\mathrm{s}}_\mathrm{r}(\mathrm{m}))$        ▷ Inverse DFT
    return $s_r(l), l \in [L]$
end procedure
```

References

[1] H. Hassanieh, M. Mayzel, L. Shi, D. Katabi, and V. Y. Orekhov, “Fast multi-dimensional NMR acquisition and processing using the sparse FFT,” *Journal of Biomolecular NMR*, pp. 1–11, 2015.

[2] L. Shi, H. Hassanieh, A. Davis, D. Katabi, and F. Durand, “Light field reconstruction using sparsity in the continuous Fourier domain,” *ACM Transactions on Graphics (TOG)*, vol. 34, no. 1, p. 12, 2014.

[3] B. Ghazi, H. Hassanieh, P. Indyk, D. Katabi, E. Price, and L. Shi, “Sample-optimal average-case sparse Fourier transform in two dimensions,” in *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, pp. 1258–1265, IEEE, 2013.

[4] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, “Simple and practical algorithm for sparse Fourier transform,” in *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’12, pp. 1183–1194, SIAM, 2012.

[5] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, “Nearly optimal sparse Fourier transform,” in *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, pp. 563–578, ACM, 2012.

[6] M. A. Richards, *Fundamentals of Radar Signal Processing.* Tata McGraw-Hill Education, 2005.

[7] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I: Detection, Estimation, and Linear Modulation Theory.* John Wiley & Sons, 2004.

[8] N. Levanon and E. Mozeson, *Radar Signals.* John Wiley & Sons, 2004.

[9] M. I. Skolnik, *Radar Handbook.* McGraw-Hill Education, 3rd ed., 2008.

[10] I. V. Komarov and S. M. Smolskiy, *Fundamentals of Short-Range FM Radar.* Artech House, 2003.

[11] S. M. Patole, M. Torlak, D. Wang, and M. Ali, “Automotive radars: A review of signal processing techniques,” *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 22–35, 2017.

[12] R. C. Hansen, *Phased Array Antennas*, vol. 213. John Wiley & Sons, 2009.

[13] M. A. Richards, J. Scheer, W. A. Holm, and W. L. Melvin, *Principles of Modern Radar.* Citeseer, 2010.

[14] D. J. Rabideau and P. A. Parker, “Ubiquitous MIMO multifunction digital array radar and the role of time-energy management in radar,” tech. rep., DTIC Document, 2004.

[15] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation*, ch. 5, pp. 349–350. Wiley, New York, 2002.

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

[17] A. M. Haimovich, R. S. Blum, and L. J. Cimini, "MIMO radar with widely separated antennas," *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 116–129, 2008.

[18] E. Fishler, A. Haimovich, R. Blum, R. Cimini, D. Chizhik, and R. Valenzuela, "Performance of mimo radar systems: advantages of angular diversity," in *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.*, vol. 1, pp. 305–309 Vol.1, Nov 2004.

[19] E. Fishler, A. Haimovich, R. Blum, D. Chizhik, L. Cimini, and R. Valenzuela, "MIMO radar: an idea whose time has come," in *Proceedings of the 2004 IEEE Radar Conference (IEEE Cat. No.04CH37509)*, pp. 71–78, April 2004.

[20] H. He, P. Stoica, and J. Li, "Designing unimodular sequence sets with good correlations—including an application to MIMO radar," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4391–4405, 2009.

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

[22] A. C. Gilbert, P. Indyk, M. Iwen, and L. Schmidt, "Recent developments in the sparse Fourier transform: A compressed Fourier transform for big data," *Signal Processing Magazine, IEEE*, vol. 31, no. 5, pp. 91–100, 2014.

[23] F. Ong, S. Pawar, and K. Ramchandran, "Fast sparse 2-D DFT computation using sparse-graph alias codes," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 4059–4063, IEEE, 2016.

[24] A. Rauh and G. R. Arce, "Sparse 2D fast Fourier transform," *Proceedings of the 10th International Conference on Sampling Theory and Applications*, 2013.

[25] P. Indyk and M. Kapralov, "Sample-optimal Fourier sampling in any constant dimension," in *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pp. 514–523, IEEE, 2014.

[26] L. Kämmerer, "Reconstructing multivariate trigonometric polynomials from samples along rank-1 lattices," in *Approximation Theory XIV: San Antonio 2013*, pp. 255–271, Springer, 2014.

[27] D. Potts and T. Volkmer, "Sparse high-dimensional FFT based on rank-1 lattice sampling," *Applied and Computational Harmonic Analysis*, 2015.

[28] D. Potts, M. Tasche, and T. Volkmer, "Efficient spectral estimation by MUSIC and ESPRIT with application to sparse fft," *Frontiers in Applied Mathematics and Statistics*, vol. 2, 2016.

[29] A. Christlieb, D. Lawlor, and Y. Wang, "A multiscale sub-linear time fourier algorithm for noisy data," *Applied and Computational Harmonic Analysis*, vol. 40, no. 3, pp. 553–574, 2016.

[30] P. Boufounos, V. Cevher, A. C. Gilbert, Y. Li, and M. J. Strauss, "What's the frequency, kenneth?: Sublinear Fourier sampling off the grid," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 61–72, Springer, 2012.

[31] E. Price and Z. Song, "A robust sparse Fourier transform in the continuous setting," in *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pp. 583–600, IEEE, 2015.

[32] S. Pawar and K. Ramchandran, "FFAST: An algorithm for computing an exactly k-sparse DFT in O(klogk) time," *IEEE Transactions on Information Theory*, vol. PP, no. 99, pp. 1–1, 2017.

[33] H. Hassanieh, F. Adib, D. Katabi, and P. Indyk, "Faster GPS via the sparse Fourier transform," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, pp. 353–364, ACM, 2012.

[34] H. Hassanieh, L. Shi, O. Abari, E. Hamed, and D. Katabi, "GHz-wide sensing and decoding using the sparse Fourier transform," in *INFOCOM, 2014 Proceedings IEEE*, pp. 2256–2264, IEEE, 2014.

[35] M. A. Iwen, "Combinatorial sublinear-time fourier algorithms," *Foundations of Computational Mathematics*, vol. 10, no. 3, pp. 303–338, 2010.

[36] S. Wang, V. M. Patel, and A. Petropulu, "The robust sparse Fourier transform (RSFT) and its application in radar signal processing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 6, pp. 2735–2755, 2017.

[37] S. Wang, V. M. Patel, and A. Petropulu, "Multidimensional sparse Fourier transform based on the Fourier projection-slice theorem," *IEEE Transactions on Signal Processing*, vol. 67, no. 1, pp. 54–69, 2019.

[38] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3042–3054, 2010.

[39] P. Stoica and R. L. Moses, *Spectral Analysis of Signals.* Pearson/Prentice Hall Upper Saddle River, NJ, 2005.

[40] H. So, Y. Chan, Q. Ma, and P. Ching, "Comparison of various periodograms for sinusoid detection and frequency estimation," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 35, no. 3, pp. 945–952, 1999.

[41] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978.

[42] M. Frigo and S. G. Johnson, "FFTW: An adaptive software architecture for the FFT," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 3, pp. 1381–1384, IEEE, 1998.

[43] D. Potts and M. Tasche, "Parameter estimation for exponential sums by approximate Prony method," *Signal Processing*, vol. 90, no. 5, pp. 1631–1642, 2010.

[44] D. L. Olson and D. Delen, *Advanced Data Mining Techniques.* Springer Science & Business Media, 2008.

[45] D. A. Schade, T. C. Winant, J. Alforque, J. Faul, K. B. Groves, V. Horvatich, M. A. Middione, C. Tarantino, and J. R. Turner, "Fast acting active protection system," Apr. 10 2007. US Patent 7,202,809.

[46] J. Dickmann, J. Klappstein, M. Hahn, N. Appenrodt, H.-L. Bloecher, K. Werber, and A. Sailer, "Automotive radar the key technology for autonomous driving: From detection and ranging to environmental understanding," in *2016 IEEE Radar Conference (RadarConf)*, pp. 1–6, IEEE, 2016.

[47] F. Engels, P. Heidenreich, A. M. Zoubir, F. K. Jondral, and M. Wintermantel, "Advances in automotive radar: A framework on computationally efficient high-resolution

frequency estimation," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 36–46, 2017.

[48] H. Sun, F. Brigui, and M. Lesturgie, "Analysis and comparison of MIMO radar waveforms," in *2014 International Radar Conference*, pp. 1–6, IEEE, 2014.

[49] K. Hamid and M. Viberg, "Two decades of array signal processing research," *IEEE signal processing magazine*, vol. 13, no. 4, pp. 67–94, 1996.

[50] L. Kämmerer, "Multiple rank-1 lattices as sampling schemes for multivariate trigonometric polynomials," *Journal of Fourier Analysis and Applications*, pp. 1–28, 2016.

[51] D. Hand and M. S.-W. Chen, "A non-uniform sampling ADC architecture with embeded alias-free asynchronous filter," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, pp. 3707–3712, IEEE, 2012.

[52] R. M. Mersereau and A. V. Oppenheim, "Digital reconstruction of multidimensional signals from their projections," *Proceedings of the IEEE*, vol. 62, no. 10, pp. 1319–1338, 1974.

[53] P. Duhamel and M. Vetterli, "Fast Fourier transforms: a tutorial review and a state of the art," *Signal processing*, vol. 19, no. 4, pp. 259–299, 1990.

[54] S. Wang, V. M. Patel, and A. Petropulu, "FPS-SFT: a multi-dimensional sparse Fourier transform based on the Fourier projection-slice theorem," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*, IEEE, 2018.

[55] J. O. Smith and X. Serra, *PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation.* CCRMA, Department of Music, Stanford University, 1987.

[56] M. Murad, I. Bilik, M. Friesen, J. Nickolaou, J. Salinger, K. Geary, and J. Colburn, "Requirements for next generation automotive radars," in *Radar Conference (RADAR), 2013 IEEE*, pp. 1–6, IEEE, 2013.

[57] F. Lalezari, "Orthogonal linear transmit receive array radar," Aug. 21 2012. US Patent 8,248,298.

[58] K. Shirakawa, S. Kobashi, Y. Kurono, M. Shono, and O. Isaji, "3D-scan millimeter-wave radar for automotive application," *Fujitsu Ten Tech. J*, vol. 38, pp. 3–7, 2013.

[59] R. E. Pavek and J. M. Willey, "Low cost 3D radar imaging and 3D association method from low count linear arrays for all weather autonomous vehicle navigation," Aug. 22 2017. US Patent 9,739,881.

[60] Y. Seliktar, *Space-time adaptive monopulse processing.* PhD thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, 1998.

[61] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4353–4361, 2015.

[62] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 1735–1742, IEEE, 2006.

[63] J. M. Butler, *Tracking and control in multi-function radar.* PhD thesis, University of London, 1998.

[64] K. Granstrom, M. Baum, and S. Reuter, "Extended object tracking: Introduction, overview and applications," *arXiv preprint arXiv:1604.00970*, 2016.

[65] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[66] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," in *Advances in neural information processing systems*, pp. 737–744, 1994.

[67] A. Akavia, "Deterministic sparse fourier approximation via fooling arithmetic progressions.," in *COLT*, pp. 381–393, 2010.

[68] M. A. Iwen, "Improved approximation guarantees for sublinear-time fourier algorithms," *Applied And Computational Harmonic Analysis*, vol. 34, no. 1, pp. 57–82, 2013.

[69] F. Bretz, P. Westfall, and T. Hothorn, *Multiple Comparisons Using R.* Chapman and Hall/CRC, 2016.

[70] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *Journal of the Royal statistical society: series B (Methodological)*, vol. 57, no. 1, pp. 289–300, 1995.

[71] R. B. Ash and C. Doleans-Dade, *Probability and Measure Theory*, p. 309. Academic Press, 2000.

[72] K. H. Rosen, *Elementary Number Theory and Its Applications.* Addison-Wesley, 1993.