Radar Imaging and Feature Extraction

Jian Li

Department of Electrical and Computer Engineering
P.O. Box 116130
University of Florida, Gainesville, FL 32611
Phone: (352) 392-2642; Fax: (352) 392-0044; Email: li@dsp.ufl.edu

Submitted as a final report for
the Office of Naval Research Grant ONR N00014-96-1-0817
1996-1999

Department of Electrical and Computer Engineering

Gainesville, Florida 32611

UNIVERSITY OF FLORIDA

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

DTIC QUALITY INSPECTED 4

19990617 061

Abstract

This document presents advanced spectral estimation methods for radar imaging and target feature extraction. In the document we study problems involved in inverse synthetic aperture radar (ISAR) autofocus and imaging, synthetic aperture radar (SAR) autofocus and motion compensation, superresolution SAR image formation, three-dimensional (3-D) target feature extraction via curvilinear SAR (CLSAR), and high resolution time delay estimation.

For the ISAR autofocus and imaging problems, we study both parametric and non-parametric methods. For the parametric method, we present a robust autofocus algorithm, referred to as AUTOCLEAN (AUTOfocus via CLEAN), for ISAR imaging. It is a parametric approach based on a very flexible data model that takes into account *arbitrary* translational and planar rotational motion. For the non-parametric methods, we study the general forward-backward MAtch-FIlterband (MAFI) spectral estimation approaches including the widely-used Capon as well as the more recently introduced APES (Amplitude and Phase EStimation) methods. We present an adaptive Capon spectral estimation algorithm and apply it to the complex ISAR image formation of maneuvering targets. We also present a recursive APES algorithm for time-varying spectral analysis and use it for ISAR imaging as well as feature extraction of targets with complex maneuvering motion.

For the problems of SAR autofocus and motion compensation, and superresolution SAR image formation, we study both semi-parametric and parametric methods. For the semi-parametric method, we present a SPAR (Semi-PARametric) algorithm based on a flexible data model for target feature extraction and superresolution complex image formation for SAR. SPAR exhibits better estimation resolution performance over non-parametric approaches and is more robust against data model errors than parametric ones. For the parametric methods, we propose a MCRELAX (Motion Compensation RELAX) algorithm and a MCCLEAN (Motion Compensation CLEAN) algorithm for simultaneous target feature extraction and cross-range phase error compensation in SAR imaging. Both MCRELAX and MCCLEAN assume a two-dimensional (2-D) sinusoidal model for the target signal but assume nothing (*arbitrary* unknown) for the phase error distribution, and are thus robust

algorithms against high-order phase errors. Compared to MCRELAX, MCCLEAN has better convergence property and is computationally much more efficient when used in the SAR imaging of a large scene.

For the 3-D target feature extraction problem, we study using CLSAR to extract target features. An AUTOfocus algorithm based on the RELAXation-based optimization approach (AUTORELAX) is proposed to compensate the aperture errors in CLSAR and to extract 3-D target features.

For the time delay estimation problem, we first present a Weighted Fourier transform and RELAXation-based (WRELAX) approach for the time delay estimation of either complex- or real-valued signals. WRELAX is then extended to deal with the real-valued signals with highly oscillatory correlation functions. Further, by using MODE (Method Of Direction Estimation) together with our efficient WRELAX algorithm, a novel MODE-WRELAX algorithm is proposed for the time delay estimation of either complex- or real-valued signals including those with highly oscillatory correlation functions to achieve superresolution.

Contents

1 Introduction **1**

2 A Robust Autofocus Algorithm for ISAR Imaging of Moving Targets **7**

2.1 Introduction 7

2.2 Problem Formulation 11

2.3 The AUTOCLEAN Algorithm 13

2.3.1 AUTOCLEAN based on a Single Dominant Scatterer 13

2.3.2 AUTOCLEAN based on Multiple Scatterers 15

2.4 Numerical and Experimental Results 20

2.5 Conclusions 21

3 Complex ISAR Imaging of Maneuvering Targets via the Capon Estimator **35**

3.1 Introduction 35

3.2 Problem Formulation 37

3.3 Forward-Only Capon 38

3.3.1 Brief Review of F-O Capon 38

3.3.2 Efficient Initialization Algorithm 40

3.3.3 Efficient Recursive Algorithm 41

3.3.4 Computational Complexity 45

3.4 Forward-Backward Capon 46

3.4.1 Brief Review of F-B Capon 46

3.4.2 Summary of the F-B Capon Algorithm 47

3.5 A Numerical Example 48

3.6 Conclusions 49

4 Time-Varying Complex Spectral Estimation with Application to ISAR Imaging **57**

4.1 Introduction 57

4.2 Problem Formulation 58

4.3 1-D Recursive APES 60

4.3.1 Overview of the 1-D APES Estimator 60

4.3.2 Efficient Initialization Method 60

4.3.3 Recursive Implementation 62

4.4 2-D Recursive APES 65
4.4.1 Overview of the 2-D APES Estimator 65
4.4.2 Efficient Initialization Method 66
4.4.3 Recursive Implementation 67
4.5 Numerical and Experimental Examples 69
4.6 Conclusions 71

5 A Semi-Parametric Spectral Estimation Approach to SAR Target Feature Extraction and Image Formation 76
5.1 Introduction 76
5.2 Problem Formulation 78
5.3 Effects of the Semi-Parametric Data Model on SAR Image Formation 79
5.3.1 Model Ambiguities 79
5.3.2 Image Formation 81
5.3.3 Model Ambiguity Effects on SAR Image Formation 83
5.4 The SPAR Algorithm 85
5.4.1 Target Feature Extraction 85
5.5 Modified RELAX-NLS Algorithm 90
5.6 Numerical and Experimental Results 91
5.7 Conclusions 94

6 Synthetic Aperture Radar Motion Compensation and Feature Extraction via a Relaxation Based Algorithm 106
6.1 Introduction 106
6.2 Data Model and Problem Formulation 107
6.3 The MCRELAX Algorithm 112
6.4 The CRB of the Parameter Estimates 115
6.5 Numerical and Experimental Examples 117
6.6 Conclusions 121

7 Autofocus and Super Resolution SAR Image Formation 134
7.1 Introduction 134
7.2 Data Model and Problem Formulation 137
7.3 Preparations 141
7.4 The MCCLEAN Algorithm 144
7.5 Experimental Examples 149

7.6 Conclusions 151

8 Super Resolution SAR Imaging via Parametric Spectral Estimation Methods **157**

8.1 Introduction 157

8.2 SAR Image Formation via Spectral Estimation Methods 159

8.2.1 RELAX 159

8.2.2 RELAX-NLS 161

8.3 Experimental Results 163

8.4 Conclusions 165

9 Using Curvilinear SAR for Three-Dimensional Target Feature Extraction **178**

9.1 Introduction 178

9.2 Data Models and Problem Formulations 179

9.2.1 High Range Resolution Radar 179

9.2.2 Full Synthetic Aperture Radar 180

9.2.3 Curvilinear SAR 184

9.3 The RELAX Algorithm 184

9.3.1 Full Aperture 185

9.3.2 Curvilinear Aperture 187

9.4 Cramér-Rao Bound of the Paramaeter Estimates 188

9.5 Numerical and Experimental Results 189

9.5.1 Performance Analysis of Different Curvilinear Apertures via CRBs . . 190

9.5.2 Experimental Examples 191

9.6 Conclusions 194

10 Autofocus and Feature Extraction in Curvilinear SAR **204**

10.1 Introduction 204

10.2 Data Model and Problem Formulation 205

10.2.1 High Range Resolution Radar 205

10.2.2 Curvilinear Synthetic Aperture Radar 206

10.2.3 Aperture Errors 210

10.3 The Relaxation-Based Autofocus Algorithm 213

10.4 Experimental and Numerical Results 218

10.5 Conclusions 220

11 An Efficient Algorithm for Time Delay Estimation **231**
11.1 Introduction 231
11.2 Problem Formulation 233
11.3 The WRELAX Algorithm 234
11.4 The Extended WRELAX Algorithms for Multiple Looks 239
11.4.1 Fixed Delays but Arbitrary Gains 239
11.4.2 Fixed Delays and Gains 241
11.5 Simulation and Experimental Results 245
11.6 Conclusions 249

12 Time Delay Estimation via Optimizing Highly Oscillatory Cost Functions **262**
12.1 Introduction 262
12.2 Problem Formulation 264
12.3 The Hybrid-WRELAX Algorithm 266
12.4 The EXIP-WRELAX Algorithm 270
12.5 Simulation Results and Discussions 273
12.6 Conclusions 275

13 Super Resolution Time Delay Estimation via MODE-WRELAX **286**
13.1 Introduction 286
13.2 Data Model and Problem Statement 288
13.3 The MODE-WRELAX Algorithm 291
13.3.1 MODE-WRELAX for Complex-Valued Signals 291
13.3.2 MODE-WRELAX for Real-Valued Signals 296
13.3.3 Extensions to Multiple Looks 298
13.4 Efficient Implementation of MODE-WRELAX 299
13.4.1 MODE 299
13.4.2 WRELAX 301
13.5 Numerical Examples 302
13.6 Conclusions 308

1. Introduction

This document presents advanced spectral estimation methods for radar imaging and target feature extraction. In the document we study problems involved in inverse synthetic aperture radar (ISAR) autofocus and imaging, synthetic aperture radar (SAR) autofocus and motion compensation, superresolution SAR image formation, three-dimensional (3-D) target feature extraction via curvilinear SAR (CLSAR), and high resolution time delay estimation.

For the ISAR autofocus and imaging problems, we study both parametric and non-parametric methods. For the parametric method, we present a robust autofocus algorithm, referred to as AUTOCLEAN (AUTOfocus via CLEAN), for ISAR imaging. It is a parametric approach based on a very flexible data model that takes into account *arbitrary* translational and planar rotational motion. For the non-parametric methods, we study the general forward-backward MAtch-FIlterband (MAFI) spectral estimation approaches including the widely-used Capon as well as the more recently introduced APES (Amplitude and Phase EStimation) methods. We present an adaptive Capon spectral estimation algorithm and apply it to the complex ISAR image formation of maneuvering targets. We also present a recursive APES algorithm for time-varying spectral analysis and use it for ISAR imaging as well as feature extraction of targets with complex maneuvering motion.

For the problems of SAR autofocus and motion compensation, and superresolution SAR image formation, we study both semi-parametric and parametric methods. For the semi-parametric method, we present a SPAR (Semi-PARametric) algorithm based on a flexible data model for target feature extraction and superresolution complex image formation for SAR. SPAR exhibits better estimation resolution performance over non-parametric approaches and is more robust against data model errors than parametric ones. For the parametric methods, we propose a MCRELAX (Motion Compensation RELAX) algorithm and a MCCLEAN (Motion Compensation CLEAN) algorithm for simultaneous target feature extraction and cross-range phase error compensation in SAR imaging. Both MCRELAX and MCCLEAN assume a two-dimensional (2-D) sinusoidal model for the target signal but assume nothing (*arbitrary* unknown) for the phase error distribution, and are thus robust

algorithms against high-order phase errors. Compared to MCRELAX, MCCLEAN has better convergence property and is computationally much more efficient when used in the SAR imaging of a large scene.

For the 3-D target feature extraction problem, we study using CLSAR to extract target features. An AUTOfocus algorithm based on the RELAXation-based optimization approach (AUTORELAX) is proposed to compensate the aperture errors in CLSAR and to extract 3-D target features.

For the time delay estimation problem, we first present a Weighted Fourier transform and RELAXation-based (WRELAX) approach for the time delay estimation of either complex- or real-valued signals. WRELAX is then extended to deal with the real-valued signals with highly oscillatory correlation functions. Further, by using MODE (Method Of Direction Estimation) together with our efficient WRELAX algorithm, a novel MODE-WRELAX algorithm is proposed for the time delay estimation of either complex- or real-valued signals including those with highly oscillatory correlation functions to achieve superresolution.

This document contains 13 chapters. In Chapters 2-4, we study ISAR autofocus and imaging methods. In Chapters 5-8, we present methods for SAR autofocus and motion compensation, and superresolution SAR image formation. In Chapters 9-10, we study 3-D target feature extraction via CLSAR. The high resolution time delay estimation methods are presented in Chapters 11-13.

In Chapter 2, we present the AUTOCLEAN algorithm for the motion compensation of ISAR imaging of moving targets. It is a parametric algorithm based on a very flexible data model which takes into account *arbitrary* range migration and *arbitrary* phase errors across the synthetic aperture that may be induced by unwanted radial motion of the target as well as propagation or system instability. The autofocusing is accomplished by minimizing a nonlinear least squares (NLS) fitting criterion by using an efficient relaxation-based optimization approach. Compared to other existing algorithms, our method is more robust since it does not rely on the existence of isolated prominent point scatterers for each range profile nor on the precise modeling of the motion trajectory of the moving targets of interest. Significant improvement on the image quality is observed with the NATO raw data.

In Chapter 3, we present an adaptive Capon spectral estimation algorithm for the complex ISAR image formation of maneuvering targets. It has better resolution and lower sidelobes than the short-time Fourier transform (STFT) method. The algorithm is an efficient recursive implementation of the 2-D Capon complex spectral estimator, which involves only fast Fourier transform (FFT) and simple matrix operations. ISAR imaging examples of maneuvering targets are provided to illustrate the performance of the proposed method.

In Chapter 4, we apply the APES algorithm to sliding short-time data sequences with maximal overlapping for the time-varying complex spectral analysis. A computationally efficient recursive APES algorithm is developed, which involves only FFT and simple matrix operations. It exhibits much better resolution than STFT. ISAR imaging examples show that it can successfully circumvent the image blurring problem caused by target maneuvering.

In Chapter 5, we first establish a flexible data model, which models each target scatterer as a 2-D complex sinusoid with arbitrary amplitude and constant phase in cross-range and with constant amplitude and phase in range, and then present the SPAR algorithm for SAR target feature extraction and superresolution image formation based on the established data model. By taking advantage of both parametric and non-parametric spectral estimation methods, SPAR exhibits better estimation and resolution performance over non-parametric approaches and is more robust against data modeling errors than parametric methods. Both numerical and experimental results demonstrate the performance of the proposed SPAR algorithm.

In Chapter 6, we study the problem of extracting target features via SAR in the presence of uncompensated aperture motion errors. A parametric data model for a spotlight-mode SAR system is established. The Cramér-Rao bounds (CRBs) for the parameters of the data model are also derived. The CRB analysis shows that the unknown motion errors can significantly affect the accuracy of a common shift of the scatterer position in the cross-range direction, but have little effect on other target parameters including the accuracy of the relative positions in range and cross-range direction. A relaxation-based MCRELAX algorithm for estimating both target features and motion errors is devised. Simulation results show that the mean-squared errors of the parameter estimates obtained by using the MCRELAX

algorithm can approach the corresponding CRBs. We also show that MCRELAX can simply be used for motion compensation only and can give better performance than the well-known Phase-Gradient Autofocus (PGA) algorithm.

In Chapter 7, We present the MCCLEAN algorithm for correcting synthetic aperture phase errors in SAR. It is a parametric algorithm based on the same data model as used in MCRELAX. The computational core of the algorithm is the CLEAN algorithm, which involves only a sequence of 2-D FFT operations. MCCLEAN is robust against high-order phase errors. Compared to MCRELAX, MCCLEAN has better convergence property (no separate initialization step is required) and is computationally much more efficient when used as an independent autofocus approach for the SAR imaging of a large scene. For certain kinds of scene content and phase error distributions, MCCLEAN performs better than the PGA algorithm. We also present a modified relaxation-based algorithm, which has a similar structure as MCCLEAN, for simultaneous autofocus and superresolution target feature extraction of a small scene or small region of interest (ROI) in a large scene. Experimental examples with a portion of the data collected by the ERIM's DCS interferometric SAR (IFSAR) system show that the proposed algorithms are very effective.

In Chapter 8, we consider superresolution SAR image formation via sophisticated parametric spectral estimation algorithms. Parametric spectral estimation methods are devised based on parametric data models and are used to estimate the model parameters. We use the parameter estimates obtained with the parametric methods to simulate data matrices of large dimensions and then use the FFT methods on them to generate SAR images with superresolution. Experimental examples using the MSTAR and ERIM data illustrate that the robust spectral estimation algorithms can generate SAR images of higher resolution than the conventional FFT methods and enhance the dominant target features.

In Chapter 9, we consider using CLSAR for 3-D feature extraction of small targets consisting of a small number of distinct point scatterers. CLSAR does not suffer from the ambiguities suffered by IFSAR. Since CLSAR is a relatively new technology, a self-contained detailed derivation of the data model is presented. The CRBs of the parameter estimates are also derived. We also describe how the RELAX algorithm can be used for 3-D target

feature extraction with CLSAR for different curvilinear apertures.

In Chapter 10, we present the AUTORELAX algorithm, which can be used to compensate for the aperture errors in CLSAR and to extract 3-D target features. A self-contained detailed derivation of the data model for the autofocus problem in CLSAR is presented. Experimental and simulation results show that AUTORELAX can be used to significantly improve the estimation accuracy of the target parameters.

In Chapter 11, we present the WRELAX algorithm for the time delay estimation problem. The method is a relaxation-based global minimizer of a complicated NLS criterion. WRELAX involves only a sequence of weighted Fourier transforms and hence the superior estimation performance of the NLS fitting approach is achieved at a much lower implementation cost. The new algorithm is successfully applied to detecting and classifying roadway subsurface anomalies by using an ultra wideband ground penetrating radar. It is also extended to the case of multiple looks for different scenarios (*i.e.*, fixed delays but arbitrary gains and fixed delays and gains). CRB analysis and numerical and experimental examples are provided to demonstrate the performance of the new algorithm.

In Chapter 12, we propose two approaches based on WRELAX to deal with the problem of optimizing highly oscillatory cost functions. One approach (referred to as Hybrid-WRELAX) uses the last step of the WRELAX algorithm to minimize the true NLS cost function corresponding to the real-valued signal amplitudes. The other one (referred to as EXIP-WRELAX) uses the extended invariance principle (EXIP). They are relaxation-based global minimizers of a highly oscillatory NLS cost function. Both of the algorithms are shown to approach the CRB and require only a sequence of weighted Fourier transforms.

In Chapter 13, we study estimating time delays and amplitudes (real- or complex-valued) from the superposition of very closely spaced signals with known shapes. Particularly, we modify the well-known high resolution MODE algorithm and use it with our efficient WRELAX algorithm to deal with superresolution time delay estimation. The proposed new method is referred to as MODE-WRELAX. MODE-WRELAX provides better accuracy than MODE and higher resolution than WRELAX. Moreover, it can be used for both complex- and real-valued signals including those with highly oscillatory correlation functions. Nu-

merical results show that the MODE-WRELAX estimates can approach the corresponding CRB. Efficient implementation of the algorithm is discussed as well.

Each of the aforementioned chapters is self-contained with its own introduction, formulation of the problem of interest, detailed presentation of approaches, conclusions, and references.

Those who have contributed to this report include Mr. Zhaoqiang Bi, Dr. Victor C. Chen, Dr. Kenneth Knaell, Mr. Hongbin Li, Dr. Jian Li, Dr. Zheng-She Liu, Dr. Petre Stoica, Dr. Renbiao Wu, and Mr. Edmund G. Zelnio.

2. A Robust Autofocus Algorithm for ISAR Imaging of Moving Targets

2.1 Introduction

ISAR (inverse synthetic aperture radar) imaging [1] of moving targets is very important for many military and civilian applications including ATR (automatic target recognition) of non-cooperative aircrafts [2, 3, 4, 5, 6], battlefield awareness [7], development as well as maintenance of low observable aircrafts [8] and target characterization [9, 10, 11], Moon and planet imaging in radio astronomy [12], and the surveillance of ground traffic on airports [13, 14]. Compared to the conventional low-resolution wide area surveillance radar, ISARs offer improved detection and tracking performance and exclusive target identification capability, which is desirable for a modern radar. Because of this, many countries in the world are now trying to shift this technology from laboratory to practice.

The principles underlying ISAR and SAR (synthetic aperture radar) can be unified within the framework of turntable imaging [12]. Today, the SAR technology, producing high-resolution maps and images of stationary targets in real-time, is a well established technology and nearly 30 spaceborne and airborne SAR systems are currently in operation for a wide range of military and civilian applications and more are being built around the world [15]. On the other hand, ISAR imaging is still at the R&D (Research and Development) stage and only a few experimental systems have been built [8, 9, 10, 16]. The reason to such an inbalanced development is that the relative motion between the radar and the target is cooperative in SAR and hence is easier to be compensated out than the non-cooperative relative motion in ISAR. In SAR imaging of stationary targets, the navigation data available on the moving platform carrying the radar can be exploited to determine a preliminary estimation of the motion parameters. Many sophisticated motion compensation algorithms have been proposed for SAR imaging and it appears that the remaining problem is how to make a better tradeoff between the image quality and the computational cost. However, the motion compensation in ISAR imaging is much more complicated than in the case of SAR imaging of stationary targets since the radar tracking data cannot achieve the

accuracy required to generate a recognizable image and the motion parameters can only be obtained via data-based autofocus algorithms. How to devise robust and efficient autofocus algorithms has become the major problem in ISAR imaging since once focused, the ISAR images could be formed by using the well-established SAR imaging technology.

Unlike in SAR, the radar used for ground-to-air ISAR imaging is usually stationary. The relative motion needed to obtain the synthetic aperture is induced by the moving target itself. Usually, the target motion with respect to the radar line of sight (RLOS) can be decomposed into a radial motion of an arbitrary reference point on the target and a tangential motion about the reference point [1]. The tangential motion can be used to form a synthetic aperture to provide the needed high resolution in cross-range, whereas the radial motion must be compensated out since it has nothing to do with ISAR imaging but can cause range migration and phase errors across the synthetic aperture. Successful compensation of the unwanted radial motion is crucial to ISAR imaging.

The diagram of conventional Range-Doppler (R-D) ISAR image processing is shown in Figure 2.1. The phase history data (dechirped or demodulated step-frequency signal) received by the radar receiver is first range compressed. Next, radial motion is compensated out via rough range alignment followed by fine cross-range phase correction. Finally, Fourier transform or other super resolution spectral analysis methods [18, 19, 20] can be used to generate the ISAR images of targets of interest. In the literature, algorithms for range alignment are fairly standard. It can be done either by envelope cross-correlation or tracking the time history of a reference point (such as the peak or the centroid) in the range compressed data and fitting it to a polynomial [1]. However, the requirement for cross-range phase tracking is much more stringent than that for range alignment and the range errors

[1]For arbitrarily maneuvering targets with rigid bodies, the motion can be decomposed into translational motion of a reference point on the target and rotational motion (yaw, pitch, roll) with respect to that reference point [17]. In this chapter, we assume that the non-planar motion can be ignored. This assumption is valid in most cases and is widely used in the literature. Under this assumption, the target motion can be decomposed into radial and tangential components. Herein tangential motion refers to the equivalent rotation caused by the translational motion of the reference point as well as the target self rotation with respect to the reference point.

must be controlled to within a small fraction of a radar wavelength. For example, for radar working at X band with a wavelength of 3 centimeters, a range change of 3 millimeters would produce a phase error of 72°. Because of this, much efforts have been put on the cross-range phase estimation and many algorithms have been proposed. Most of the existing algorithms obtain the cross-range phase errors via tracking the phase history of a single well-isolated dominant scatterer on the target [1, 21, 22] (referred to as Dominant Scatterer Algorithm (DSA) or Prominent Point Processing (PPP)), a synthesized scatterer such as the centroid of multiple scatterers [23, 24, 25, 26] (referred to as Multiple Scatterer Algorithm (MSA)), the statistic scattering centroid (SSC) [27], or the Doppler centroid [28, 29]. DSAs [1, 21, 22] perform very well when there is an isolated dominant scatterer on the target. However, this requirement cannot be satisfied in most cases due to target scintillation and shadowing effect. Centroid-based algorithms [23, 24, 25, 26, 27, 28, 29] can relax this requirement in some way and hence are more robust than DSAs. However, phase averaging is needed by most of these algorithms. Without correct phase unwrapping (which is not an easy task), the averaging procedure will do more harm than good. The well-known SAR autofocus algorithm, referred to as the PGA (phase gradient autofocus) algorithm [30, 31], can also be used for the cross-range phase compensation in ISAR imaging. However, we have found that its performance depends on the quality of range alignment and on the choice of the threshold needed by the automatic windowing step or the size reduction rate needed by the progressive windowing scheme. Several variations of DSAs [32, 33] have also been proposed for cross-range phase error estimation, which use low-order polynomials (usually quadratic) to model the phase errors and then obtain the polynomial coefficients by optimizing the maximum likelihood (ML) cost functions. All of the aforementioned methods attempt to achieve the range alignment and phase correction in separate steps and the range alignment quality may have a significant impact on the phase estimation algorithms and the final image quality. Moreover, many of the above algorithms are implicitly based on the assumption of the availability of well-isolated dominant scatterers and hence cannot work effectively in most practical cases.

In addition to the above separate processing approaches, parametric motion estimation algorithms for joint range alignment and cross-range phase tracking have also been proposed

in the literature [34]. In [34], low-order polynomials are used to model the phase variations due to target motions in the phase history domain and then the polynomial coefficients are obtained via optimizing image focus indicators such as the entropy measures [34]. Since the image domain entropy method is computationally very intensive, a phase history domain burst derivative algorithm was also proposed in [34]. The burst derivative algorithm is computationally more efficient than the entropy method but the burst derivative is periodic and there is no unique global minimum and hence can only be used in combination with the entropy method for the fine adjustment of motion estimates. Moreover, like any other parametric motion estimation algorithms based on low-order polynomial phase error models [32, 33], both the entropy and burst derivative methods are sensitive to system instabilities.

In this chapter, a robust autofocus approach, referred to as AUTOCLEAN (AUTOfocus via CLEAN), is proposed for the motion compensation in ISAR imaging of moving targets. It is a parametric algorithm based on a very flexible data model, which takes into account *arbitrary* range migration and *arbitrary* phase errors across the synthetic aperture that may be induced by the unwanted radial motion of the target as well as propagation or system instability. AUTOCLEAN can be classified as an MSA, but it differs considerably from other existing MSAs [23, 24, 25, 26] since it automatically selects multiple scatterers (not necessarily well-isolated or very dominant) in the two-dimensional (2-D) image domain and combines their phase and RCS (radar cross section) information in an optimal way, which avoids the troublesome phase unwrapping step. Another good feature associated with AUTOCLEAN is that it can be easily configured for different ISAR applications. Numerical and experimental results have shown that AUTOCLEAN is a very robust autofocus tool for ISAR imaging.

The remainder of this chapter is organized as follows. Section 2.2 describes our flexible data model for ISAR imaging and formulates the problem of interest. The AUTOCLEAN algorithm is presented in Section 2.3. Several examples are provided in Section 2.4 to demonstrate the performance of AUTOCLEAN. Finally, Section 2.5 concludes the chapter.

2.2 Problem Formulation

For ground-to-air ISAR imaging systems, the radar is stationary and the relative motion needed to obtain the synthetic aperture is induced by the moving target itself. When non-planar motion is negligible, the target motion with respect to the RLOS can be decomposed into radial motion of an arbitrary reference point on the target and tangential motion with respect to the reference point. In this case, the geometry of the radar and the target is illustrated by Figure 2.2. In Figure 2.2, $R_0(t)$ denotes the distance from the reference point, (X_0, Y_0) in the $X-Y$ coordinate system, which is the origin of the $x-y$ coordinate system, to the radar at time instant t, $R(t)$ denotes the distance between an arbitrary point (x, y) on the target and the radar, and $\theta(t)$ represents the relative angle of the target with respect to the local $x-y$ coordinate system.

When the variation of $\theta(t)$ is small during the coherent processing interval (CPI), the target can usually be viewed as consisting of a few point scatterers with constant RCS's and fixed locations [1] in the local $x-y$ coordinate system. Assume that a normalized linear FM (chirp) signal is transmitted, which has the form

$$s(t) = e^{j(2\pi f_0 t + \gamma t^2)}, \quad |t| \leq \frac{T_0}{2}, \tag{2.1}$$

where f_0 denotes the carrier frequency, γ is the chirp rate, and T_0 is the pulse width. Then the signal received by a radar receiver after dechirping (or deramping) and A/D conversion is [12, 35]:

$$\begin{aligned} r(n, \bar{n}) &= \left\{ \sum_{k=1}^{K} \alpha_k e^{j4\pi f_n [x_k \cos\theta(\bar{n}T) + y_k \sin\theta(\bar{n}T)]/c} \right\} e^{j[4\pi f_n \triangle R_0(\bar{n}T)/c]} + e(n, \bar{n}), \\ & \quad 0 \leq n \leq N-1, \quad 0 \leq \bar{n} \leq \bar{N}-1, \end{aligned} \tag{2.2}$$

where K is the number of scatterers on the target; α_k, x_k, and y_k are the complex amplitude (RCS and phase), range, and cross-range locations of the kth scatterer, respectively; c denotes the speed of light; N is the available sample points for each received chirp pulse and $\bar{N}$ is the number of along track positions; T is the pulse repetition interval; f_n is the discretized

frequency and can be related directly to the time samples as follows

$$f_n = f_0 + \frac{\gamma}{\pi} t_n, \tag{2.3}$$

with t_n denoting the nth sample of the *fast time* measured with respect to the tracked time delay of the reference point; $\triangle R_0(\bar{n}T)$ denotes the range migration, which is the distance at the along track position $\bar{n}$ between the true reference point (*unknown*) and the one tracked by the radar; finally, $e(n, \bar{n})$ denotes the clutter and noise.

In addition to linear FM chirp pulses, step-frequency signal waveform is also widely used for ISAR imaging because of the simplicity of its design and implementation. It is a discrete implementation of the linear FM waveform. The wide bandwidth of a linear FM signal is obtained by transmitting a burst of narrow bandwidth pulses with different carrier frequencies at each along track position. The above data model (2.2) is still valid for the transmitted step-frequency signals. In this case, f_n is just the center frequency of the nth transmitted pulse within each burst and a burst of N pulses is compressed to form a range profile at each along track position $\bar{n}$. The pulse repetition interval T in (2.2) now becomes the burst period.

Like most of the existing ISAR autofocus algorithms, we assume that the CPI is short and $\triangle\theta = \left|\theta\left[(\bar{N}-1)T\right] - \theta(0)\right|$ is small and the tangential motion is approximately uniform. Under this assumption, (2.2) can be simplified as

$$\begin{aligned} r(n, \bar{n}) &= \left[\sum_{k=1}^{K} \alpha_k e^{j(\omega_k n + \bar{\omega}_k \bar{n})}\right] e^{j[\omega_a(\bar{n})n + \psi(\bar{n})]} + e(n, \bar{n}), \\ & \quad 0 \le n \le N-1, \quad 0 \le \bar{n} \le \bar{N}-1, \end{aligned} \tag{2.4}$$

where the frequency pair $(\omega_k, \bar{\omega}_k)$ is proportional to the location (x_k, y_k) of the kth scatterer, $\omega_a(\bar{n})$ is proportional to the range migration $\triangle R_0(\bar{n}T)$, and $\{\psi(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}$ denote *arbitrary* cross-range phase errors [2].

[2] If the phase errors are only caused by the target motion, then $\psi(\bar{n})$ is also proportional to the range migration $\triangle R_0(\bar{n}T)$ and is determined by $\psi(\bar{n}) = 4\pi f_0 \triangle R_0(\bar{n}T)/c$. In this paper, we assume $\{\psi(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}$ to be *arbitrary* to include other phase errors due to propagation or system instability.

Our problem of interest herein is to estimate $\{\omega_a(\bar{n}), \psi(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}$ from the phase history data $\{r(n,\bar{n})\}$ received by the radar. Once the estimates $\{\hat{\omega}_a(\bar{n}), \hat{\psi}(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}$ of $\{\omega_a(\bar{n}), \psi(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}$ are available, motion compensation can be done by simply multiplying $r(n,\bar{n})$ with $e^{-j\left[\hat{\omega}_a(\bar{n})n+\hat{\psi}(\bar{n})\right]}$ and the remaining task is to apply, for example, the conventional Range-Doppler imaging algorithm to the compensated phase history data, which can be efficiently implemented by using 2-D FFT.

2.3 The AUTOCLEAN Algorithm

As pointed out previously, AUTOCLEAN is an MSA. It can also be used as a DSA and the upgrade from DSA to MSA is very straightforward. For the sake of clarification, we first derive below the AUTOCLEAN algorithm based on a single dominant scatterer and then extend it to the case where multiple dominant scatterers are used.

2.3.1 AUTOCLEAN based on a Single Dominant Scatterer

Assume that there is a single dominant scatterer on the target with complex amplitude and 2-D location $(\alpha_1, \omega_1, \bar{\omega}_1)$. We can rewrite (2.4) in the following form:

$$\begin{aligned} r(n,\bar{n}) &= s_1(n,\bar{n})e^{j[\omega_a(\bar{n})n+\psi(\bar{n})]} + e_1(n,\bar{n}), \\ &\qquad 0 \le n \le N-1, \quad 0 \le \bar{n} \le \bar{N}-1, \end{aligned} \tag{2.5}$$

where

$$s_1(n,\bar{n}) = \alpha_1 e^{j(\omega_1 n+\bar{\omega}_1\bar{n})}, \tag{2.6}$$

and $e_1(n,\bar{n})$ denotes the unmodeled target return plus the clutter and noise component $e(n,\bar{n})$. Then the estimates of $\{\omega_a(\bar{n}), \psi(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}$ and $(\alpha_1, \omega_1, \bar{\omega}_1)$ can be obtained via optimizing the following nonlinear least-squares (NLS) criterion

$$C_1\left(\alpha_1, \omega_1, \bar{\omega}_1, \{\omega_a(\bar{n}), \psi(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}\right) = \sum_{n=0}^{N-1}\sum_{\bar{n}=0}^{\bar{N}-1}\left|r(n,\bar{n}) - \alpha_1 e^{j(\omega_1 n+\bar{\omega}_1\bar{n})}e^{j[\omega_a(\bar{n})n+\psi(\bar{n})]}\right|^2. \tag{2.7}$$

The above NLS problem can be solved by using an alternating optimization approach, which iterates the two steps of motion estimation and feature extraction [3]. The two steps are outlined as follows.

Motion Estimation

Assume that the feature estimates $\left\{\hat{\alpha}_1, \hat{\omega}_1, \hat{\bar{\omega}}_1\right\}$ of $\{\alpha_1, \omega_1, \bar{\omega}_1\}$ are given. Let

$$\hat{s}_1(n, \bar{n}) = \hat{\alpha}_1 e^{j(\hat{\omega}_1 n + \hat{\bar{\omega}}_1 \bar{n})}. \tag{2.8}$$

Then minimizing $C_1\left(\alpha_1, \omega_1, \bar{\omega}_1, \{\omega_a(\bar{n}), \psi(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}\right)$ becomes minimizing

$$C_2\left(\{\omega_a(\bar{n}), \psi(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}\right) = \sum_{n=0}^{N-1} \sum_{\bar{n}=0}^{\bar{N}-1} \left| r(n, \bar{n}) - \hat{s}_1(n, \bar{n}) e^{j[\omega_a(\bar{n})n + \psi(\bar{n})]} \right|^2, \tag{2.9}$$

or

$$C_3\left(\omega_a(\bar{n}), \psi(\bar{n})\right) = \sum_{n=0}^{N-1} \left| r(n, \bar{n}) - \hat{s}_1(n, \bar{n}) e^{j[\omega_a(\bar{n})n + \psi(\bar{n})]} \right|^2, \tag{2.10}$$

which yields

$$\hat{\omega}_a(\bar{n}) = \arg\max_{\omega_a(\bar{n})} \left| \sum_{n=0}^{N-1} e^{-j\omega_a(\bar{n})n} \left[\hat{s}_1^*(n, \bar{n}) r(n, \bar{n})\right] \right|, \tag{2.11}$$

and

$$\hat{\psi}(\bar{n}) = \text{angle}\left\{ \sum_{n=0}^{N-1} e^{-j\hat{\omega}_a(\bar{n})n} \left[\hat{s}_1^*(n, \bar{n}) r(n, \bar{n})\right] \right\}, \tag{2.12}$$

where angle(x) denotes the phase of x and $(\cdot)^*$ the complex conjugate. Note that $\hat{\omega}_a(\bar{n})$ can be obtained via 1-D FFT with zero-paddings (for high accuracy) and $\hat{\psi}(\bar{n})$ can be calculated easily as well.

Feature Extraction

Now assume that the motion estimates $\{\hat{\omega}_a(\bar{n}), \hat{\psi}(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}$ of $\{\omega_a(\bar{n}), \psi(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}$ are given. Then minimizing $C_1\left(\alpha_1, \omega_1, \bar{\omega}_1, \{\omega_a(\bar{n}), \psi(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}\right)$ becomes minimizing

$$C_4\left(\alpha_1, \omega_1, \bar{\omega}_1\right) = \sum_{n=0}^{N-1} \sum_{\bar{n}=0}^{\bar{N}-1} \left| \tilde{r}(n, \bar{n}) - \alpha_1 e^{j(\omega_1 n + \bar{\omega}_1 \bar{n})} \right|^2, \tag{2.13}$$

[3]Feature extraction is a terminology widely used in pattern recognition. Many target recognition systems use the complex amplitudes and locations of target scatterers as features for classification. Herein we borrow this terminology to represent the estimation of complex amplitudes and locations of target scatterers.

where

$$\tilde{r}(n,\bar{n}) = r(n,\bar{n})e^{-j\left[\hat{\omega}_a(\bar{n})n+\hat{\psi}(\bar{n})\right]}. \tag{2.14}$$

After simple mathematical manipulations [36], the feature estimates $\left\{\hat{\alpha}_1, \hat{\omega}_1, \hat{\bar{\omega}}_1\right\}$ of $\{\alpha_1, \omega_1, \bar{\omega}_1\}$ can be determined as

$$\left(\hat{\omega}_1, \hat{\bar{\omega}}_1\right) = \arg\max_{\omega_1,\bar{\omega}_1} \left| \sum_{n=0}^{N-1} \sum_{\bar{n}=0}^{\bar{N}-1} \tilde{r}(n,\bar{n}) e^{-j(\omega_1 n+\bar{\omega}_1\bar{n})} \right|^2, \tag{2.15}$$

and

$$\hat{\alpha}_1 = \frac{\sum_{n=0}^{N-1}\sum_{\bar{n}=0}^{\bar{N}-1} \tilde{r}(n,\bar{n}) e^{-j(\hat{\omega}_1 n+\hat{\bar{\omega}}_1\bar{n})}}{N\bar{N}}. \tag{2.16}$$

Note that $\left(\hat{\omega}_1, \hat{\bar{\omega}}_1\right)$ are obtained as the location of the dominant peak of the scaled 2-D periodogram $\left|\sum_{n=0}^{N-1}\sum_{\bar{n}=0}^{\bar{N}-1} \tilde{r}(n,\bar{n}) e^{-j(\omega_1 n+\bar{\omega}_1\bar{n})}\right|^2$, which can be efficiently computed by using 2-D FFT with zero-paddings. Then $\hat{\alpha}_1$ is simply the complex height of the peak of $\sum_{n=0}^{N-1}\sum_{\bar{n}=0}^{\bar{N}-1} \tilde{r}(n,\bar{n}) e^{-j(\hat{\omega}_1 n+\hat{\bar{\omega}}_1\bar{n})}/N\bar{N}$.

AUTOCLEAN is an alternating optimization approach to the NLS criterion in (2.7) which iterates the above two steps. To speed up the convergence, we use the envelope cross-correlation method [1] for the initial motion estimation. With the above preparations, we now summarize the AUTOCLEAN algorithm based on a single dominant scatterer.

Step 0: Obtain the initial motion estimates via the envelope cross-correlation method.

Step 1: Compute the feature estimates $\left\{\hat{\alpha}_1, \hat{\omega}_1, \hat{\bar{\omega}}_1\right\}$ of $\{\alpha_1, \omega_1, \bar{\omega}_1\}$ by using (2.15) and (2.16).

Step 2: Calculate the motion parameters by using (2.11) and (2.12).

Step 3: Repeat Steps 1 and 2 until the relative change of the cost function in (2.7) between two consecutive iterations is less than some pre-determined threshold, say 10^{-3}.

2.3.2 AUTOCLEAN based on Multiple Scatterers

Like other existing DSAs [1, 21, 22], when there is a well-isolated very dominant scatterer on the target, the above AUTOCLEAN algorithm works very well and the latter performs slightly better than the former since the latter can track the range migration more accurately than the former. However, the assumption about the availability of such an isolated dominant

scatterer is more often than not violated in practice. This problem can be alleviated by tracking the centroid of multiple strong but not necessarily very dominant scatterers. This is the basic idea behind MSAs. Existing MSAs [23, 24, 25, 26] need troublesome phase unwrapping and do not exploit the RCS information of each scatterer. Below, we avoid this problem by extending the AUTOCLEAN algorithm to the case of multiple scatterers.

Assume that there are $\breve{K}$ strong scatterers with features $\{\alpha_k, \omega_k, \bar{\omega}_k\}_{k=1}^{\breve{K}}$ (selected automatically by the AUTOCLEAN algorithm). (Note that $\breve{K}$ can be much smaller than the true number K of the target scatterers.) Let $\tilde{K}$ denote an intermediate number of strong scatterers, i.e., $\tilde{K} = 1,\ 2,\ \cdots,\ \breve{K}$. By assuming $\tilde{K}$ strong scatterers, we can, similarly to (2.5), rewrite (2.4) as

$$\begin{aligned} r(n,\bar{n}) &= s_{\tilde{K}}(n,\bar{n})e^{j[\omega_a(\bar{n})n+\psi(\bar{n})]} + e_{\tilde{K}}(n,\bar{n}), \\ & \qquad 0 \le n \le N-1, \quad 0 \le \bar{n} \le \bar{N}-1, \end{aligned} \tag{2.17}$$

where

$$s_{\tilde{K}}(n,\bar{n}) = \sum_{k=1}^{\tilde{K}} \alpha_k e^{j(\omega_k n + \bar{\omega}_k \bar{n})}, \tag{2.18}$$

and $e_{\tilde{K}}(n,\bar{n})$ denotes the unmodeled target return plus the clutter and noise component $e(n,\bar{n})$. Both the motion estimates and the feature estimates of the $\tilde{K}$ scatterers can be obtained via minimizing the following NLS criterion

$$C_5\left(\{\alpha_k,\omega_k,\bar{\omega}_k\}_{k=1}^{\tilde{K}}, \{\omega_a(\bar{n}),\psi(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}\right) = \sum_{n=0}^{N-1}\sum_{\bar{n}=0}^{\bar{N}-1}\left| r(n,\bar{n}) - \left[\sum_{k=1}^{\tilde{K}} \alpha_k e^{j(\omega_k n+\bar{\omega}_k\bar{n})}\right] e^{j[\omega_a(\bar{n})n+\psi(\bar{n})]}\right|^2. \tag{2.19}$$

Before we present the AUTOCLEAN algorithm utilizing multiple scatterers, let us first introduce the two steps (motion estimation and feature extraction), which constitute the computational kernel of the proposed algorithm.

Motion Estimation

The motion estimation algorithm presented in the previous subsection can be directly extended to the multiple scatterer case. Assume that the feature estimates $\left\{\hat{\alpha}_k, \hat{\omega}_k, \hat{\bar{\omega}}_k\right\}_{k=1}^{\tilde{K}}$

of $\{\alpha_k, \omega_k, \bar{\omega}_k\}_{k=1}^{\tilde{K}}$ are given. Let

$$\hat{s}_{\tilde{K}}(n, \bar{n}) = \sum_{k=1}^{\tilde{K}} \hat{\alpha}_k e^{j(\hat{\omega}_k n + \hat{\bar{\omega}}_k \bar{n})}, \tag{2.20}$$

denote the estimate of $s_{\tilde{K}}(n, \bar{n})$ in (2.18). By replacing the $\hat{s}_1(n, \bar{n})$ in Equations (2.9) through (2.12) with $\hat{s}_{\tilde{K}}(n, \bar{n})$, we get

$$\hat{\omega}_a(\bar{n}) = \arg\max_{\omega_a(\bar{n})} \left| \sum_{n=0}^{N-1} e^{-j\omega_a(\bar{n})n} \left[\hat{s}^*_{\tilde{K}}(n, \bar{n}) r(n, \bar{n}) \right] \right|, \tag{2.21}$$

and

$$\hat{\psi}(\bar{n}) = \text{angle} \left\{ \sum_{n=0}^{N-1} e^{-j\hat{\omega}_a(\bar{n})n} \left[\hat{s}^*_{\tilde{K}}(n, \bar{n}) r(n, \bar{n}) \right] \right\}. \tag{2.22}$$

Feature Extraction

Once the motion estimates $\{\hat{\omega}_a(\bar{n}), \hat{\psi}(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}$ are available, the feature estimates $\left\{\hat{\alpha}_k, \hat{\omega}_k, \hat{\bar{\omega}}_k\right\}_{k=1}^{\tilde{K}}$ can be obtained via minimizing

$$C_6\left(\{\alpha_k, \omega_k, \bar{\omega}_k\}_{k=1}^{\tilde{K}}\right) = \sum_{n=0}^{N-1} \sum_{\bar{n}=0}^{\bar{N}-1} \left| \tilde{r}(n, \bar{n}) - \sum_{k=1}^{\tilde{K}} \alpha_k e^{j(\omega_k n + \bar{\omega}_k \bar{n})} \right|^2, \tag{2.23}$$

where $\tilde{r}(n, \bar{n})$ is defined in (2.14). Relaxation-based optimization techniques including CLEAN [36, 37] and RELAX [36] may be used to deal with the above NLS optimization problem. We have found (see Appendix) that for our problem of interest, in which motion estimation is of the major concern and feature estimates are only by-products of the optimization process, CLEAN is computationally much more efficient than RELAX and can provide similar motion estimation performance than the latter and hence is preferred.

Before we summarize the CLEAN algorithm, let us first present the following preparations. Assume $\left\{\hat{\alpha}_i, \hat{\omega}_i, \hat{\bar{\omega}}_i\right\}_{i=1, i\neq k}^{\tilde{K}}$ are given. Let

$$z_k(n, \bar{n}) = \tilde{r}(n, \bar{n}) - \sum_{i=1, i\neq k}^{\tilde{K}} \alpha_i e^{j(\omega_i n + \bar{\omega}_i \bar{n})}. \tag{2.24}$$

Then the estimates of $(\alpha_k, \omega_k, \bar{\omega}_k)$ can be obtained via minimizing

$$C_7\left(\alpha_k, \omega_k, \bar{\omega}_k\right) = \sum_{n=0}^{N-1} \sum_{\bar{n}=0}^{\bar{N}-1} \left| z_k(n, \bar{n}) - \alpha_k e^{j(\omega_k n + \bar{\omega}_k \bar{n})} \right|^2, \tag{2.25}$$

which yields

$$\left(\hat{\omega}_k, \hat{\bar{\omega}}_k\right) = \arg\max_{\omega_k, \bar{\omega}_k} \left| \sum_{n=0}^{N-1} \sum_{\bar{n}=0}^{\bar{N}-1} z_k(n, \bar{n}) e^{-j(\omega_k n + \bar{\omega}_k \bar{n})} \right|^2, \tag{2.26}$$

and

$$\hat{\alpha}_k = \frac{\sum_{n=0}^{N-1} \sum_{\bar{n}=0}^{\bar{N}-1} z_k(n, \bar{n}) e^{-j(\hat{\omega}_k n + \hat{\bar{\omega}}_k \bar{n})}}{N\bar{N}}. \tag{2.27}$$

With the above preparations, we now summarize the CLEAN algorithm. Let $\bar{K}$ denote the intermediate number of target scatterers.

Step (1): Assume $\bar{K} = 1$. Obtain $\left\{\hat{\omega}_1, \hat{\bar{\omega}}_1\right\}$ and $\hat{\alpha}_1$ from $\tilde{r}(n, \bar{n})$ by using (2.26) and (2.27), respectively.

Step (2): Assume $\bar{K} = 2$. Compute $z_2(n, \bar{n})$ with (2.24) by using $\left\{\hat{\omega}_i, \hat{\bar{\omega}}_i, \hat{\alpha}_i\right\}_{i=1}$ obtained in Step (1). Obtain $\left\{\hat{\omega}_2, \hat{\bar{\omega}}_2\right\}$ and $\hat{\alpha}_2$ from $z_2(n, \bar{n})$ by using (2.26) and (2.27), respectively.

Step (3): Assume $\bar{K} = 3$. Compute $z_3(n, \bar{n})$ with (2.24) by using $\left\{\hat{\omega}_i, \hat{\bar{\omega}}_i, \hat{\alpha}_i\right\}_{i=1}^{2}$ obtained in Steps (1) and (2). Obtain $\left\{\hat{\omega}_3, \hat{\bar{\omega}}_3\right\}$ and $\hat{\alpha}_3$ from $z_3(n, \bar{n})$ by using (2.26) and (2.27), respectively.

Remaining Steps: Continue similarly until $\bar{K} = \tilde{K}$.

The flow chart of the proposed AUTOCLEAN algorithm is shown in Figure 2.3, which iterates the above motion estimation and feature extraction steps.

The proposed AUTOCLEAN algorithm can be summarized as follows.

Step 0: Obtain the initial motion estimates via the envelope cross-correlation method.

Step 1: Assume $\tilde{K} = 1$.

Substep (a): Obtain $\left\{\hat{\omega}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\right\}_{k=1}$ by using Step (1) of CLEAN.

Substep (b): Calculate $\{\hat{\omega}_a(\bar{n}), \hat{\psi}(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}$ by using (2.21) and (2.22).

Substep (c): Compute $\left\{\hat{\omega}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\right\}_{k=1}$ via CLEAN by assuming $\tilde{K}$ strong scatterers.

Substep (d): If practical convergence (to be discussed later on) is achieved, then go to the next step; otherwise, go back to Substep 1(b).

Step 2: Assume $\tilde{K} = 2$.

Substep (a): Obtain $\left\{\hat{\omega}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\right\}_{k=2}$ by using Step (2) of CLEAN.

Substep (b): Calculate $\{\hat{\omega}_a(\bar{n}), \hat{\psi}(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}$ by using (2.21) and (2.22).

Substep (c): Compute $\left\{\hat{\omega}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\right\}_{k=1}^{2}$ via CLEAN by assuming $\tilde{K}$ strong scatterers.

Substep (d): If practical convergence is achieved, then go to the next step; otherwise, go back to Substep 2(b).

Step 3: Assume $\tilde{K} = 3$.

Substep (a): Obtain $\left\{\hat{\omega}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\right\}_{k=3}$ by using Step (3) of CLEAN.

Substep (b): Calculate $\{\hat{\omega}_a(\bar{n}), \hat{\psi}(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}$ by using (2.21) and (2.22).

Substep (c): Compute $\left\{\hat{\omega}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\right\}_{k=1}^{3}$ via CLEAN by assuming $\tilde{K}$ strong scatterers.

Substep (d): If practical convergence is achieved, then go to the next step; otherwise, go back to Substep 3(b).

Remaining Steps: Continue similarly until $\tilde{K} = \breve{K}$.

The "practical convergence" in Step $\tilde{K}$ of the above AUTOCLEAN algorithm may be determined by checking whether the relative change of the cost function in (2.19) between two consecutive iterations is less than some pre-determined threshold, say 10^{-3}.

We remark that the above AUTOCLEAN algorithm differs considerably from other MSAs [23, 24, 25, 26] in several aspects: a) dominant scatterers are selected automatically in the 2-D image domain; b) scatterers may not be well-isolated or very dominant; c) complex amplitude information from each selected scatterer is combined in an optimal way; d) the troublesome phase unwrapping step is avoided. Compared to other parametric algorithms [32, 33, 34], AUTOCLEAN is more robust since it is based on a more flexible data model than the former and can more effectively mitigate the interference among different scatterers. The computational kernel of AUTOCLEAN is CLEAN, which is computationally very efficient and involves only a sequence of 2-D FFTs that can be easily implemented using currently available FFT chips, such as TMC2310 [38], A41102 [39], and TM-66 *swiFFT* [40]. Hence it can be easily configured for real-time applications. Another good feature associated with AUTOCLEAN is that its performance can be progressively improved by assuming a reasonably larger number of strong scatterers, $\breve{K}$, for the target. In other words, $\breve{K}$ can be either pre-determined or increased until the autofocused ISAR image is satisfactory.

2.4 Numerical and Experimental Results

We have tested AUTOCLEAN with a large confidential set of measured ISAR data [41]. The proposed algorithm has proved to be very robust and significantly outperforms other existing algorithms (including DSAs, MSAs, and PGA) [41]. Unfortunately, since the data set is classified, we are not authorized to report the experimental results in the open literature. Instead, we use numerical examples to demonstrate the performance of AUTOCLEAN.

In the following examples, we use the simulated MIG-25 aircraft data provided by the Naval Research Laboratory as the turntable target data and then add range migration and phase errors to the turntable data to simulate a moving aircraft. The data matrix is 64×64 (i.e., $N = \bar{N} = 64$). The $\hat{\omega}_a(\bar{n})$ in (2.11) and (2.21) is obtained via 1-D FFT with zero-padding to 128. The $\left(\hat{\omega}_1, \hat{\bar{\omega}}_1\right)$ in (2.15) and $\left(\hat{\omega}_k, \hat{\bar{\omega}}_k\right)$ in (2.26) are obtained via 2-D FFT with zero-padding to 128×128. We have used 10^{-3} to test the practical convergence of our algorithm. Windowed (Taylor window with parameter 4 and 50 dB sidelobe level [42]) 2-D FFT with zero-padding to 256×256 is used to form the ISAR images. The performance of the proposed AUTOCLEAN algorithm is also compared with that of other two popular autofocus approaches including Haywood and Evans' MSA [24] and PGA [30]. (Other existing MSAs [23, 25, 26] perform similarly to Haywood and Evans' method and hence are not considered herein.) Since Haywood and Evans' MSA and PGA can only be used for phase correction, the conventional envelope cross-correlation method [1] is used first for initial motion compensation, particularly for range alignment.

Let us first consider an example in which there is an isolated very dominant scatterer on the target. The 2-D image and 3-D mesh plot of the target are shown in Figures 2.4(a) and (b), respectively. Figure 2.4(c) shows the uncompensated image from which we can observe severe image blurring due to the target motion. The ISAR images after autofocusing using Haywood and Evans' MSA, PGA, and AUTOCLEAN are shown in Figures 2.4(d), (e), and (f), respectively. In this example, only one dominant scatterer is used by Haywood and Evans' MSA and AUTOCLEAN and hence both methods are in fact DSAs. Note that, as

expected, all of the three algorithms produce well focused ISAR images and AUTOCLEAN performs slightly better than the other two approaches since AUTOCLEAN can provide much more accurate range alignment than the envelope cross-correlation method.

Now we consider an example where the target does not have a single very strong isolated dominant scatterer. The 2-D image and 3-D mesh plot of the target are shown in Figures 2.5(a) and (b), respectively. Figure 2.5(c) shows the original uncompensated blurred image. Five scatterers (i.e., $\check{K} = 5$) are used by Haywood and Evans' MSA and AUTOCLEAN to extract the motion parameters. The ISAR images obtained via autofocusing using Haywood and Evans' MSA, PGA, and AUTOCLEAN are shown in Figures 2.5(d), (e), and (f), respectively. Note that AUTOCLEAN significantly outperforms the other two methods. Haywood and Evans' MSA completely fails in this example since several strong scatterers share the same range and they are not selected as the dominant scatterers used for motion estimation according to the minimum variance criterion [24]. Instead, other less dominant scatterers containing less accurate phase error information are selected and hence the phase averaging procedure does more harm than good since the use of these less dominant scatterers makes phase error estimation worse. Again, AUTOCLEAN gives the best result.

Now we compare the computational efficiency of Haywood and Evans' MSA, PGA, and AUTOCLEAN. Since both PGA and AUTOCLEAN are iterative algorithms, it is hard to give explicit expressions for their computational complexities. Instead, we roughly compare their computational times needed on an ordinary PC with a Pentium II 400 MHz CPU. The computational times for Haywood and Evans' MSA, PGA, and AUTOCLEAN are 0.39, 1.39, and 1.76 seconds, respectively, for the first example and 0.40, 8.59, and 10.38 seconds, respectively, for the second example. Note that AUTOCLEAN and PGA require similar computational times for our examples.

2.5 Conclusions

We have presented a robust ISAR autofocus algorithm, referred to as AUTOCLEAN, which is an efficient parametric algorithm based on a very flexible data model. It is more ro-

bust than other existing parametric autofocus algorithms since the former assumes *arbitrary* range migration and *arbitrary* cross-range phase errors and does not rely on the availability of a well-isolated very dominant scatterer. AUTOCLEAN is a multiple scatterer algorithm (MSA). By utilizing phase as well as RCS information of each scatterer in an optimal way and avoiding the troublesome phase unwrapping procedure, AUTOCLEAN significantly outperforms other existing MSAs. AUTOCLEAN is computationally very efficient and requires only a sequence of FFTs. Hence it can be easily implemented in real-time by using currently available high speed FFT chips. Numerical and experimental results have shown that AUTOCLEAN is a very robust autofocus tool for ISAR imaging.

Acknowledgement

The authors would like to thank Mr. J. Man for preparing some of the figures used in this chapter.

Appendix: Further Discussion on Feature Extraction Algorithms

For the feature extraction step discussed in Subsection 2.3.2, besides CLEAN, another popular approach is RELAX [36]. CLEAN is computationally more efficient than RELAX but its resolution and estimation accuracy are inferior to RELAX [36]. Hence RELAX is preferred in many applications where high resolution and high estimation accuracy of individual scatterers are desired. However, for our problem of interest, motion estimation is of the major concern and the feature estimates are only by-products of the optimization process. By comparing (2.4) with (2.18), it can be observed that the accuracy of the motion estimates is mainly determined by the target fitting error $\sum_{n=0}^{N-1}\sum_{\bar{n}=0}^{\bar{N}-1}|\hat{s}_{\check{K}}(n,\bar{n}) - s_K(n,\bar{n})|^2$ rather than by the accuracy of feature estimates $\{\hat{\alpha}_k, \hat{\omega}_k, \hat{\bar{\omega}}_k\}_{k=1}^{\check{K}}$. We have found that although the resolution and accuracy of CLEAN are inferior to RELAX [36], the target fitting error of the former is very close to that of the latter. This is especially true when the number of scatterers used for motion estimation ($\check{K}$) is small as compared to the true number of scatterers on the target (K). Below, we provide a simple example to illustrate this observation.

Assume that a target consists of $K = 8$ scatterers with equal strength (i.e., $\alpha_k = 1$ for $k = 1, 2, \cdots, 8$). The eight scatterers are distributed in the pattern as shown in Figure 2.6(a). There are four pairs of scatterers with each pair having a spacing around 2/3 of the conventional FFT resolution limit. Hence the scatterer pairs cannot be resolved by using FFT, as can be seen from Figure 2.6(b), which shows the original windowed FFT image of the target. Range migration and phase errors, which are the same as those used in Figures 2.4 and 2.5, are added to the simulated target. For the sake of clarification, no noise is added (i.e., $e(n, \bar{n}) = 0$) in this example. This is a very tough situation for ISAR autofocusing since all scatterers are of equal strength and spaced very closely.

Ambiguities exist for the data model in (2.4). Let β_0, β_1, and γ_0 denote arbitrary constants and define

$$\tilde{\psi}(\bar{n}) = \beta_0 + \beta_1 \bar{n} + \psi(\bar{n}), \tag{2.28}$$

$$\tilde{\omega}_a(\bar{n}) = \gamma_0 + \omega_a(\bar{n}), \tag{2.29}$$

$$\tilde{\alpha}_k = \alpha_k e^{-j\beta_0}, \tag{2.30}$$

$$\tilde{\omega}_k = \omega_k - \gamma_0, \tag{2.31}$$

$$\tilde{\bar{\omega}}_k = \bar{\omega}_k - \beta_1. \tag{2.32}$$

Then it can be easily verified that $\left(\{\tilde{\alpha}_k, \tilde{\omega}_k, \tilde{\bar{\omega}}_k\}_{k=1}^{K}, \{\tilde{\omega}_a(\bar{n}), \tilde{\psi}(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}\right)$ also satisfy the data model in (2.4). Nevertheless, these ambiguous solutions will only shift the original ISAR images and hence will not affect the image quality. However, we cannot evaluate the estimation accuracy for most of the parameters based on this ambiguous data model.

Yet we note that no ambiguity occurs to the magnitude of the complex amplitude and hence they can be used to evaluate the feature estimation performance. Figure 2.6(c) compares the estimates of $\{|\alpha_k|\}_{k=1}^{K}$ obtained via CLEAN ("×") and RELAX ("○") to the true values (dot-dashed line) where the horizontal axis denotes the index of the scatterer (k). Note that, as expected, the RELAX feature estimates of individual scatterers are unbiased whereas the CLEAN estimates are not. To evaluate the motion estimation performance, we use the entropy as an image focus indicator. Entropy is a terminology borrowed from

information theory and was used for ISAR autofocus in [34] (see also the references therein). Assume $U(l,\bar{l}), \quad l = 0, 1, \cdots, L-1, \ \bar{l} = 0, 1, \cdots, \bar{L}-1$, denote the complex ISAR image after autofocusing using the motion estimates. Then we calculate the entropy E of the focused image as follows:

$$E = -\sum_{l=0}^{L-1} \sum_{\bar{l}=0}^{\bar{L}-1} \tilde{U}(l,\bar{l}) \log \tilde{U}(l,\bar{l}), \tag{2.33}$$

where

$$\tilde{U}(l,\bar{l}) = \frac{|U(l,\bar{l})|}{\sum_{l=0}^{L-1} \sum_{\bar{l}=0}^{\bar{L}-1} |U(l,\bar{l})|}. \tag{2.34}$$

In Figure 2.6(d), the entropy of the focused ISAR images obtained by using CLEAN ("×") and RELAX ("o") as the feature extraction methods is compared to the original value (dot-dashed line), which corresponds to the original image shown in Figure 2.6(b). In Figure 2.6(d), the horizontal axis denotes the number of scatterers used for motion estimation ($\breve{K}$). Note that, although CLEAN is inferior to RELAX in resolution capability and estimation accuracy for feature extraction, they provide similar motion compensation performance. As $\breve{K}$ is increased from 1 to 8, almost perfect motion compensation, as indicated by image entropy, is achieved via AUTOCLEAN. Figure 2.7 further illustrates this observation. Figure 2.7(a) shows the windowed FFT image after initial motion compensation using the envelope cross-correlation method. Note that the image is still blurred due to uncompensated motion errors. Figures 2.7(b) through (i) show the windowed FFT images after autofocusing via AUTOCLEAN by increasing $\breve{K}$ from 1 to 8. Note that when $\breve{K} = 4$, the image is basically focused but there is still some blurring. When $\breve{K} = 8$, the image is almost identical to the true one shown in Figure 2.6(b).

Reference

[1] C. C. Chen and H. C. Andrews, "Target-motion-induced radar imaging," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-16, pp. 2–14, January 1980.

[2] L. M. Novak, "A comparison of 1-D and 2-D algorithms for radar target classification," *Proceedings of the 1991 IEEE International Conference on Systems Engineering*, pp. 6–12, August 1-3, 1991, Fairborn, OH, USA.

[3] M. M. Menon, E. R. Boudreau, and P. J. Kolodzy, "An automatic ship classification system for ISAR imagery," *The MIT Lincoln Laboratory Journal*, vol. 6, no. 2, pp. 289–308, 1993.

[4] E. C. Botha, "Classification of aerospace targets using super resolution ISAR images," *Proceedings of the 1994 IEEE South African Symposium on Communications and Signal Processing*, pp. 138–145, October 4, 1994, Stellenbosch, South Africa.

[5] T. Fechner, R. Hantsche, and R. Tanger, "Classification of objects in ISAR imagery using artificial neural networks," *Proceedings of SPIE: Applications and Science of Artificial Neural Networks II*, pp. 339–345, April 9-12, 1996, Orlando, FL, USA.

[6] S. Musman, D. Kerr, and C. Bachmann, "Automatic recognition of ISAR ship images," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 4, pp. 1392–1404, October 1996.

[7] M. T. Fennell and R. P. Wishner, "Battlefield awareness via synergistic SAR and MTI exploitation," *IEEE Aerospace and Electronic Systems Magazine*, vol. 13, no. 2, pp. 39–45, February, 1998.

[8] R. Goodman, W. Nagy, J. Wilhelm, and S. Crippen, "A high fidelity ground to air imaging radar system," *The Record of the 1994 IEEE National Radar Conference*, pp. 29–34, March 29-31, 1994, Atlanta, Georgia, USA.

[9] R. Voles, "Resolving revolutions: imaging and mapping by modern radar," *IEE Proceedings, Pt. F*, vol. 140, no. 1, pp. 1–11, February 1993.

[10] G. Gombert and F. Bechner, "High resolution 2-D ISAR image collection and processing," *Proceedings of the IEEE 1994 National Aerospace and Electronics Conference*, pp. 371–377, May 23-27, 1994, Dayton, OH, USA.

[11] A. Jain and I. Patel, "Dynamic imaging and RCS measurements of aircrafts," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, no. 1, pp. 211–226, January 1995.

[12] D. A. Ausherman, A. Kozma, J. L. Walker, H. M. Jones, and E. C. Poggio, "Developments in radar imaging," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-20, no. 4, pp. 363–399, July 1984.

[13] M. Soumekh, "Automatic aircraft landing using interferometric inverse synthetic aperture radar imaging," *IEEE Transactions on Image Processing*, vol. 5, no. 9, pp. 1335–1345, September 1996.

[14] T. Sauer, K. H. Bethke, F. Buettner, B. Roede, and A. Schroth, "Imaging of commercial aircraft by inverse synthetic aperture radar and their classification in a near-range radar network (NRN)," *Proceedings of the IEEE National Radar Conference*, pp. 19–24, May 13-15, 1997, Syracuse, NY, USA.

[15] R. Klemm, "Current trends in SAR technology," *IEEE Aerospace and Electronic Systems Magazine*, vol. 12, no. 3, pp. 3–8, March 1997.

[16] P. Ling, G. Lu, and H. Huan, "C-band inverse synthetic radar system," *Proceedings of the CIE (Chinese Institute of Electronics) International Conference on Radar*, pp. 250–253, October 8-10, 1996, Beijing, China.

[17] V. C. Chen and W. J. Miceli, "Time-varying spectral analysis for radar imaging of maneuvering targets," *IEE Proceedings-Radar, Sonar, and Navigation*, vol. 145, no. 5, pp. 262–268, October 1998.

[18] S. R. DeGraaf, "SAR imaging via modern 2-D spectral estimation methods," *IEEE Transactions on Image Processing*, vol. 7, pp. 729–761, May 1998.

[19] G. R. Benitz, "High-definition vector imaging," *MIT Lincoln Laboratory Journal-Special Issue on Superresolution*, vol. 10, no. 2, pp. 147–170, 1997.

[20] Z. Bi, J. Li, and Z.-S. Liu, "Super resolution SAR image formation via parametric spectral estimation methods," *to appear in IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 1, January 1999.

[21] B. D. Steinberg, D. L. Carlson, and W. Lee, "Microwave imaging of aircraft," *Proceedings of the IEEE*, vol. 76, pp. 1578–1592, December 1988.

[22] S. Werness, W. Carrara, L. Joyce, and D. Franczak, "Moving target imaging algorithm for SAR data," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 1, pp. 57–67, January 1994.

[23] E. H. Attia, "Self-cohering airborne distributed arrays using the robust minimum variance algorithm," *IEEE AP-S Digest*, pp. 603–606, June 1986.

[24] B. Haywood and R. J. Evans, "Motion compensation for ISAR imaging," *Australian Symposium on Signal Processing and Applications*, pp. 112–117, April 17-19, 1989, Adelaide, Australia.

[25] B. Kang, H. M. Subbaram, and B. D. Steinberg, "Improved adaptive beamforming targer for self-calibrating a distorted phase array," *IEEE Transactions on Antennas and Propagation*, vol. 38, no. 2, pp. 186–194, February 1990.

[26] H. Wu, G. Y. Delisle, and D. G. Fang, "Translational motion compensation in ISAR image processing," *IEEE Transactions on Image Processing*, vol. 4, no. 11, pp. 1561–1571, November 1995.

[27] Z. Bao, W. Deng, and J. Yang, "A new motion compensation scheme for ISAR imaging," *Proceedings of the 1991 CIE (Chinese Institute of Electronics) International Conference on Radar*, pp. 327–330, October 1991, Beijing, China.

[28] M. J. Pricket and C. C. Chen, "Principle of inverse synthetic radar (ISAR) imaging," *IEEE Electronics and Aerospace Systems Conference Record*, pp. 340–345, 1980, Arlington, VA, USA.

[29] T. Itoh, H. Sueda, and Y. Watanabe, "Motion compensation for ISAR via centroid tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 3, pp. 1192–1197, July 1996.

[30] D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and C. V. Jakowatz, Jr., "Phase gradient autofocus-a robust tool for high resolution SAR phase correction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, pp. 827–835, July 1994.

[31] C. V. Jakowatz, Jr., D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. A. Thompson, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach.* Norwell, MA: Kluwer Academic Publishers, 1996.

[32] K. K. Eerland, "Application of inverse synthetic aperture radar on aircraft," *Proceedings of the International Conference on Radar*, pp. 618–623, 1984, Paris, France.

[33] Y. Wang, H. Ling, and V. C. Chen, "ISAR motion compensation via adaptive joint time-frequency technique," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, pp. 670–677, April 1998.

[34] B. C. Flores, S. Tariq, and J. S. Son, "Image focus quality indicators for efficient inverse synthetic aperture radar phase correction," *Proceedings of SPIE: Algorithms for Synthetic Aperture Radar Imagery V*, vol. 2757, pp. 2–13, April 1996, Orlando, FL, USA.

[35] F. Berizzi and G. Corsini, "Autofocusing of inverse synthetic aperture radar images using contrast optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, pp. 1185–1191, July 1996.

[36] J. Li and P. Stoica, "Efficient mixed-spectrum estimation with applications to target feature extraction," *IEEE Transactions on Signal Processing*, vol. 44, pp. 281–295, February 1996.

[37] J. A. Högbom, "Aperture synthesis with a non-regular distribution of interferometer baselines," *Astronomy and Astrophysics Supplements*, vol. 15, pp. 417–426, 1974.

[38] *TMC2310*, TRW LSI Products, Inc.

[39] *A Users' Guide for the A41102*, Austeck Microsystems, 1988.

[40] *TM-66 swiFFT chip*, Texas Memory Systems, Inc., 1998.

[41] J. Li and V. C. Chen, "A robust autofocusing algorithm for ISAR imaging," submitted to *NATO Sensors and Electronics Technology Panel Symposium on High Resolution Radar Techniques*, March 1999, Seville, Spain.

[42] R. C. Hansen, "Array pattern control and synthesis," *Proceedings of the IEEE*, vol. 80, no. 1, pp. 141–151, January 1992.

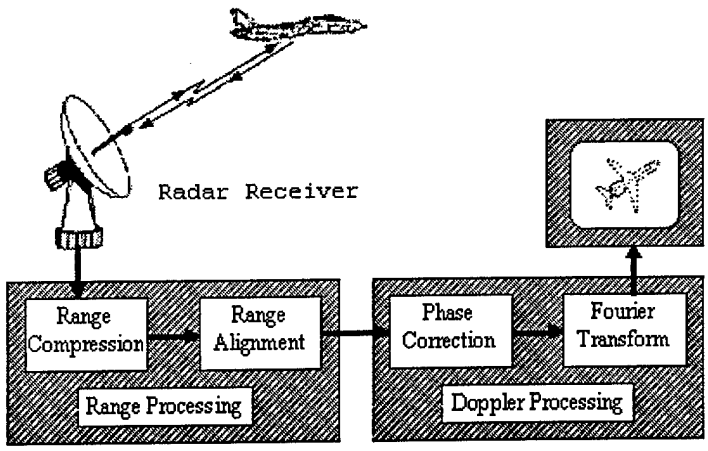


Figure 2.1: Diagram of a conventional ISAR imaging system.

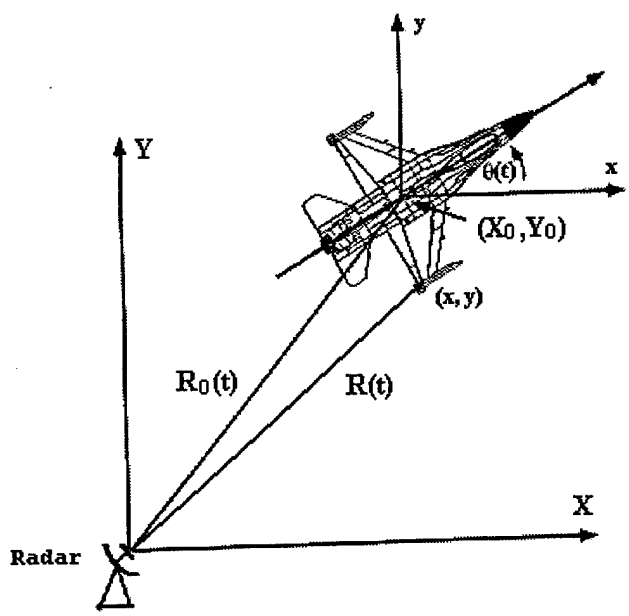


Figure 2.2: A geometry of the radar and the target.

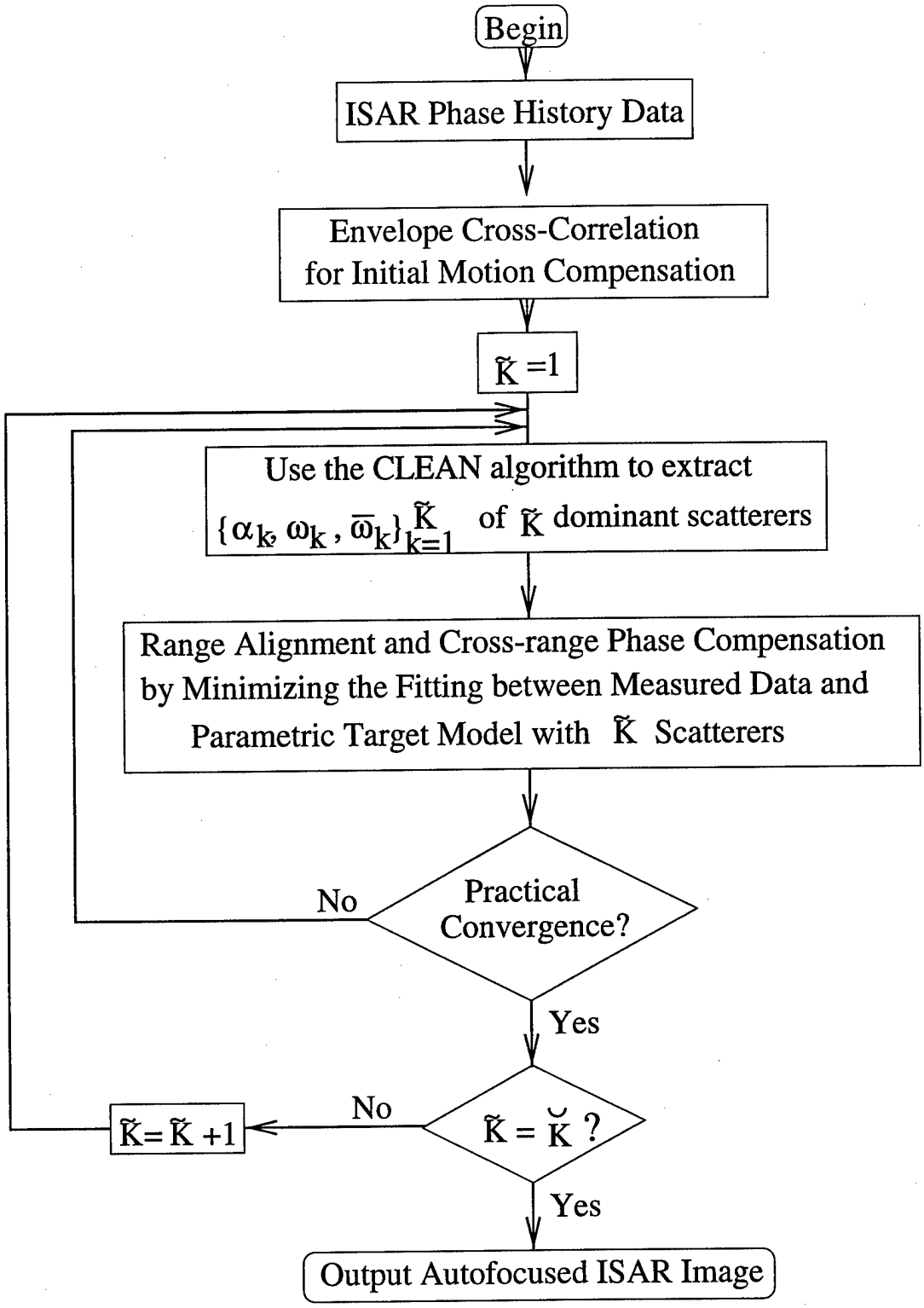


Figure 2.3: Flow chart of the AUTOCLEAN algorithm.

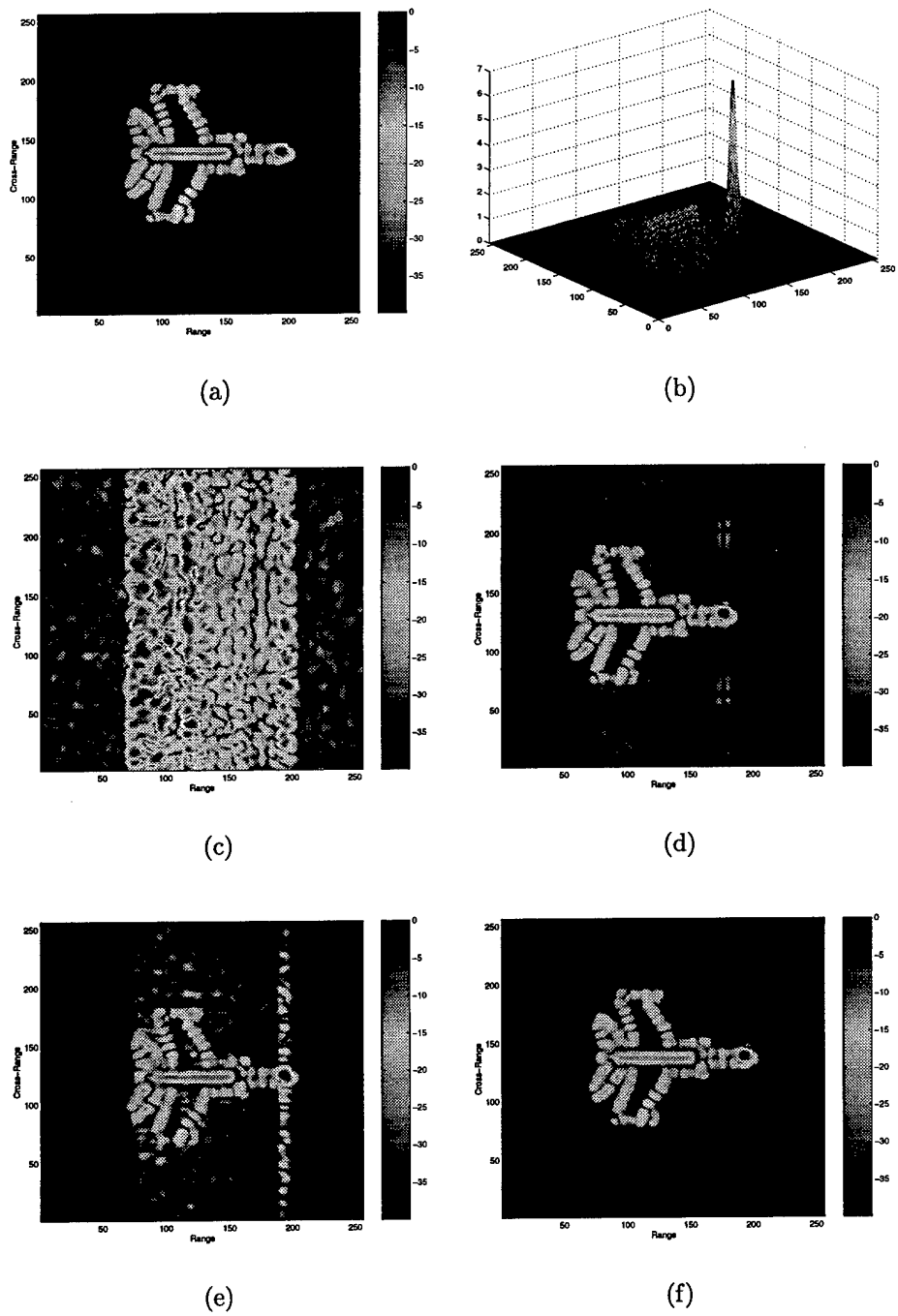


Figure 2.4: ISAR images of a simulated MIG-25 aircraft with one very strong isolated dominant scatterer. (a) Original 2-D windowed FFT image. (b) Mesh plot of the original 2-D windowed FFT image. (c) Windowed FFT image before motion compensation. (d) Windowed FFT image after autofocusing using Haywood and Evans' MSA based on a single dominant scatterer. (e) Windowed FFT image after autofocusing using PGA. (f) Windowed FFT image after autofocusing using AUTOCLEAN based on a single dominant scatterer.

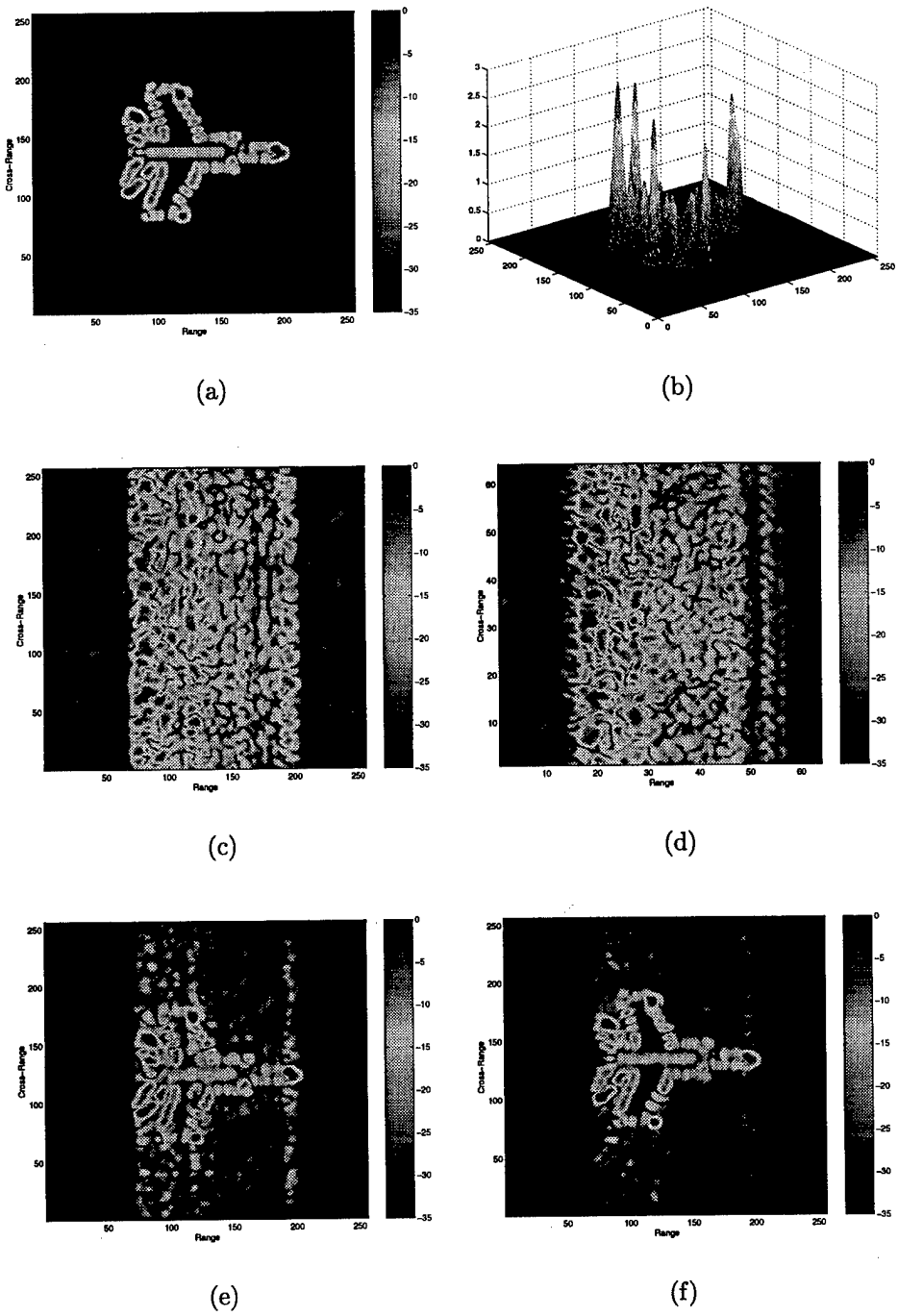


Figure 2.5: ISAR images of a simulated MIG-25 aircraft with no very strong isolated dominant scatterers. (a) Original 2-D windowed FFT image. (b) Mesh plot of the original 2-D windowed FFT image. (c) Windowed FFT image before motion compensation. (d) Windowed FFT image after autofocusing using Haywood and Evans' MSA based on $\breve{K} = 5$ dominant scatterers. (e) Windowed FFT image after autofocusing using PGA. (f) Windowed FFT image after autofocusing using AUTOCLEAN based on $\breve{K} = 5$ dominant scatterers.

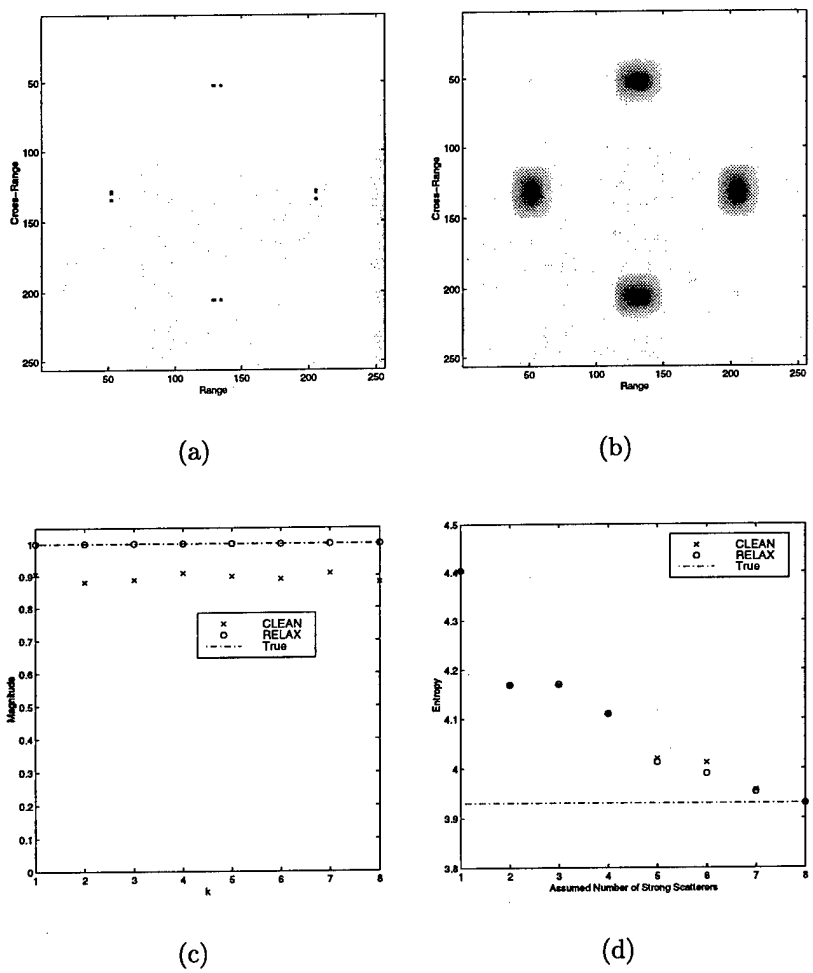


Figure 2.6: Comparison of CLEAN and RELAX. (a) The true image. (b) Original windowed FFT image of the simulated target. (c) Comparison of amplitude estimation accuracy. (d) Comparison of image focus quality (entropy).

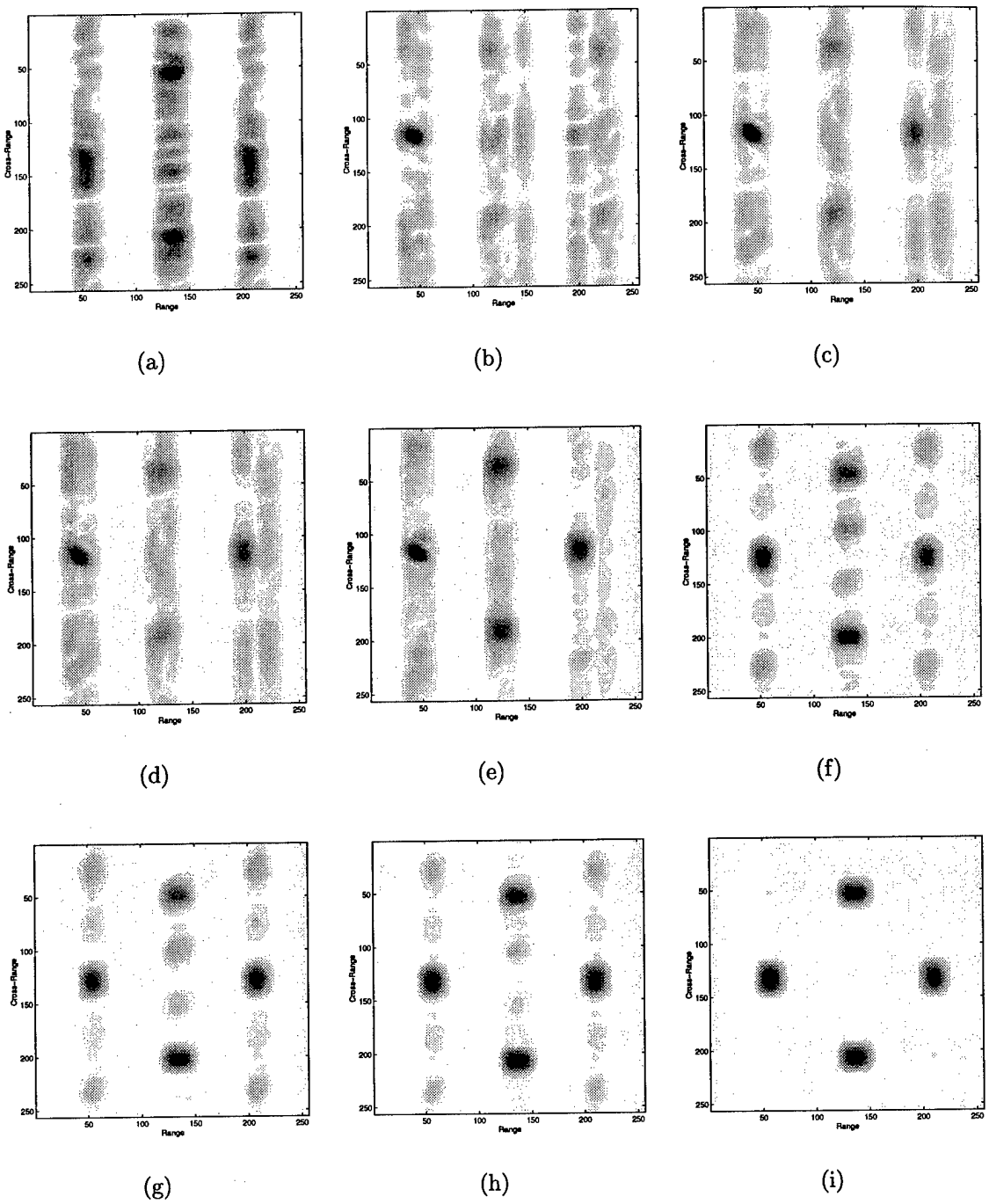


Figure 2.7: Performance of AUTOCLEAN as $\breve{K}$ varies. (a) Windowed FFT image after initial motion compensation by using the envelope cross-correlation method. (b)-(i): Windowed FFT images after autofocusing via AUTOCLEAN by increasing $\breve{K}$ from 1 to 8.

3. Complex ISAR Imaging of Maneuvering Targets via the Capon Estimator

3.1 Introduction

Synthetic aperture radar (SAR) and inverse synthetic aperture radar (ISAR) [1, 2] image formation and target feature extraction are becoming increasingly important in many civilian and military applications [3, 4]. In SAR/ISAR imaging, the high resolution in range is obtained by transmitting a signal with a large bandwidth, whereas the high resolution in cross-range can be achieved by utilizing the relative motion between the target and the radar to form a large synthetic aperture. SAR and ISAR are used to produce high resolution images of stationary objects and moving targets, respectively. The principle underlying SAR (especially spotlight-mode SAR) and ISAR is extremely similar. The major difference between SAR and ISAR lies in the nature of the relative motion. In SAR, the radar is moving while the object to be imaged is stationary and hence the relative motion is cooperative. This situation is reversed in ISAR. In ISAR, the radar is stationary (or moving) while the target to be imaged is moving in a noncooperative way. This noncooperative motion makes the ISAR imaging more difficult than the SAR imaging. Because of this, although SAR technology is well established and nearly 30 spaceborne and airborne SAR systems are currently in operation [3], ISAR technology is still at the R&D stage and only a few experimental systems are reported under development in the literature [5].

High resolution *complex* SAR/ISAR image formation are very important for improving the automatic target recognition (ATR) performance. The conventional image formation algorithms based on Fourier transform are known to be robust but suffer from poor resolution and accuracy and high sidelobes. Many modern spectral estimation techniques have been devised and applied to SAR/ISAR image formation to improve resolution and accuracy and reduce sidelobes. In [6, 7], many parametric and nonparametric spectral estimation methods are compared and discussed for their merits for SAR/ISAR imaging. Many of these methods are used for high resolution intensity image formation or power spectral estimation. Recently, *complex* image formation and analysis have attracted a lot of attention since extracting the

desired information about targets from radar returns requires using both the amplitude and phase of the complex image [8].

The well-known Capon method [9] was proposed to estimate frequency-wavenumber power spectral density. Since then, its principle has been widely applied to spectral estimation, direction-of-arrival estimation, and adaptive temporal/spatial/spatial-temporal filtering. It can also be used for complex spectral estimation. It has been shown recently in [10] that the nonparametric Capon estimator belongs to the class of matched-filterbank spectral estimators. Capon and its reduced-rank variations [11, 12] can produce high resolution SAR images with low sidelobes and reduced speckles, which can be used to significantly improve the ATR performance [13, 14]. Moreover, since the Capon method is nonparametric, it is more robust against data mismodeling errors than parametric algorithms.

Most ISAR imaging algorithms (including the aforementioned Capon) are based on the range-Doppler processing, which implies that the Doppler shifts nearly remain constant during the coherent integration time. Unfortunately, for maneuvering targets, this assumption is more often than not violated. This observation has motivated the use of various time-varying spectral analysis methods on ISAR imaging of maneuvering targets [15]. In [15], the popular Wigner-Ville Distribution (WVD) method [16] is applied for ISAR imaging of maneuvering targets. However, quadratic time-frequency analysis algorithms including WVD *cannot* be used for the *complex* image formation problem considered herein. Another popular approach is the Short-Time Fourier Transform (STFT) method [16], which is a linear time-frequency signal representation method and can be used for complex image formation. Major drawbacks associated with the STFT method include poor resolution and accuracy and high sidelobes.

In this chapter, we present an adaptive Capon spectral estimation algorithm for the complex ISAR image formation of maneuvering targets. It has better resolution and lower sidelobes than the STFT method. The algorithm is an efficient recursive implementation of the two-dimensional (2-D) Capon complex spectral estimator, which involves only FFT and simple matrix operations.

The reminder of this chapter is organized as follows. The problem of interest is formu-

lated in Section 3.2. In Sections 3.3 and 3.4, we present the recursive forward-only (F-O) and forward-backward (F-B) Capon algorithms, respectively. ISAR imaging examples are provided in Section 3.5 to illustrate the performance of the proposed methods. Finally, Section 3.6 concludes this chapter.

3.2 Problem Formulation

For maneuvering targets with nonuniform rotational motion, the Doppler shift corresponding to the cross-range of each scatterer is time-varying [15]. In this case, the direct application of spectral estimation methods will produce blurred images. To mitigate this problem, we can apply the STFT method to produce time-varying complex ISAR images. However, the STFT method suffers from poor resolution and accuracy and high sidelobes. In this paper, we will present an adaptive Capon algorithm for the complex ISAR imaging of maneuvering targets. Similar to STFT, we will use a sliding rectangular window along the cross-range dimension and the observed signal within the sliding window is assumed to be approximately stationary. Then the Capon method is applied to the data within the window to produce high resolution ISAR images. As the window slides along the cross-range dimension, time-varying high resolution ISAR images can be obtained. This approach overcomes the drawbacks of the STFT method and preserves the good features of the Capon estimator, including high resolution, low sidelobes, reduced speckles, and robustness against data mismodeling errors.

Let N denote the number of range samples and $\bar{N}$ denote the sliding window length along the cross-range dimension. Let $\{z_{n,t-\bar{n}}, n = 0, 1, \cdots, N-1, \bar{n} = 0, 1, \cdots, \bar{N}-1\}$ denote the phase history of a maneuvering target of interest within the sliding window ended at position t in cross-range. For a frequency pair $(\omega, \bar{\omega})$ of interest, which are proportional to range and cross-range, respectively, $z_{n,t-\bar{n}}$ can be written as

$$z_{n,t-\bar{n}} = \alpha_t(\omega, \bar{\omega}) e^{j\{n\omega + (\bar{N}-1-\bar{n})\bar{\omega}\}} + e_{n,t-\bar{n}}(\omega, \bar{\omega}), \quad n = 0, 1, \cdots, N-1, \bar{n} = 0, 1, \cdots, \bar{N}-1, \tag{3.1}$$

where $\alpha_t(\omega, \bar{\omega})$ denotes the time-varying complex amplitude of a 2-D sinusoid with frequency

$(\omega, \bar{\omega})$ and $e_{n,t-\bar{n}}(\omega, \bar{\omega})$ denotes the unmodeled noise and interference at frequency $(\omega, \bar{\omega})$. Note that $\alpha_t(\omega, \bar{\omega})$ is assumed to change little within the sliding window so that $\{z_{n,t-\bar{n}}, n = 0, 1, \cdots, N-1, \bar{n} = 0, 1, \cdots, \bar{N}-1\}$ is approximately stationary.

Our problem of interest herein is to obtain the estimate of $\alpha_t(\omega, \bar{\omega})$ from the 2-D data sequence $\{z_{n,t-\bar{n}}, n = 0, 1, \cdots, N-1, \bar{n} = 0, 1, \cdots, \bar{N}-1\}$ for all $(\omega, \bar{\omega})$ of interest as t varies. More specifically, we want to devise an efficient recursive implementation of the 2-D Capon method for this time-varying complex spectral estimation problem.

Both the nonparametric F-O and F-B Capon algorithms can be used for complex spectral estimation. The former uses only the forward sample covariance matrix to calculate the finite impulse response (FIR) filters adaptively while both the forward and backward sample covariance matrices are used by the latter. It has been shown in [10] that F-B Capon outperforms F-O Capon with smaller biases. However, for the recursive implementation, the latter is computationally more efficient than the former. In the following two sections, we will present the recursive implementations for both of them.

3.3 Forward-Only Capon

In this section, we first briefly review the 2-D F-O Capon algorithm. Then an efficient batch-mode implementation scheme is presented for initialization. Next, we derive the recursive F-O Capon algorithm. Finally, we analyze the computational complexity of the proposed implementation.

3.3.1 Brief Review of F-O Capon

Capon is a *nonparametric* adaptive matched-filterbank approach [17, 10]. It follows two main steps: (a) pass the data $\{z_{n,t-\bar{n}}, n = 0, 1, \cdots, N-1, \bar{n} = 0, 1, \cdots, \bar{N}-1\}$ through a 2-D bandpass filter with varying center frequency $(\omega, \bar{\omega})$; (b) obtain the estimates $\hat{\alpha}_t(\omega, \bar{\omega})$ of $\alpha_t(\omega, \bar{\omega})$, for all $\omega \in [0, 2\pi)$ and $\bar{\omega} \in [0, 2\pi)$ of interest, from the filtered data. Assume

that the bandpass filter $\mathbf{H}_t(\omega,\bar{\omega})$ used is an $(M \times \bar{M})$-tap 2-D FIR filter having the form

$$\mathbf{H}_t(\omega,\bar{\omega}) = \begin{bmatrix} h_{0,0,t}(\omega,\bar{\omega}) & h_{0,1,t}(\omega,\bar{\omega}) & \cdots & h_{0,\bar{M}-1,t}(\omega,\bar{\omega}) \\ h_{1,0,t}(\omega,\bar{\omega}) & h_{1,1,t}(\omega,\bar{\omega}) & \cdots & h_{1,\bar{M}-1,t}(\omega,\bar{\omega}) \\ \vdots & \vdots & \ddots & \vdots \\ h_{M-1,0,t}(\omega,\bar{\omega}) & h_{M-1,1,t}(\omega,\bar{\omega}) & \cdots & h_{M-1,\bar{M}-1,t}(\omega,\bar{\omega}) \end{bmatrix}. \tag{3.2}$$

Let

$$\mathbf{Z}_{l,t-\bar{l}} = \begin{bmatrix} z_{l,t-\bar{l}-\bar{M}+1} & z_{l,t-\bar{l}-\bar{M}+2} & \cdots & z_{l,t-\bar{l}} \\ z_{l+1,t-\bar{l}-\bar{M}+1} & z_{l+1,t-\bar{l}-\bar{M}+2} & \cdots & z_{l+1,t-\bar{l}} \\ \vdots & \vdots & \ddots & \vdots \\ z_{l+M-1,t-\bar{l}-\bar{M}+1} & z_{l+M-1,t-\bar{l}-\bar{M}+2} & \cdots & z_{l+M-1,t-\bar{l}} \end{bmatrix},$$
$$(\quad l = 0,1,\cdots,L-1,\ \bar{l} = 0,1,\cdots,\bar{L}-1) \tag{3.3}$$

denote the $(l,\bar{l})$th $M \times \bar{M}$ forward data matrix constructed from the data sequence $\{z_{n,t-\bar{n}}, n = 0,1,\cdots,N-1, \bar{n} = 0,1,\cdots,\bar{N}-1\}$, where $L = N - M + 1,\ \bar{L} = \bar{N} - \bar{M} + 1$. Define

$$\mathbf{h}_t(\omega,\bar{\omega}) = \text{vec}\left\{\mathbf{H}_t(\omega,\bar{\omega})\right\}, \tag{3.4}$$

and

$$\mathbf{z}_{l,t-\bar{l}} = \text{vec}\left\{\mathbf{Z}_{l,t-\bar{l}}\right\}, \tag{3.5}$$

where $\text{vec}\{\mathbf{X}\} = \begin{bmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T & \cdots & \mathbf{x}_K^T \end{bmatrix}^T$ with $\mathbf{x}_k$ being the kth column of $\mathbf{X}$ and $(\cdot)^T$ denoting the transpose. Let

$$\mathbf{P}_t = \sum_{\bar{l}=0}^{\bar{L}-1} \sum_{l=0}^{L-1} \mathbf{z}_{l,t-\bar{l}} \mathbf{z}_{l,t-\bar{l}}^H, \tag{3.6}$$

denote the forward covariance matrix, where $(\cdot)^H$ denotes the conjugate transpose, and

$$\mathbf{a}_{M,\bar{M}}(\omega,\bar{\omega}) = \mathbf{a}_{\bar{M}}(\bar{\omega}) \otimes \mathbf{a}_M(\omega), \tag{3.7}$$

with $\otimes$ denoting the Kronecker product [18],

$$\mathbf{a}_M(\omega) = \begin{bmatrix} 1 & e^{j\omega} & \cdots & e^{j(M-1)\omega} \end{bmatrix}^T, \tag{3.8}$$

and

$$\mathbf{a}_{\bar{M}}(\bar{\omega}) = \begin{bmatrix} 1 & e^{j\bar{\omega}} & \cdots & e^{j(\bar{M}-1)\bar{\omega}} \end{bmatrix}^T. \tag{3.9}$$

The adaptive Capon filter passes the frequency $(\omega, \bar{\omega})$ without any attenuation, i.e., we let $\mathbf{h}_t^H(\omega, \bar{\omega})\mathbf{a}_{M,\bar{M}}(\omega, \bar{\omega}) = 1$ and minimize the output power $\mathbf{h}_t^H(\omega, \bar{\omega})\mathbf{P}_t\mathbf{h}_t(\omega, \bar{\omega})$. The F-O adaptive Capon filter $\mathbf{h}_{\mathrm{FO},t}(\omega, \bar{\omega})$ has the form [10]

$$\mathbf{h}_{\mathrm{FO},t}(\omega, \bar{\omega}) = \frac{\mathbf{P}_t^{-1}\mathbf{a}_{M,\bar{M}}(\omega, \bar{\omega})}{\mathbf{a}_{M,\bar{M}}^H(\omega, \bar{\omega})\mathbf{P}_t^{-1}\mathbf{a}_{M,\bar{M}}(\omega, \bar{\omega})}. \tag{3.10}$$

The F-O Capon estimate of the complex amplitude $\alpha_t(\omega, \bar{\omega})$ is given by [10]

$$\hat{\alpha}_{\mathrm{FO},t}(\omega, \bar{\omega}) = \frac{\mathbf{a}_{M,\bar{M}}^H(\omega, \bar{\omega})\mathbf{P}_t^{-1}\left[\sum_{\bar{l}=0}^{\bar{L}-1}\sum_{l=0}^{L-1}\mathbf{z}_{l,t-\bar{l}}e^{-j\{l\omega+(\bar{L}-1-\bar{l})\bar{\omega}\}}\right]}{\bar{L}L\mathbf{a}_{M,\bar{M}}^H(\omega, \bar{\omega})\mathbf{P}_t^{-1}\mathbf{a}_{M,\bar{M}}(\omega, \bar{\omega})}. \tag{3.11}$$

3.3.2 Efficient Initialization Algorithm

Direct implementation of the 2-D Capon estimator in (3.11) is computationally demanding since it involves the inversion of the large dimension matrix $\mathbf{P}_t$ and the complex amplitudes $\alpha_t(\omega, \bar{\omega})$ at different frequency points $(\omega, \bar{\omega})$ are calculated independently. Next, we present an efficient batch-mode implementation algorithm, which can be used as the initialization step of the proposed recursive Capon algorithm.

Let

$$\mathbf{Z}_t = \begin{bmatrix} \mathbf{z}_{0,t-\bar{L}+1} & \cdots & \mathbf{z}_{L-1,t-\bar{L}+1} & \cdots & \mathbf{z}_{0,t} & \cdots & \mathbf{z}_{L-1,t} \end{bmatrix}, \tag{3.12}$$

denote the data matrix formed from the forward data vectors. Then (3.11) can be rewritten as

$$\begin{aligned} \hat{\alpha}_{\mathrm{FO},t}(\omega, \bar{\omega}) &= \frac{\mathbf{a}_{M,\bar{M}}^H(\omega, \bar{\omega})\mathbf{P}_t^{-1}\mathbf{Z}_t\mathbf{a}_{L,\bar{L}}^*(\omega, \bar{\omega})}{\bar{L}L\mathbf{a}_{M,\bar{M}}^H(\omega, \bar{\omega})\mathbf{P}_t^{-1}\mathbf{a}_{M,\bar{M}}(\omega, \bar{\omega})}, \\ &\triangleq \frac{c_t(\omega, \bar{\omega})}{\bar{L}Ld_t(\omega, \bar{\omega})}, \end{aligned} \tag{3.13}$$

where

$$c_t(\omega, \bar{\omega}) \triangleq \mathbf{a}_{M,\bar{M}}^H(\omega, \bar{\omega})\mathbf{P}_t^{-1}\mathbf{Z}_t\mathbf{a}_{L,\bar{L}}^*(\omega, \bar{\omega}), \tag{3.14}$$

and

$$d_t(\omega,\bar{\omega}) \triangleq \mathbf{a}_{M,\bar{M}}^H(\omega,\bar{\omega})\mathbf{P}_t^{-1}\mathbf{a}_{M,\bar{M}}(\omega,\bar{\omega}). \tag{3.15}$$

Direct calculation of $c_t(\omega,\bar{\omega})$, $d_t(\omega,\bar{\omega})$, and hence $\hat{\alpha}_{\mathrm{FO},t}(\omega,\bar{\omega})$ is computationally intensive. Below we present an efficient FFT based implementation that can be used to significantly reduce the amount of computations.

Let $\mathbf{D}_t$ be the Hermitian square root of $\mathbf{P}_t^{-1}$ [19], i.e.,

$$\mathbf{P}_t^{-1} = \mathbf{D}_t\mathbf{D}_t^H. \tag{3.16}$$

Then $c_t(\omega,\bar{\omega})$ and $d_t(\omega,\bar{\omega})$ can be computed in the following way:

$$c_t(\omega,\bar{\omega}) = \mathbf{F}_\alpha(\omega,\bar{\omega})\mathbf{F}_\beta(\omega,\bar{\omega}), \tag{3.17}$$

and

$$d_t(\omega,\bar{\omega}) = \|\mathbf{F}_\alpha(\omega,\bar{\omega})\|^2, \tag{3.18}$$

where

$$\mathbf{F}_\alpha(\omega,\bar{\omega}) = \mathbf{a}_{M,\bar{M}}^H(\omega,\bar{\omega})\mathbf{D}_t, \tag{3.19}$$

and

$$\mathbf{F}_\beta(\omega,\bar{\omega}) = \left(\mathbf{D}_t^H\mathbf{Z}_t\right)\mathbf{a}_{L,\bar{L}}^*(\omega,\bar{\omega}). \tag{3.20}$$

Note that $\mathbf{F}_\alpha(\omega,\bar{\omega})$ and $\mathbf{F}_\beta(\omega,\bar{\omega})$ can be calculated efficiently via 2-D FFT. By applying FFT whenever appropriate, significant computational savings can be achieved in constrast to the direct implementation. Quantitative results on the speedups will be given in Subsection 3.3.4.

3.3.3 Efficient Recursive Algorithm

After initialization, $\hat{\alpha}_{\mathrm{FO},t+1}(\omega,\bar{\omega})$ can be updated from $\hat{\alpha}_{\mathrm{FO},t}(\omega,\bar{\omega})$ by updating $c_{t+1}(\omega,\bar{\omega})$ and $d_{t+1}(\omega,\bar{\omega})$, respectively, from $c_t(\omega,\bar{\omega})$ and $d_t(\omega,\bar{\omega})$. Our problem, like the windowed least squares estimation problem [20], requires both updating and downdating. (Updating and downdating are used to adjust the solution to the problem of interest when a new observation is added (updating) or an old observation is deleted (downdating).) We have found that

significant computational savings can be achieved by applying the Matrix Inversion Lemma [21], which has been widely used in adaptive filtering [22] and recursive system identification [23], to update and downdate $\mathbf{P}_{t+1}^{-1}$ from $\mathbf{P}_t^{-1}$. One major feature of our algorithm is to apply FFT whenever appropriate to further reduce the amount of computations. Below we present our recursive algorithm.

Let $\bar{c}_{t+1}(\omega,\bar{\omega})$ and $\bar{d}_{t+1}(\omega,\bar{\omega})$ be defined similarly to $c_{t+1}(\omega,\bar{\omega})$ and $d_{t+1}(\omega,\bar{\omega})$ in (3.14) and (3.15), respectively, except that $\{z_{n,t+1-\bar{n}}, n=0,1\cdots,N-1,\bar{n}=0,1,\cdots,\bar{N}-1\}$ is replaced by an intermediate data sequence $\{z_{n,t+1-\bar{n}}, n=0,1\cdots,N-1,\bar{n}=0,1,\cdots,\bar{N}\}$, which is obtained by adding a new observation vector $\{z_{n,t+1}, n=0,1\cdots,N-1\}$ to the former data. Then the recursive algorithm can be summarized into two steps: (a) update $\bar{c}_{t+1}(\omega,\bar{\omega})$ and $\bar{d}_{t+1}(\omega,\bar{\omega})$ from $c_t(\omega,\bar{\omega})$ and $d_t(\omega,\bar{\omega})$, respectively, from the intermediate data sequence; (b) downdate $c_{t+1}(\omega,\bar{\omega})$ and $d_{t+1}(\omega,\bar{\omega})$, respectively, from $\bar{c}_{t+1}(\omega,\bar{\omega})$ and $\bar{d}_{t+1}(\omega,\bar{\omega})$ by removing the oldest observation vector $\{z_{n,t-\bar{N}+2}, n=0,1\cdots,N-1\}$ from the intermediate data sequence. Hence the Matrix Inversion Lemma is applied twice for updating $\hat{\alpha}_{\mathrm{FO},t+1}(\omega,\bar{\omega})$ from $\hat{\alpha}_{\mathrm{FO},t}(\omega,\bar{\omega})$.

We first consider how to update $\bar{c}_{t+1}(\omega,\bar{\omega})$ and $\bar{d}_{t+1}(\omega,\bar{\omega})$ from $c_t(\omega,\bar{\omega})$ and $d_t(\omega,\bar{\omega})$, respectively.

Let

$$\tilde{\mathbf{Z}}_{t+1} = \begin{bmatrix} \mathbf{z}_{0,t+1} & \mathbf{z}_{1,t+1} & \cdots & \mathbf{z}_{L-1,t+1} \end{bmatrix}, \tag{3.21}$$

$$\bar{\mathbf{Z}}_{t+1} = \begin{bmatrix} \mathbf{Z}_t & \tilde{\mathbf{Z}}_{t+1} \end{bmatrix}, \tag{3.22}$$

$$\bar{\mathbf{P}}_{t+1} = \bar{\mathbf{Z}}_{t+1}\bar{\mathbf{Z}}_{t+1}^H = \mathbf{P}_t + \tilde{\mathbf{Z}}_{t+1}\tilde{\mathbf{Z}}_{t+1}^H, \tag{3.23}$$

and let the $L \times L$ matrix $\mathbf{G}_{t+1}$ be the Hermitian square root of $\left(\mathbf{I} + \tilde{\mathbf{Z}}_{t+1}^H \mathbf{P}_t^{-1} \tilde{\mathbf{Z}}_{t+1}\right)^{-1}$, i.e.,

$$\mathbf{G}_{t+1}\mathbf{G}_{t+1}^H = \left(\mathbf{I} + \tilde{\mathbf{Z}}_{t+1}^H \mathbf{P}_t^{-1} \tilde{\mathbf{Z}}_{t+1}\right)^{-1}. \tag{3.24}$$

By using the Matrix Inversion Lemma, $\bar{\mathbf{P}}_{t+1}^{-1}$ can be updated from $\mathbf{P}_t$ as follows

$$\bar{\mathbf{P}}_{t+1}^{-1} = \mathbf{P}_t^{-1} - \mathbf{\Phi}_{t+1}\mathbf{\Phi}_{t+1}^H, \tag{3.25}$$

where

$$\mathbf{\Phi}_{t+1} = \mathbf{P}_t^{-1}\tilde{\mathbf{Z}}_{t+1}\mathbf{G}_{t+1}. \tag{3.26}$$

Let

$$\mathbf{f}_\Phi(\omega,\bar{\omega}) = \mathbf{\Phi}_{t+1}^T\mathbf{a}_{M,\bar{M}}^*(\omega,\bar{\omega}), \tag{3.27}$$

$$\mathbf{f}_\Psi(\omega,\bar{\omega}) = (\mathbf{\Phi}_{t+1}^H\mathbf{Z}_t)\mathbf{a}_{L,\bar{L}}^*(\omega,\bar{\omega}), \tag{3.28}$$

and

$$\mathbf{f}_G(\omega) = \mathbf{G}_{t+1}^H\mathbf{a}_L^*(\omega). \tag{3.29}$$

Then it follows that

$$\begin{aligned}\bar{c}_{t+1}(\omega,\bar{\omega}) &\triangleq \mathbf{a}_{M,\bar{M}}^H(\omega,\bar{\omega})\bar{\mathbf{P}}_{t+1}^{-1}\bar{\mathbf{Z}}_{t+1}\mathbf{a}_{L,\bar{L}+1}^*(\omega,\bar{\omega}),\\ &= c_t(\omega,\bar{\omega}) + \mathbf{f}_\Phi^T(\omega,\bar{\omega})\left\{\mathbf{f}_G(\omega)e^{-j\bar{L}\bar{\omega}} - \mathbf{f}_\Psi(\omega,\bar{\omega})\right\},\end{aligned} \tag{3.30}$$

and

$$\begin{aligned}\bar{d}_{t+1}(\omega,\bar{\omega}) &\triangleq \mathbf{a}_{M,\bar{M}}^H(\omega,\bar{\omega})\bar{\mathbf{P}}_{t+1}^{-1}\mathbf{a}_{M,\bar{M}}(\omega,\bar{\omega}),\\ &= d_t(\omega,\bar{\omega}) - \|\mathbf{f}_\Phi(\omega,\bar{\omega})\|^2,\end{aligned} \tag{3.31}$$

where $\|\cdot\|$ denotes the Euclidean norm. Note that $\mathbf{f}_\Phi(\omega,\bar{\omega})$, $\mathbf{f}_\Psi(\omega,\bar{\omega})$, and $\mathbf{f}_G(\omega)$ in (3.30) and (3.31) can be calculated efficiently via FFT.

Now consider downdating $c_{t+1}(\omega,\bar{\omega})$ and $d_{t+1}(\omega,\bar{\omega})$, respectively, from $\bar{c}_{t+1}(\omega,\bar{\omega})$ and $\bar{d}_{t+1}(\omega,\bar{\omega})$. The derivation is very similar to the previous updating process and the downdating process is outlined as follows. From

$$\mathbf{P}_{t+1} = \bar{\mathbf{P}}_{t+1} - \tilde{\mathbf{Z}}_{t-\bar{L}+1}\tilde{\mathbf{Z}}_{t-\bar{L}+1}^H, \tag{3.32}$$

and using the Matrix Inversion Lemma once again we have

$$\mathbf{P}_{t+1}^{-1} = \bar{\mathbf{P}}_{t+1}^{-1} + \bar{\mathbf{\Phi}}_{t-\bar{L}+1}\bar{\mathbf{\Phi}}_{t-\bar{L}+1}^H, \tag{3.33}$$

where

$$\bar{\mathbf{\Phi}}_{t-\bar{L}+1} = \bar{\mathbf{P}}_{t+1}^{-1}\tilde{\mathbf{Z}}_{t-\bar{L}+1}\bar{\mathbf{G}}_{t-\bar{L}+1}, \tag{3.34}$$

with the $L \times L$ matrix $\bar{\mathbf{G}}_{t-\bar{L}+1}$ satisfying

$$\bar{\mathbf{G}}_{t-\bar{L}+1}\bar{\mathbf{G}}^H_{t-\bar{L}+1} = \left(\mathbf{I} - \tilde{\mathbf{Z}}^H_{t-\bar{L}+1}\bar{\mathbf{P}}^{-1}_{t+1}\tilde{\mathbf{Z}}_{t-\bar{L}+1}\right)^{-1}. \tag{3.35}$$

Hence $c_{t+1}(\omega,\bar{\omega})$ and $d_{t+1}(\omega,\bar{\omega})$ can be downdated using the following equations

$$c_{t+1}(\omega,\bar{\omega}) = \left\{\bar{c}_{t+1}(\omega,\bar{\omega}) - \mathbf{f}^T_{\bar{\Phi}}(\omega,\bar{\omega})\left[\mathbf{f}_{\bar{G}}(\omega) - \mathbf{f}_{\bar{\Psi}}(\omega,\bar{\omega})\right]\right\} e^{j\bar{\omega}}, \tag{3.36}$$

and

$$d_{t+1}(\omega,\bar{\omega}) = \bar{d}_{t+1}(\omega,\bar{\omega}) + \left\|\mathbf{f}_{\bar{\Phi}}(\omega,\bar{\omega})\right\|^2, \tag{3.37}$$

where

$$\mathbf{f}_{\bar{\Phi}}(\omega,\bar{\omega}) = \bar{\mathbf{\Phi}}^T_{t-\bar{L}+1}\mathbf{a}^*_{M,\bar{M}}(\omega,\bar{\omega}), \tag{3.38}$$

$$\mathbf{f}_{\bar{\Psi}}(\omega,\bar{\omega}) = \left(\bar{\mathbf{\Phi}}^H_{t-\bar{L}+1}\bar{\mathbf{Z}}_{t+1}\right)\mathbf{a}^*_{L,\bar{L}+1}(\omega,\bar{\omega}), \tag{3.39}$$

and

$$\mathbf{f}_{\bar{G}}(\omega) = \bar{\mathbf{G}}^H_{t-\bar{L}+1}\mathbf{a}^*_L(\omega). \tag{3.40}$$

Note again that $\mathbf{f}_{\bar{\Phi}}(\omega,\bar{\omega})$, $\mathbf{f}_{\bar{\Psi}}(\omega,\bar{\omega})$, and $\mathbf{f}_{\bar{G}}(\omega)$ in (3.36) and (3.37) can be calculated efficiently via FFT.

Once $c_{t+1}(\omega,\bar{\omega})$ and $d_{t+1}(\omega,\bar{\omega})$ have been obtained, $\hat{\alpha}_{\mathrm{FO},t+1}(\omega,\bar{\omega})$ can be calculated using (3.13). The F-O Capon algorithm is summarized in Table 3-I.

Now we compare our recursive implementation algorithm with other possible implementations. Since F-O Capon can be interpreted as an adaptive FIR filtering approach, one possible method is to apply the existing efficient VLSI implementation algorithms, such as the adaptive Capon beamforming (also called minimum variance distortionless response (MVDR) beamformer) algorithms proposed in [24, 25, 26], to obtain the adaptive Capon filters, which are then used to calculate the complex spectrum estimate. However, this is computationally expensive since the adaptive filters $\mathbf{h}_{\mathrm{FO},t}(\omega,\bar{\omega})$ must be computed for every frequency pair $(\omega,\bar{\omega})$ of interest and the number of desired frequency pairs tends to be very large (e.g., 256×256 or larger). Our approach avoids this problem by combining the adaptive filter calculation with the complex spectral estimation together. Another possible way is to directly update and downdate the Hermitian square root $\mathbf{D}_t$ of $\mathbf{P}^{-1}_t$. This is the

inverse updating and downdating concept discussed in [20]. However, even without considering the overhead of updating $\mathbf{D}_{t+1}$ from $\mathbf{D}_t$, the matrix multiplication $\mathbf{D}_t^H\mathbf{Z}_t$ in (3.20) is computationally more demanding than our recursive algorithm. We have also considered other updating and downdating techniques for different matrix factorizations [19] and found that they are also not very suitable to our problem of interest.

3.3.4 Computational Complexity

The computational complexity of the proposed algorithm depends on the following parameters: (a) N, the number of samples in range; (b) $\bar{N}$, the sliding window length in cross-range; (c) M and $\bar{M}$, the dimensions of the 2-D adaptive FIR filter; and (d) N_p and $\bar{N}_p$, the dimensions of the ISAR images. For the sake of simplicity, we assume $N = \bar{N}$, $M = \bar{M}$, $L = N - M + 1$, and $N_p = \bar{N}_p$. The computational complexities for the direct implementation (without using FFT) and the proposed efficient initialization method are compared in Table 3-II, where one flop is defined as one complex multiplication plus one accumulation. The computational complexity needed by F-O Capon to update an image is also given in Table 3-II. From Table 3-II, it can be noted that, for the batch processing mode, the proposed initialization method is more efficient than the direct implementation since typically $N_p \gg M$ and $N_p \gg N$. This improvement is achieved due to the appropriate use of FFT. The larger the ratio of N_p over M, the larger the speedup ratio. The recursive implementation is more efficient than the efficient initialization method. Computational savings of the recursive algorithm come from two aspects: (a) the appropriate application of FFT, and (b) the recursive updating and downdating of the sample covariance matrix inversion by using the Matrix Inversion Lemma. Compared to the STFT method, the F-O Capon algorithms are computationally more expensive. However, better performance (high resolution, low sidelobes, reduced speckles) can be achieved with the latter [10].

With the rapid development of DSP chips, it is not enough to only count the number of flops as a measure of algorithm efficiency. The computational kernel involved is also very important for the implementation using DSP chips. Matrix multiplications (including $\tilde{\mathbf{Z}}_{t+1}^H\mathbf{P}_t^{-1}\tilde{\mathbf{Z}}_{t+1}$, $\mathbf{\Phi}_{t+1} = \mathbf{P}_t^{-1}\tilde{\mathbf{Z}}_{t+1}\mathbf{G}_{t+1}$, $\mathbf{\Phi}_{t+1}^H\mathbf{Z}_t$, $\mathbf{\Phi}_{t+1}\mathbf{\Phi}_{t+1}^H$, $\tilde{\mathbf{Z}}_{t-\bar{L}+1}^H\bar{\mathbf{P}}_{t+1}^{-1}\tilde{\mathbf{Z}}_{t-\bar{L}+1}$, $\bar{\mathbf{\Phi}}_{t-\bar{L}+1} =$

$\bar{\mathbf{P}}_{t+1}^{-1}\tilde{\mathbf{Z}}_{t-\bar{L}+1}\bar{\mathbf{G}}_{t-\bar{L}+1}$, $\bar{\mathbf{\Phi}}_{t-\bar{L}+1}^{H}\bar{\mathbf{Z}}_{t+1}$, and $\bar{\mathbf{\Phi}}_{t-\bar{L}+1}\bar{\mathbf{\Phi}}_{t-\bar{L}+1}^{H}$) and FFT are the dominant parts of the updating and downdating processes of our recursive algorithm. They can be easily and efficiently implemented using currently available DSP chips, such as the TM-66 *swi*FFT chip [27].

3.4 Forward-Backward Capon

In this section, we first briefly review F-B Capon with emphasis on pointing out its difference from F-O Capon. Since the deriviations of the initialization and recursive implementation of F-B Capon is very similar to those of F-O Capon, we only outline the necessary steps.

3.4.1 Brief Review of F-B Capon

The F-O adaptive weight vector in (3.10) uses only the forward sample covariance matrix $\mathbf{P}_t$. It is commonly believed that using both the forward and backward sample covariance matrices will lead to enhanced statistical performance. Using both the forward and backward sample matrices has the advantages of yielding a numerically better conditioned matrix [28], which may be an important reason to prefer it when the forward sample covariance matrix is ill-conditioned. The method so-obtained is referred to as F-B Capon. Further performance analysis shows that F-B Capon has smaller biases than F-O Capon [10]. Below we briefly review the F-B Capon algorithm.

Let

$$\breve{\mathbf{P}}_t = \mathbf{P}_t + \mathbf{J}\mathbf{P}_t^T\mathbf{J}, \tag{3.41}$$

denote the forward-backward sample covariance matrix, where $\mathbf{J}$ denotes the exchange matrix whose anti-diagonal elements are ones and all the others are zero. By replacing the $\mathbf{P}_t$ in (3.10) with $\breve{\mathbf{P}}_t$, the F-B adaptive Capon filter has the following form [10]:

$$\mathbf{h}_{\mathrm{FB},t}(\omega,\bar{\omega}) = \frac{\breve{\mathbf{P}}_t^{-1}\mathbf{a}_{M,\bar{M}}(\omega,\bar{\omega})}{\mathbf{a}_{M,\bar{M}}^H(\omega,\bar{\omega})\breve{\mathbf{P}}_t^{-1}\mathbf{a}_{M,\bar{M}}(\omega,\bar{\omega})}. \tag{3.42}$$

The forward and backward estimate of $\alpha_t(\omega,\bar{\omega})$ is given by [10]

$$\hat{\alpha}_{\mathrm{FB},t}(\omega,\bar{\omega}) = \frac{\mathbf{a}_{M,\bar{M}}^H(\omega,\bar{\omega})\breve{\mathbf{P}}_t^{-1}\left[\sum_{\bar{l}=0}^{\bar{L}-1}\sum_{l=0}^{L-1}\mathbf{z}_{l,t-\bar{l}}e^{-j\{l\omega+(\bar{L}-1-\bar{l})\bar{\omega}\}}\right]}{\bar{L}L\mathbf{a}_{M,\bar{M}}^H(\omega,\bar{\omega})\breve{\mathbf{P}}_t^{-1}\mathbf{a}_{M,\bar{M}}(\omega,\bar{\omega})}. \tag{3.43}$$

By comparing (3.43) with (3.11), we observe that the only difference between F-O and F-B Capon is that the forward sample covariance matrix $\mathbf{P}_t$ of the former is replaced by its forward-backward counterpart $\breve{\mathbf{P}}_t$. Since $\breve{\mathbf{P}}_t$ can be calculated efficiently from $\mathbf{P}_t$ by using (3.41), for the batch processing mode, the amount of computations needed by F-O and F-B Capon is basically the same. However, as shown next, for recursive implementation, F-O Capon is more efficient than F-B Capon.

3.4.2 Summary of the F-B Capon Algorithm

The F-B Capon estimate of $\alpha_t(\omega,\bar{\omega})$ in (3.43) can be rewritten in the following compact form:

$$\hat{\alpha}_{\mathrm{FB},t}(\omega,\bar{\omega}) = \frac{\breve{c}_t(\omega,\bar{\omega})}{\bar{L}L\breve{d}_t(\omega,\bar{\omega})}, \tag{3.44}$$

where

$$\breve{c}_t(\omega,\bar{\omega}) = \mathbf{a}_{M,\bar{M}}^H(\omega,\bar{\omega})\breve{\mathbf{P}}_t^{-1}\mathbf{Z}_t\mathbf{a}_{L,\bar{L}}^*(\omega,\bar{\omega}), \tag{3.45}$$

and

$$\breve{d}_t(\omega,\bar{\omega}) = \mathbf{a}_{M,\bar{M}}^H(\omega,\bar{\omega})\breve{\mathbf{P}}_t^{-1}\mathbf{a}_{M,\bar{M}}(\omega,\bar{\omega}). \tag{3.46}$$

Similarly to the F-O Capon algorithm, $\hat{\alpha}_{\mathrm{FB},t+1}(\omega,\bar{\omega})$ can also be updated from $\hat{\alpha}_{\mathrm{FB},t}(\omega,\bar{\omega})$ by updating $\breve{c}_{t+1}(\omega,\bar{\omega})$ and $\breve{d}_{t+1}(\omega,\bar{\omega})$, respectively, from $\breve{c}_t(\omega,\bar{\omega})$ and $\breve{d}_t(\omega,\bar{\omega})$. This can be done in two steps: (a) update $\bar{\breve{c}}_{t+1}(\omega,\bar{\omega})$ and $\bar{\breve{d}}_{t+1}(\omega,\bar{\omega})$, respectively, from $\breve{c}_t(\omega,\bar{\omega})$ and $\breve{d}_t(\omega,\bar{\omega})$, where $\bar{\breve{c}}_{t+1}(\omega,\bar{\omega})$ and $\bar{\breve{d}}_{t+1}(\omega,\bar{\omega})$ are defined similarly to $\breve{c}_{t+1}(\omega,\bar{\omega})$ and $\breve{d}_{t+1}(\omega,\bar{\omega})$ in (3.14) and (3.15), respectively, except that the data sequence $\{z_{l,t+1-\bar{l}}, l=0,1\cdots,L-1,\bar{l}=0,1,\cdots,\bar{L}-1\}$ is replaced by $\{z_{l,t+1-\bar{l}}, l=0,1\cdots,L-1,\bar{l}=0,1,\cdots,\bar{L}\}$; (b) downdate $\breve{c}_{t+1}(\omega,\bar{\omega})$ and $\breve{d}_{t+1}(\omega,\bar{\omega})$ from $\bar{\breve{c}}_t(\omega,\bar{\omega})$ and $\bar{\breve{d}}_t(\omega,\bar{\omega})$, respectively.

Before we summarize the steps needed by F-B Capon, let us first define the following notations. Let

$$\breve{\mathbf{z}}_{l,t-\bar{l}} = \mathbf{J}\mathbf{z}^*_{l,t-\bar{l}}, \tag{3.47}$$

denote the $(l, \bar{l})$th backward data vector where $(\cdot)^*$ denotes the complex conjugate and $\mathbf{z}_{l,t-\bar{l}}$ is defined in (3.5) as the $(l, \bar{l})$th forward data vector. Define

$$\breve{\tilde{\mathbf{Z}}}_{t+1} = \begin{bmatrix} \breve{\mathbf{z}}_{0,t+1} & \breve{\mathbf{z}}_{1,t+1} & \cdots & \breve{\mathbf{z}}_{L-1,t+1} \end{bmatrix}, \tag{3.48}$$

and

$$\tilde{\tilde{\mathbf{Z}}}_{t+1} = \begin{bmatrix} \tilde{\mathbf{Z}}_{t+1} & \breve{\tilde{\mathbf{Z}}}_{t+1} \end{bmatrix}, \tag{3.49}$$

where $\tilde{\mathbf{Z}}_{t+1}$ is defined in (3.21). The recursive implementation steps of the F-B Capon algorithm are summarized in Table 3-III. Note that there is no duality between the recursive F-O and F-B Capon algorithms since, although the latter uses both the forward and backward sample covariance matrices to calaulate the adaptive weight vectors, it uses only the forward data vectors for amplitude estimation [10].

As pointed out before, for the batch processing mode, both F-O and F-B Capon have the same computational complexities. However, for the recursive implementation, F-B Capon is less efficient than F-O Capon and $O\left(4M^2L^3 + 10M^4L + 5LN_p^2 \log_2 N_p\right)$ flops are required by F-B Capon to update an image, which are about twice as much as those needed by F-O Capon. Matrix computations and FFT are still the dominant parts of the recursive implementation, which can also be easily and efficiently implemented by using currently available DSP chips.

3.5 A Numerical Example

We present an example to illustrate the performance of using the Capon algorithms for ISAR imaging of maneuvering targets. The signal phase history data of a simulated fast rotating MIG-25 airplane was provided to us by the Naval Research Laboratory. The numbers of range and cross-range samples are 32 and 512 , respectively. In this example, we choose the sliding window length in cross-range to be $\bar{N} = N = 32$, the adaptive filter taps

$M = \bar{M} = 12$, and 2-D Kaiser windows [29] with shape parameter $\beta = 4$ for the windowed FFT and STFT methods. Figure 3.1 shows the ISAR images obtained by applying the FFT and windowed FFT methods (with zero-padding) to the entire 32×512 data matrix. Note that the images are blurred due to the time-varying Doppler shifts in cross-range. Two examples out of the 481 time-varying ISAR images of dimension 256×256 obtained by using windowed STFT, F-O Capon, and F-B Capon are shown in Figures 3.2(a)(b), (c)(d), and (e)(f), respectively. Comparing Figures 3.1 and 3.2, we note that the image blurring problem is mitigated. From Figure 3.2, it can be concluded that both F-O and F-B Capon outperform the STFT method and F-B Capon performs better than F-O Capon.

3.6 Conclusions

We have presented a computationally efficient approach of recursively implementing both the forward-only and forward-backward Capon algorithms for time-varying complex spectral estimation, which has applications in the complex ISAR imaging of maneuvering targets. By applying the Matrix Inversion Lemma and FFT where appropriate, the proposed approach is computationally much more efficient than the direct implementation of the algorithms.

Acknowledgement

The authors would like to thank Dr. V. C. Chen for providing us the simulated MIG-25 airplane data.

Reference

[1] W. G. Carrara, R. S. Goodman, and R. M. Majewski, *Spotlight Synthetic Aperture Radar: Signal Processing Algorithms.* Norwood, MA, USA: Artech House, Inc., 1995.

[2] C. V. Jakowatz, Jr., D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. A. Thompson, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach.* Norwell, MA: Kluwer Academic Publishers, 1996.

[3] R. Klemm, "Current trends in SAR technology," *IEEE Aerospace and Electronic Systems Magazine*, vol. 12, no. 3, pp. 3–8, March 1997.

[4] M. T. Fennell and R. P. Wishner, "Battlefield awareness via synergistic SAR and MTI exploitation," *IEEE Aerospace and Electronic Systems Magazine*, vol. 13, no. 2, pp. 39–45, February.

[5] R. Voles, "Resolving revolutions: imaging and mapping by modern radar," *IEE Proceedings, Pt. F*, vol. 140, no. 1, pp. 1–11, February 1993.

[6] S. R. DeGraaf, "SAR imaging via modern 2-D spectral estimation methods," *SPIE Proceedings on Optical Engineering in Aerospace Sensing Orlando, FL*, April 1994.

[7] S. R. DeGraaf, "SAR imaging via modern 2-D spectral estimation methods," *IEEE Transactions on Image Processing*, vol. 7, pp. 729–761, May 1998.

[8] A. W. Rihaczek and S. J. Hershkowitz, *Radar Resolution and Complex-Image Analysis.* Artech House, Inc., 1996.

[9] J. Capon, "High resolution frequency-wavenumber spectrum analysis," *Proceedings of IEEE*, vol. 57, pp. 1408–1418, August 1969.

[10] H. Li, J. Li, and P. Stoica, "Performance analysis of forward-backward matched-filterbank spectral estimators," *IEEE Transactions on Signal Processing*, vol. 46, no. 7, pp. 1954–1966, July 1998.

[11] G. R. Benitz, "Adaptive high-definition imaging," *SPIE Proceedings on Optical Engineering in Aerospace Sensing,* Orlando , FL, April 1994.

[12] G. R. Benitz, "High definition vector imaging for synthetic aperture radar," *Proceedings of the 31st Asilomar Conference on Signals, Systems and Computers,* Pacific Grove, CA, November 1997.

[13] L. M. Novak, G. R. Nenitz, G. J. Owirka, and L. A. Bessette, "ATR performance using enhanced resolution SAR," *SPIE Proceedings on Optical Engineering in Aerospace Sensing,* Orlando , FL, pp. 332–337, April 1996.

[14] L. M. Novak, G. R. Nenitz, and G. J. Owirka, "Classifier performance using enhanced resolution SAR data," *Proceedings of the IEE International Conference on Radar,* Edinburgh, UK, pp. 634–638, October 1997.

[15] V. C. Chen, "Time-frequency based ISAR image formation technique," *Proceedings of SPIE:Algorithms for Synthetic Aperture Radar Imagery IV*, pp. 43–54, April 1997.

[16] F. Hlawatsch and G. F. Boudreaux-Bartels, "Linear and quadratic time-frequency signal representations," *IEEE Signal Processing Magazine*, pp. 21–67, April 1992.

[17] P. Stoica, A. Jakobsson, and J. Li, "Capon, APES and matched-filterbank spectral estimation," *Signal Processing*, vol. 66, no. 1, pp. 45–59, April 1998.

[18] A. Graham, *Kronecker Products and Matrix Calculus with Applications.* Chichester, UK: Ellis Horwood Ltd., 1981.

[19] G. H. Golub and C. F. Van Loan, *Matrix Computations.* Baltimore, MD: The John Hopkins University Press, 1996.

[20] A. Björck, H. Park, and L. Eldén, "Accurate downdating of the least squares solutions," *SIAM Journal on Matrix Analysis and Applications*, vol. 15, no. 2, pp. 549–568, April 1994.

[21] W. W. Hager, "Updating the inverse of a matrix," *SPIE Review*, vol. 31, no. 2, pp. 221–239, June 1989.

[22] S. Haykin, *Adaptive Filter Theory.* Englewood Cliffs, NJ: Prentice-Hall, 1986.

[23] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification.* Cambridge, Massachusetts: The MIT Press, 1983.

[24] J. G. McWhirter and T. J. Shepherd, "Systolic array processor for MVDR beamforming," *IEE Proceedings, Pt. F*, vol. 136, no. 2, pp. 75–80, April 1989.

[25] M. Moonen, "Systolic MVDR beamforming with inverse updating," *IEE Proceedings, Pt. F*, vol. 140, no. 3, pp. 175–178, June 1993.

[26] S. M. Yuen, "Exact least squares adaptive beamforming using an orthogonalization network," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 27, no. 2, pp. 311–330, March 1991.

[27] *TM-66 swiFFT chip*, Texas Memory Systems, Inc., 1998.

[28] P. Stoica and T. Söderström, "On a novel subspace-based approach to parameter estimation," *Digital Signal Processing: A Review Journal*, vol. 5, no. 4, pp. 237–242, October 1995.

[29] P. Stoica and R. L. Moses, *Introduction to Spectral Analysis.* Englewood Cliffs, NJ: Prentice-Hall, 1997.

TABLE 3-I

SUMMARY OF RECURSIVE F-O CAPON

Step 1: Initialization

Compute $\mathbf{P}_t = \sum_{\bar{l}=0}^{\bar{L}-1} \sum_{l=0}^{L-1} \mathbf{z}_{l,t-\bar{l}} \mathbf{z}_{l,t-\bar{l}}^H$;

Compute $\mathbf{P}_t^{-1}$ and find its Hermitian square root $\mathbf{D}_t$;

Compute:

$\mathbf{F}_\alpha(\omega,\bar{\omega}) = \mathbf{a}_{M,\bar{M}}^H(\omega,\bar{\omega})\mathbf{D}_t$;

$\mathbf{F}_\beta(\omega,\bar{\omega}) = \left(\mathbf{D}_t^H \mathbf{Z}_t\right) \mathbf{a}_{L,\bar{L}}^*(\omega,\bar{\omega})$;

$c_t(\omega,\bar{\omega}) = \mathbf{F}_\alpha(\omega,\bar{\omega})\mathbf{F}_\beta(\omega,\bar{\omega})$;

$d_t(\omega,\bar{\omega}) = \|\mathbf{F}_\alpha(\omega,\bar{\omega})\|^2$.

Step 2: Updating $\bar{c}_{t+1}(\omega,\bar{\omega})$ and $\bar{d}_{t+1}(\omega,\bar{\omega})$

Compute $\left(\mathbf{I} + \tilde{\mathbf{Z}}_{t+1}^H \mathbf{P}_t^{-1} \tilde{\mathbf{Z}}_{t+1}\right)^{-1}$ and find its Hermitian square root $\mathbf{G}_{t+1}$;

Compute:

$\mathbf{\Phi}_{t+1} = \mathbf{P}_t^{-1} \tilde{\mathbf{Z}}_{t+1} \mathbf{G}_{t+1}$;

$\bar{\mathbf{P}}_{t+1}^{-1} = \mathbf{P}_t^{-1} - \mathbf{\Phi}_{t+1}\mathbf{\Phi}_{t+1}^H$;

$\mathbf{f}_\Phi(\omega,\bar{\omega}) = \mathbf{\Phi}_{t+1}^T \mathbf{a}_{M,\bar{M}}^*(\omega,\bar{\omega})$;

$\mathbf{f}_\Psi(\omega,\bar{\omega}) = (\mathbf{\Phi}_{t+1}^H \mathbf{Z}_t)\mathbf{a}_{L,\bar{L}}^*(\omega,\bar{\omega})$;

$\mathbf{f}_G(\omega) = \mathbf{G}_{t+1}^H \mathbf{a}_L^*(\omega)$;

Update:

$\bar{c}_{t+1}(\omega,\bar{\omega}) = c_t(\omega,\bar{\omega}) + \mathbf{f}_\Phi^T(\omega,\bar{\omega}) \left\{\mathbf{f}_G(\omega)e^{-j\bar{L}\bar{\omega}} - \mathbf{f}_\Psi(\omega,\bar{\omega})\right\}$;

$\bar{d}_{t+1}(\omega,\bar{\omega}) = d_t(\omega,\bar{\omega}) - \|\mathbf{f}_\Phi(\omega,\bar{\omega})\|^2$.

Step 3: Downdating $c_{t+1}(\omega,\bar{\omega})$ and $d_{t+1}(\omega,\bar{\omega})$

Compute $\left(\mathbf{I} - \tilde{\mathbf{Z}}_{t-\bar{L}+1}^H \bar{\mathbf{P}}_{t+1}^{-1} \tilde{\mathbf{Z}}_{t-\bar{L}+1}\right)^{-1}$ and find its Hermitian square root $\bar{\mathbf{G}}_{t-\bar{L}+1}$;

Compute:

$\bar{\mathbf{\Phi}}_{t-\bar{L}+1} = \bar{\mathbf{P}}_{t+1}^{-1} \tilde{\mathbf{Z}}_{t-\bar{L}+1} \bar{\mathbf{G}}_{t-\bar{L}+1}$;

$\mathbf{P}_{t+1} = \bar{\mathbf{P}}_{t+1} - \tilde{\mathbf{Z}}_{t-\bar{L}+1} \tilde{\mathbf{Z}}_{t-\bar{L}+1}^H$;

$\mathbf{f}_{\bar{\Phi}}(\omega,\bar{\omega}) = \bar{\mathbf{\Phi}}_{t-\bar{L}+1}^T \mathbf{a}_{M,\bar{M}}^*(\omega,\bar{\omega})$;

$\mathbf{f}_{\bar{\Psi}}(\omega,\bar{\omega}) = \left(\bar{\mathbf{\Phi}}_{t-\bar{L}+1}^H \bar{\mathbf{Z}}_{t+1}\right) \mathbf{a}_{L,\bar{L}+1}^*(\omega,\bar{\omega})$;

$\mathbf{f}_{\bar{G}}(\omega) = \bar{\mathbf{G}}_{t-\bar{L}+1}^H \mathbf{a}_L^*(\omega)$;

Downdate:

$c_{t+1}(\omega,\bar{\omega}) = \left\{\bar{c}_{t+1}(\omega,\bar{\omega}) - \mathbf{f}_{\bar{\Phi}}^T(\omega,\bar{\omega}) \left[\mathbf{f}_{\bar{G}}(\omega) - \mathbf{f}_{\bar{\Psi}}(\omega,\bar{\omega})\right]\right\} e^{j\bar{\omega}}$;

$d_{t+1}(\omega,\bar{\omega}) = \bar{d}_{t+1}(\omega,\bar{\omega}) + \|\mathbf{f}_{\bar{\Phi}}(\omega,\bar{\omega})\|^2$.

Step 4: Computing $\hat{\alpha}_{\mathrm{FO},t+1}(\omega,\bar{\omega}) = \frac{c_{t+1}(\omega,\bar{\omega})}{\bar{L}L d_{t+1}(\omega,\bar{\omega})}$.

TABLE 3-II

COMPARISON OF COMPUTATIONAL COMPLEXITIES

STFT or windowed STFT: $O\left(\frac{1}{2}N_p^2 \log_2 N_p\right)$ flops
Direct Implementation: $O\left(M^2(M^2+L^2)N_p^2\right)$ flops
Efficient Initialization: $O\left(7M^6 + \frac{3}{2}M^4L^2 + M^2N_p^2 \log_2 N_p\right)$ flops
Recursive F-O Capon: $O\left(2M^2L^3 + 4M^4L + 2LN_p^2 \log_2 N_p\right)$ flops

TABLE 3-III
SUMMARY OF RECURSIVE F-B CAPON

Step 1: Initialization

Compute $\mathbf{P}_t = \sum_{\bar{l}=0}^{\bar{L}-1} \sum_{l=0}^{L-1} \mathbf{z}_{l,t-\bar{l}} \mathbf{z}_{l,t-\bar{l}}^H$ and $\breve{\mathbf{P}}_t = \mathbf{P}_t + \mathbf{J}\mathbf{P}_t^T\mathbf{J}$;

Compute $\breve{\mathbf{P}}_t^{-1}$ and find its Hermitian square root $\breve{\mathbf{D}}_t$;

Compute:

$\breve{\mathbf{F}}_\alpha(\omega,\bar{\omega}) = \mathbf{a}_{M,\bar{M}}^H(\omega,\bar{\omega})\breve{\mathbf{D}}_t$;

$\breve{\mathbf{F}}_\beta(\omega,\bar{\omega}) = \left(\breve{\mathbf{D}}_t^H \mathbf{Z}_t\right) \mathbf{a}_{L,\bar{L}}^*(\omega,\bar{\omega})$;

$\breve{c}_t(\omega,\bar{\omega}) = \breve{\mathbf{F}}_\alpha(\omega,\bar{\omega})\breve{\mathbf{F}}_\beta(\omega,\bar{\omega})$;

$\breve{d}_t(\omega,\bar{\omega}) = \left\|\breve{\mathbf{F}}_\alpha(\omega,\bar{\omega})\right\|^2$.

Step 2: Updating $\bar{\breve{c}}_{t+1}(\omega,\bar{\omega})$ and $\bar{\breve{d}}_{t+1}(\omega,\bar{\omega})$

Compute $\left(\mathbf{I} + \tilde{\bar{\mathbf{Z}}}_{t+1}^H \breve{\mathbf{P}}_t^{-1} \tilde{\bar{\mathbf{Z}}}_{t+1}\right)^{-1}$ and find its Hermitian square root $\breve{\mathbf{G}}_{t+1}$;

Compute:

$\breve{\mathbf{\Phi}}_{t+1} = \breve{\mathbf{P}}_t^{-1} \tilde{\bar{\mathbf{Z}}}_{t+1} \breve{\mathbf{G}}_{t+1}$;

$\bar{\breve{\mathbf{P}}}_{t+1}^{-1} = \breve{\mathbf{P}}_t^{-1} - \breve{\mathbf{\Phi}}_{t+1}\breve{\mathbf{\Phi}}_{t+1}^H$;

$\mathbf{f}_{\breve{\Phi}}(\omega,\bar{\omega}) = \breve{\mathbf{\Phi}}_{t+1}^T \mathbf{a}_{M,\bar{M}}^*(\omega,\bar{\omega})$;

$\mathbf{f}_{\breve{\Psi}}(\omega,\bar{\omega}) = \left(\breve{\mathbf{\Phi}}_{t+1}^H \bar{\mathbf{Z}}_{t+1}\right) \mathbf{a}_{L,\bar{L}+1}^*(\omega,\bar{\omega})$;

$\mathbf{f}_Z(\omega,\bar{\omega}) = \left(\tilde{\mathbf{Z}}_{t+1}^T \breve{\mathbf{P}}_t^{-T}\right) \mathbf{a}_{M,\bar{M}}^*(\omega,\bar{\omega})$;

Update:

$\bar{\breve{c}}_{t+1}(\omega,\bar{\omega}) = \breve{c}_t(\omega,\bar{\omega}) + \mathbf{f}_Z^T(\omega,\bar{\omega})\mathbf{a}_L^*(\omega)e^{-j\bar{L}\bar{\omega}} - \mathbf{f}_{\bar{\breve{\Phi}}}^T(\omega,\bar{\omega})\mathbf{f}_{\breve{\Psi}}(\omega,\bar{\omega})$;

$\bar{\breve{d}}_{t+1}(\omega,\bar{\omega}) = \breve{d}_t(\omega,\bar{\omega}) - \| \mathbf{f}_{\breve{\Phi}}(\omega,\bar{\omega}) \|^2$.

Step 3: Downdating $\breve{c}_{t+1}(\omega,\bar{\omega})$ and $\breve{d}_{t+1}(\omega,\bar{\omega})$

Compute $\left(\mathbf{I} - \tilde{\bar{\mathbf{Z}}}_{t-\bar{L}+1}^H \bar{\breve{\mathbf{P}}}_{t+1}^{-1} \tilde{\bar{\mathbf{Z}}}_{t-\bar{L}+1}\right)^{-1}$ and find its Hermitian square root $\bar{\breve{\mathbf{G}}}_{t-\bar{L}+1}$;

Compute:

$\bar{\breve{\mathbf{\Phi}}}_{t-\bar{L}+1} = \bar{\breve{\mathbf{P}}}_{t+1}^{-1} \tilde{\bar{\mathbf{Z}}}_{t-\bar{L}+1} \bar{\breve{\mathbf{G}}}_{t-\bar{L}+1}$;

$\breve{\mathbf{P}}_{t+1}^{-1} = \bar{\breve{\mathbf{P}}}_{t+1}^{-1} + \bar{\breve{\mathbf{\Phi}}}_{t-\bar{L}+1}\bar{\breve{\mathbf{\Phi}}}_{t-\bar{L}+1}^H$;

$\mathbf{f}_{\bar{\breve{\Phi}}}(\omega,\bar{\omega}) = \bar{\breve{\mathbf{\Phi}}}_{t-\bar{L}+1}^T \mathbf{a}_{M,\bar{M}}^*(\omega,\bar{\omega})$;

$\mathbf{f}_{\bar{\breve{\Psi}}}(\omega,\bar{\omega}) = \left(\bar{\breve{\mathbf{\Phi}}}_{t-\bar{L}+1}^H \mathbf{Z}_{t+1}\right) \mathbf{a}_{L,\bar{L}}^*(\omega,\bar{\omega})$;

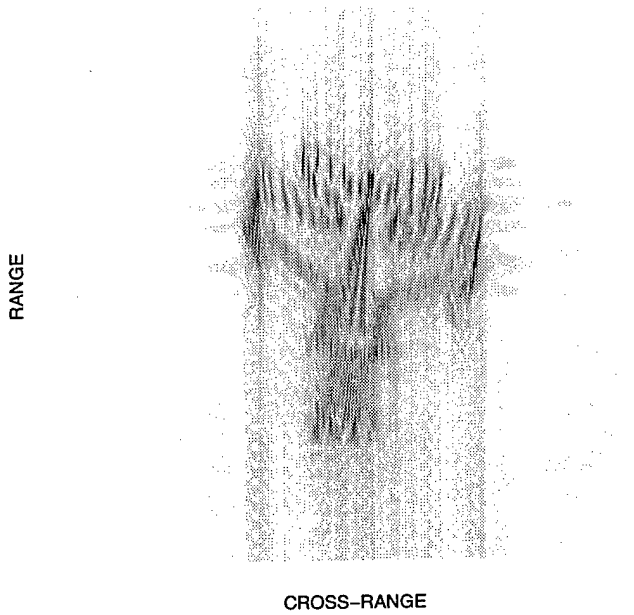
$\mathbf{f}_{\bar{Z}}(\omega,\bar{\omega}) = \left(\tilde{\mathbf{Z}}_{t-\bar{L}+1}^T \bar{\breve{\mathbf{P}}}_{t+1}^{-T}\right) \mathbf{a}_{M,\bar{M}}^*(\omega,\bar{\omega})$;

Downdate:

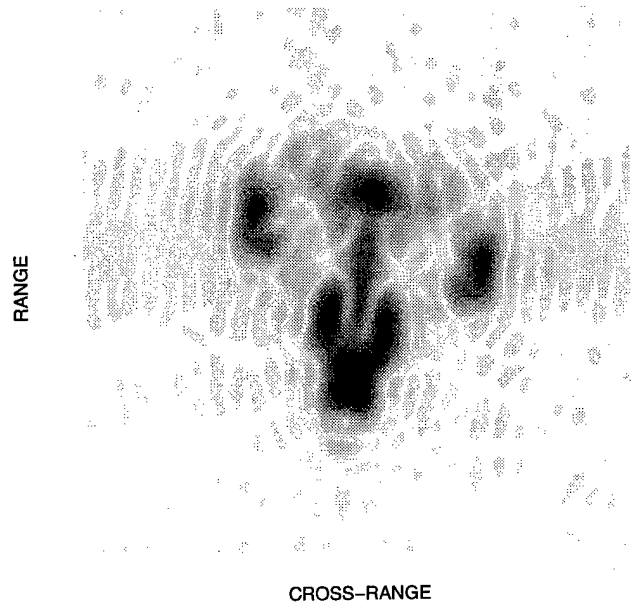
$\breve{c}_{t+1}(\omega,\bar{\omega}) = \left\{\bar{\breve{c}}_{t+1}(\omega,\bar{\omega}) - \mathbf{f}_{\bar{Z}}^T(\omega,\bar{\omega})\mathbf{a}_L^*(\omega) + \mathbf{f}_{\bar{\breve{\Phi}}}^T(\omega,\bar{\omega})\mathbf{f}_{\bar{\breve{\Psi}}}(\omega,\bar{\omega})\right\} e^{j\bar{\omega}}$;

$\breve{d}_{t+1}(\omega,\bar{\omega}) = \bar{\breve{d}}_{t+1}(\omega,\bar{\omega}) + \| \mathbf{f}_{\bar{\breve{\Phi}}}(\omega,\bar{\omega}) \|^2$.

Step 4: Computing $\hat{\alpha}_{\mathrm{FB},t+1}(\omega,\bar{\omega}) = \frac{\breve{c}_{t+1}(\omega,\bar{\omega})}{\bar{L}L\breve{d}_{t+1}(\omega,\bar{\omega})}$.



(a)



(b)

Figure 3.1: Blurred ISAR images of a simulated moving MIG-25 airplane obtained by using (a) the FFT and (b) the windowed FFT.

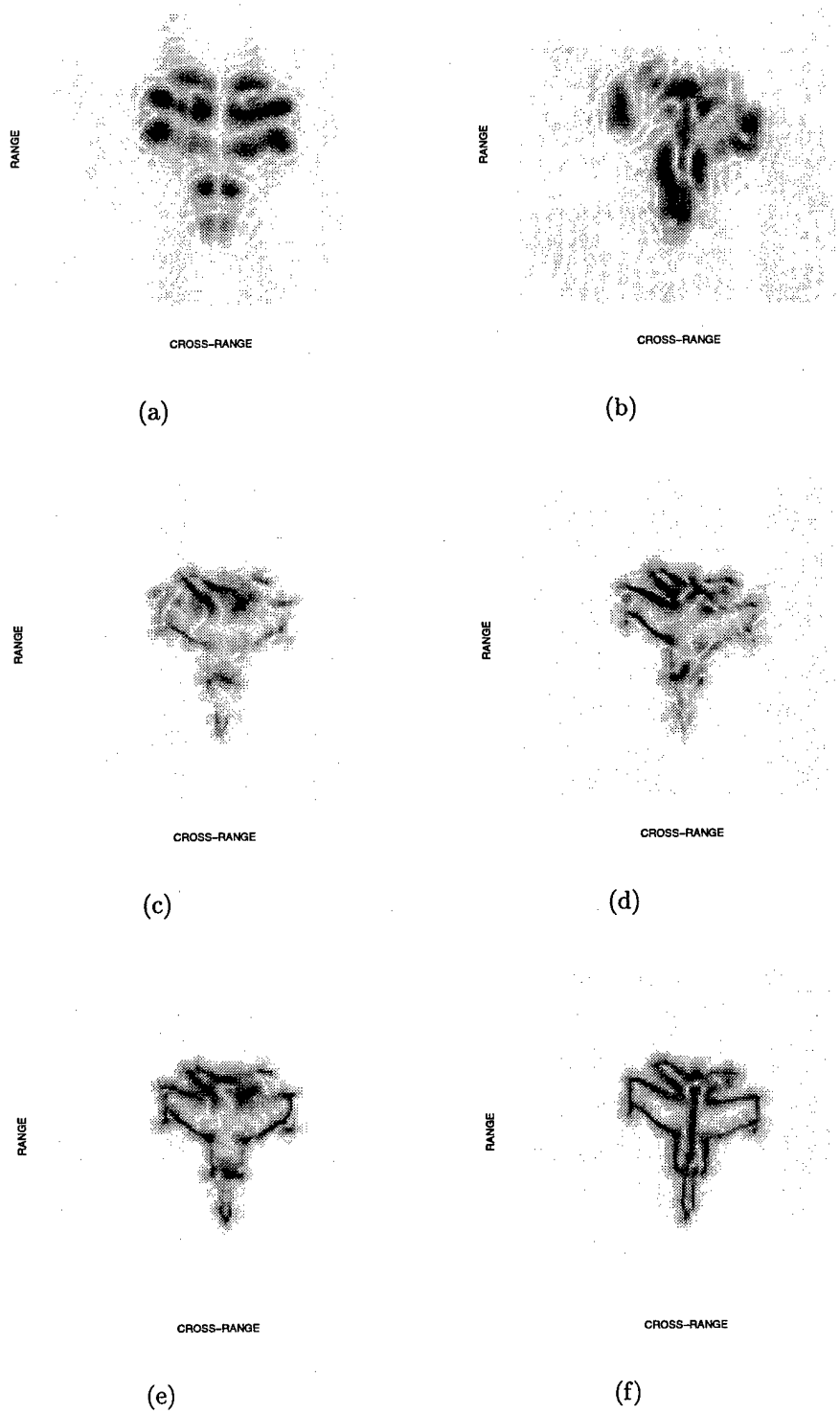


Figure 3.2: Two examples out of the 481 time-varying ISAR images of a simulated moving MIG-25 airplane obtained by using (a)(b) 2-D windowed STFT, (c)(d) 2-D F-O Capon, and (e)(f) 2-D F-B Capon.

4. Time-Varying Complex Spectral Estimation with Application to ISAR Imaging

4.1 Introduction

Complex spectral estimation is very important for many applications including synthetic aperture radar (SAR) and inverse synthetic aperture radar (ISAR) imaging and feature extraction of stationary or moving targets [1]. Two widely used methods for this problem are the fast Fourier transform (FFT) approach and the Capon method [2]. Recently, a new approach, which is referred to as APES (Amplitude and Phase EStimation) was proposed in [3]. Both APES and Capon make use of adaptive FIR (finite impulse response) filters and belong to the class of matched filterbank spectral estimators. They can yield spectral estimates with much lower sidelobes and narrower spectral peaks than the FFT method. Further analysis results show that the Capon estimates are always biased downward whereas the APES estimates are unbiased (to within a second-order approximation) [4]. The theoretical results therein supplemented with the empirical observation that Capon usually underestimates the spectrum in samples of practical length while APES is nearly unbiased are believed to provide a compelling reason for preferring APES over Capon.

In SAR/ISAR, the high range resolution can be achieved by transmitting signals with large bandwidth, whereas high resolution cross-range discrimination is obtained through the relative motion between the target and the radar to form a large synthetic aperture. As pointed out in [5], in SAR/ISAR, *motion* is the *solution* and the *problem*. This statement is especially true for ISAR imaging. In ISAR, the synthetic aperture is formed through the noncooperative motion of the targets. Most ISAR imaging algorithms are based on the range-Doppler processing, which implies that the Doppler shifts nearly remain constant during the coherent integration interval. Unfortunately, for maneuvering targets, this assumption is more often than not violated. This observation has motivated the use of various time-varying spectral analysis methods to ISAR imaging of maneuvering targets [6].

Short-Time Fourier Transform (STFT) is the simplest way for time-varying complex spectral analysis [7]. Major drawbacks associated with STFT include poor resolution and

accuracy and high sidelobes. The Wigner-Ville Distribution (WVD) method [7] is another popular time-frequency analysis approach which exhibits better resolution than STFT. However, it suffers from the the cross-term interference problem that occurs when multiple signals exist together. Moreover, WVD, including its modifications [8, 9], belongs to the family of quadratic time-frequency representation methods and hence *cannot* be used for the *complex* spectral estimation problem considered herein.

In this chapter, we attempt to apply the APES algorithm to sliding short-time data sequences with maximal overlapping for the time-varying complex spectral analysis. The direct application of APES to each data sequence is computationally prohibitive. A computationally efficient recursive APES algorithm is developed in this chapter, which involves only FFT and simple matrix operations. It exhibits much better resolution than STFT. ISAR imaging examples show that it can successfully circumvent the imaging blurring problem caused by target maneuvering.

The remainder of this chapter is organized as follows. In Section 4.2, we formulate the problem of interest. Efficient recursive implementations of one- and two-dimensional APES algorithms are given in Sections 4.3 and 4.4, respectively. In Section 4.5, we present several numerical and experimental examples showing both the imaging quality and the computational efficiency of the proposed implementations. Finally, Section 4.6 gives our conclusions.

4.2 Problem Formulation

Let $\{y_{t-n}\}_{n=0}^{N-1}$ denote a one-dimensional (1-D) discrete-time data sequence within a sliding window of length N ending at time instant t. For a frequency ω of interest, y_{t-n} can be written as

$$y_{t-n} = \alpha_t(\omega)e^{j(N-1-n)\omega} + w_{t-n}(\omega), \quad n = 0, 1, \cdots, N-1, \tag{4.1}$$

where $\alpha_t(\omega)$ denotes the complex amplitude of a sinusoid with frequency ω at time instant t and $w_{t-n}(\omega)$ denotes the unmodeled noise and interference at frequency ω. The problem of interest herein is to estimate $\alpha_t(\omega)$ from $\{y_{t-n}\}_{n=0}^{N-1}$ for all ω of interest as t varies.

Let $\{z_{n,t-\bar{n}}, n = 0, 1, \cdots, N-1, \bar{n} = 0, 1, \cdots, \bar{N}-1\}$ denote a two-dimensional (2-D) discrete-time data sequence within a sliding window of size $N \times \bar{N}$ ending at time instant t. (In ISAR, t is related to the position in cross-range and is referred to as *slow* time.) For a frequency pair $(\omega, \bar{\omega})$ of interest, $z_{n,t-\bar{n}}$ can be written as

$$z_{n,t-\bar{n}} = \alpha_t(\omega, \bar{\omega}) e^{j\{n\omega + (\bar{N}-1-\bar{n})\bar{\omega}\}} + w_{n,t-\bar{n}}(\omega, \bar{\omega}), \quad n = 0, 1, \cdots, N-1, \bar{n} = 0, 1, \cdots, \bar{N}-1, \tag{4.2}$$

where $\alpha_t(\omega, \bar{\omega})$ denotes the complex amplitude of a 2-D sinusoid with frequency $(\omega, \bar{\omega})$ at time instant t and $w_{n,t-\bar{n}}(\omega, \bar{\omega})$ denotes the unmodeled noise and interference at frequency $(\omega, \bar{\omega})$. The problem now is to obtain the estimate of $\alpha_t(\omega, \bar{\omega})$ from the 2-D data sequence $\{z_{n,t-\bar{n}}, n = 0, 1, \cdots, N-1, \bar{n} = 0, 1, \cdots, \bar{N}-1\}$ for all $(\omega, \bar{\omega})$ of interest as t varies. One of the applications of this topic is ISAR imaging of maneuvering targets or wide-angle ISAR imaging after standard motion compensation [10, 11]. In ISAR imaging applications, $\alpha_t(\omega, \bar{\omega})$ would be proportional to the RCS of a scatterer of a moving target located at a range proportional to ω and cross-range proportional to $\bar{\omega}$ at time instant t.

STFT is the simplest method for estimating $\alpha_t(\omega)$ from $\{y_{t-n}\}_{n=0}^{N-1}$ or $\alpha_t(\omega, \bar{\omega})$ from $\{z_{n,t-\bar{n}}, n = 0, 1, \cdots, N-1, \bar{n} = 0, 1, \cdots, \bar{N}-1\}$, respectively. However, the STFT method is known to suffer from poor resolution and accuracy and high sidelobes. The Capon and APES algorithms are effective in reducing the sidelobes and improving the resolutions and APES outperforms Capon in the estimation accuracy. We can apply APES to the short-time data sequences for the purpose of time-varying complex spectral estimation. However, the direct application of APES to each data sequence is computationallly intensive. We must exploit the maximally overlapping features of the short-time data sequences to achieve significant computational savings.

Two versions of the APES algorithms may be considered, i.e., the forward-only and forward-backward APES. It is commonly believed that the forward-backward algorithms have better performance than their forward-only versions, such as for Capon. However, theoretical analysis of the estimation performance and empirical evidence show that the forward-backward APES is not significantly better than the forward-only counterpart [4]

and the recursive version of the former is not significantly faster than the direct use of the former. Hence only the forward-only APES algorithm will be considered in this paper. We shall consider efficient recursive APES algorithms for the aforementioned 1-D and 2-D data sequences in the following sections.

4.3 1-D Recursive APES

4.3.1 Overview of the 1-D APES Estimator

Let M denote the number of taps used by the 1-D APES adaptive FIR filter and

$$\mathbf{y}_{t-l} = \begin{bmatrix} y_{t-l-M+1} & y_{t-l-M+2} & \cdots & y_{t-l} \end{bmatrix}^T, \quad l = 0, 1, \cdots, L-1, \tag{4.3}$$

be the overlapping vectors of the 1-D data sequence, where $L = N - M + 1$ and $(\cdot)^T$ denotes the transpose. Let

$$\mathbf{R}_t = \sum_{l=0}^{L-1} \mathbf{y}_{t-l}\mathbf{y}_{t-l}^H, \tag{4.4}$$

$$\mathbf{g}_t(\omega) = \sum_{l=0}^{L-1} \mathbf{y}_{t-l} e^{-j(L-1-l)\omega}, \tag{4.5}$$

and

$$\mathbf{Q}_t(\omega) = \mathbf{R}_t - \mathbf{g}_t(\omega)\mathbf{g}_t^H(\omega)/L, \tag{4.6}$$

where $(\cdot)^H$ denotes the conjugate transpose. Then the APES estimate of $\alpha_t(\omega)$ has the form [3]:

$$\hat{\alpha}_{\mathrm{APES},t}(\omega) = \frac{\mathbf{a}_M^H(\omega)\mathbf{Q}_t^{-1}(\omega)\mathbf{g}_t(\omega)}{L\mathbf{a}_M^H(\omega)\mathbf{Q}_t^{-1}(\omega)\mathbf{a}_M(\omega)}, \tag{4.7}$$

where

$$\mathbf{a}_M(\omega) = \begin{bmatrix} 1 & e^{j\omega} & \cdots & e^{j(M-1)\omega} \end{bmatrix}^T. \tag{4.8}$$

4.3.2 Efficient Initialization Method

Now we consider how to efficiently calculate $\hat{\alpha}_{\mathrm{APES},t}(\omega)$ in a non-recursive way [12]. It can be used as the initialization method for the recursive algorithm that follows or as a separate time-varying APES estimator working at the batch mode.

According to the matrix inversion lemma, we have

$$\mathbf{Q}_t^{-1}(\omega) = \mathbf{R}_t^{-1} + \frac{\mathbf{R}_t^{-1}\mathbf{g}_t(\omega)\mathbf{g}_t^H(\omega)\mathbf{R}_t^{-1}}{L - \mathbf{g}_t^H(\omega)\mathbf{R}_t^{-1}\mathbf{g}_t(\omega)}. \tag{4.9}$$

Let

$$\mathbf{Y}_t = \begin{bmatrix} \mathbf{y}_{t-L+1} & \mathbf{y}_{t-L+2} & \cdots & \mathbf{y}_t \end{bmatrix}, \tag{4.10}$$

$$b_t(\omega) = \mathbf{a}_M^H(\omega)\mathbf{R}_t^{-1}\mathbf{a}_M(\omega), \tag{4.11}$$

$$c_t(\omega) = \mathbf{a}_M^H(\omega)\mathbf{R}_t^{-1}\mathbf{Y}_t\mathbf{a}_L^*(\omega), \tag{4.12}$$

and

$$d_t(\omega) = \mathbf{a}_L^T\mathbf{Y}_t^H\mathbf{R}_t^{-1}\mathbf{Y}_t\mathbf{a}_L^*(\omega). \tag{4.13}$$

Then we can then rewrite (4.7) as

$$\hat{\alpha}_{\mathrm{APES},t}(\omega) = \frac{c_t(\omega)}{b_t(\omega)\left[L - d_t(\omega)\right] + |c_t(\omega)|^2}. \tag{4.14}$$

The remaining problem is how to calculate $b_t(\omega)$, $c_t(\omega)$, and $d_t(\omega)$ in an efficient way. Let $\mathbf{C}_t$ be the Hermitian square root of $\mathbf{R}_t$, i. e.,

$$\mathbf{R}_t^{-1} = \mathbf{C}_t\mathbf{C}_t^H. \tag{4.15}$$

Then $b_t(\omega)$, $c_t(\omega)$, and $d_t(\omega)$ can be cauculated efficiently via FFT in the following way

$$b_t(\omega) = \|\mathbf{F}_\alpha(\omega)\|^2, \tag{4.16}$$

$$c_t(\omega) = \mathbf{F}_\alpha(\omega)\mathbf{F}_\beta(\omega), \tag{4.17}$$

$$d_t(\omega) = \|\mathbf{F}_\beta(\omega)\|^2, \tag{4.18}$$

where

$$\mathbf{F}_\alpha(\omega) = \mathbf{a}_M^H(\omega)\mathbf{C}_t, \tag{4.19}$$

and

$$\mathbf{F}_\beta(\omega) = \left(\mathbf{C}_t^H\mathbf{Y}_t\right)\mathbf{a}_L^*(\omega). \tag{4.20}$$

Note that both $\mathbf{F}_\alpha(\omega)$ and $\mathbf{F}_\beta(\omega)$ can be computed via FFT.

Compared to those intuitive implementation schemes that do not use FFT to calculate the APES estimates, such as the one used in [4], the computational efficiency can be considerably improved by the above implementation approach. The larger the number of frequencies of interest in the complex spectrum, the larger the improvement. If we use the above scheme as a separate time-varying APES estimator, it is still computationally expensive. Below we describe how to calculate $\hat{\alpha}_{t+1}(\omega)$ from $\hat{\alpha}_t(\omega)$ by taking advantage of the fact that the short-time data sequences at t and $t+1$ are maximally overlapped to achieve more computational savings.

4.3.3 Recursive Implementation

Note that by using (4.14), $\hat{\alpha}_{\text{APES},t+1}(\omega)$ can be updated from $\hat{\alpha}_{\text{APES},t}(\omega)$ by updating $b_{t+1}(\omega)$, $c_{t+1}(\omega)$, and $d_{t+1}(\omega)$, respectively, from $b_t(\omega)$, $c_t(\omega)$, and $d_t(\omega)$.

Below we discuss how to update $b_t(\omega)$, $c_t(\omega)$, and $d_t(\omega)$ as t varies. Let $\bar{b}_{t+1}(\omega)$, $\bar{c}_{t+1}(\omega)$, and $\bar{d}_{t+1}(\omega)$ be defined similarly to $b_{t+1}(\omega)$, $c_{t+1}(\omega)$, and $d_{t+1}(\omega)$ in (4.11), (4.12), and (4.13), respectively, except that the data sequence $\{\mathbf{y}_{t+1-l}\}_{l=0}^{L-1}$ is replaced by $\{\mathbf{y}_{t+1-l}\}_{l=0}^{L}$. We first consider how to update $\bar{b}_{t+1}(\omega)$, $\bar{c}_{t+1}(\omega)$, and $\bar{d}_{t+1}(\omega)$, respectively, from $b_t(\omega)$, $c_t(\omega)$, and $d_t(\omega)$. We then update $b_{t+1}(\omega)$, $c_{t+1}(\omega)$, and $d_{t+1}(\omega)$ from $\bar{b}_{t+1}(\omega)$, $\bar{c}_{t+1}(\omega)$, and $\bar{d}_{t+1}(\omega)$, respectively.

Let

$$\bar{\mathbf{Y}}_{t+1} = \begin{bmatrix} \mathbf{Y}_t & \mathbf{y}_{t+1} \end{bmatrix}, \tag{4.21}$$

and

$$\bar{\mathbf{R}}_{t+1} = \bar{\mathbf{Y}}_{t+1}\bar{\mathbf{Y}}_{t+1}^H = \mathbf{R}_t + \mathbf{y}_{t+1}\mathbf{y}_{t+1}^H. \tag{4.22}$$

Then according to the matrix inversion lemma, $\bar{\mathbf{R}}_{t+1}^{-1}$ can be calculated from $\mathbf{R}_t^{-1}$ as follows:

$$\bar{\mathbf{R}}_{t+1}^{-1} = \mathbf{R}_t^{-1} - \frac{\boldsymbol{\eta}_{t+1}\boldsymbol{\eta}_{t+1}^H}{1+\phi_{t+1}}, \tag{4.23}$$

where

$$\boldsymbol{\eta}_{t+1} = \mathbf{R}_t^{-1}\mathbf{y}_{t+1}, \tag{4.24}$$

and

$$\phi_{t+1} = \mathbf{y}_{t+1}^H \mathbf{R}_t^{-1} \mathbf{y}_{t+1} = \mathbf{y}_{t+1}^H \boldsymbol{\eta}_{t+1}. \tag{4.25}$$

Using (4.23) we obtain

$$\bar{b}_{t+1}(\omega) = b_t(\omega) - \frac{|F_\eta(\omega)|^2}{1+\phi_{t+1}}, \tag{4.26}$$

$$\bar{c}_{t+1}(\omega) = c_t(\omega) + \frac{F_\eta(\omega)\left[e^{-jL\omega} - F_\psi(\omega)\right]}{1+\phi_{t+1}}, \tag{4.27}$$

and

$$\bar{d}_{t+1}(\omega) = d_t(\omega) + \frac{\phi_{t+1} - |F_\psi(\omega)|^2}{1+\phi_{t+1}} + 2\mathrm{Re}\left\{\frac{F_\psi^*(\omega)}{1+\phi_{t+1}} e^{-jL\omega}\right\}, \tag{4.28}$$

where $\mathrm{Re}(x)$ denotes the real part of x,

$$F_\eta(\omega) = \mathbf{a}_M^H(\omega)\boldsymbol{\eta}_{t+1}, \tag{4.29}$$

and

$$F_\psi(\omega) = \boldsymbol{\psi}_{t+1}^H \mathbf{a}_L^*(\omega), \tag{4.30}$$

with

$$\boldsymbol{\psi}_{t+1} = \mathbf{Y}_t^H \boldsymbol{\eta}_{t+1}. \tag{4.31}$$

Note that (4.29) and (4.30) can be computed efficiently via FFT.

Now consider how to update $b_{t+1}(\omega)$, $c_{t+1}(\omega)$, and $d_{t+1}(\omega)$, respectively, from $\bar{b}_{t+1}(\omega)$, $\bar{c}_{t+1}(\omega)$, and $\bar{d}_{t+1}(\omega)$. Since

$$\mathbf{R}_{t+1} = \bar{\mathbf{R}}_{t+1} - \mathbf{y}_{t-L+1}\mathbf{y}_{t-L+1}^H, \tag{4.32}$$

we have

$$\mathbf{R}_{t+1}^{-1} = \bar{\mathbf{R}}_{t+1}^{-1} + \frac{\bar{\boldsymbol{\eta}}_{t-L+1}\bar{\boldsymbol{\eta}}_{t-L+1}^H}{1-\bar{\phi}_{t-L+1}}, \tag{4.33}$$

where

$$\bar{\boldsymbol{\eta}}_{t-L+1} = \bar{\mathbf{R}}_{t+1}^{-1}\mathbf{y}_{t-L+1}, \tag{4.34}$$

and

$$\bar{\phi}_{t-L+1} = \mathbf{y}_{t-L+1}^H \bar{\mathbf{R}}_{t+1}^{-1}\mathbf{y}_{t-L+1} = \mathbf{y}_{t-L+1}^H \bar{\boldsymbol{\eta}}_{t-L+1}. \tag{4.35}$$

Using (4.33), we obtain

$$b_{t+1}(\omega) = \bar{b}_{t+1}(\omega) + \frac{|F_{\bar{\eta}}(\omega)|^2}{1 - \bar{\phi}_{t-L+1}}, \tag{4.36}$$

$$c_{t+1}(\omega) = \left\{ \bar{c}_{t+1}(\omega) - \frac{F_{\bar{\eta}}(\omega)\left[1 - F_{\bar{\psi}}(\omega)\right]}{1 - \bar{\phi}_{t-L+1}} \right\} e^{j\omega}, \tag{4.37}$$

and

$$d_{t+1}(\omega) = \bar{d}_{t+1}(\omega) + \frac{\bar{\phi}_{t-L+1} + \left|F_{\bar{\psi}}(\omega)\right|^2}{1 - \bar{\phi}_{t-L+1}} - 2\mathrm{Re}\left\{ \frac{F^*_{\bar{\psi}}(\omega)}{1 - \bar{\phi}_{t-L+1}} \right\}, \tag{4.38}$$

where

$$F_{\bar{\eta}}(\omega) = \mathbf{a}_M^H(\omega)\bar{\boldsymbol{\eta}}_{t-L+1}, \tag{4.39}$$

and

$$F_{\bar{\psi}}(\omega) = \bar{\boldsymbol{\psi}}_{t+1}^H \mathbf{a}_{L+1}^*(\omega), \tag{4.40}$$

with

$$\bar{\boldsymbol{\psi}}_{t+1} = \bar{\mathbf{Y}}_{t+1}^H \bar{\boldsymbol{\eta}}_{t-L+1}. \tag{4.41}$$

Note again that (4.39) and (4.40) can be computed efficiently via FFT.

From the above derivations, we note that $\hat{\alpha}_{\mathrm{APES},t+1}(\omega)$ can be updated from $\hat{\alpha}_{\mathrm{APES},t}(\omega)$ efficiently via simple matrix operations and FFT. Compared to the non-recursive implementation scheme discussed in the previous subsection, the computationally demanding matrix inversion is avoided. The speed-up ratio of the recursive versus non-recursive implementations depends on M, the number $N_{\max}$ of frequencies of interest in the complex spectrum, and N. The larger the M, the larger the speed-up ratio. We will give some quantitative results on the speed-up ratio in Section 5.

4.4 2-D Recursive APES

4.4.1 Overview of the 2-D APES Estimator

Assume that APES uses an $(M \times \bar{M})$-tap 2-D adaptive FIR filter. Let

$$\mathbf{Z}_{l,t-\bar{l}} = \begin{bmatrix} z_{l,t-\bar{l}-\bar{M}+1} & z_{l,t-\bar{l}-\bar{M}+2} & \cdots & z_{l,t-\bar{l}} \\ z_{l+1,t-\bar{l}-\bar{M}+1} & z_{l+1,t-\bar{l}-\bar{M}+2} & \cdots & z_{l+1,t-\bar{l}} \\ \vdots & \vdots & \ddots & \vdots \\ z_{l+M-1,t-\bar{l}-\bar{M}+1} & z_{l+M-1,t-\bar{l}-\bar{M}+2} & \cdots & z_{l+M-1,t-\bar{l}} \end{bmatrix} \tag{4.42}$$

be the overlapping matrices of the 2-D data sequence and

$$\mathbf{z}_{l,t-\bar{l}} = \text{vec}\left\{\mathbf{Z}_{l,t-\bar{l}}\right\}, \tag{4.43}$$

where $\text{vec}\{\mathbf{X}\} = \begin{bmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T & \cdots & \mathbf{x}_K^T \end{bmatrix}^T$ with $\mathbf{x}_k$ $(1 \leq k \leq K)$ being the kth column of $\mathbf{X}$. Let

$$\mathbf{P}_t = \sum_{\bar{l}=0}^{\bar{L}-1} \sum_{l=0}^{L-1} \mathbf{z}_{l,t-\bar{l}} \mathbf{z}_{l,t-\bar{l}}^H, \tag{4.44}$$

$$\mathbf{g}_t(\omega, \bar{\omega}) = \sum_{\bar{l}=0}^{\bar{L}-1} \sum_{l=0}^{L-1} \mathbf{z}_{l,t-\bar{l}} e^{-j\{l\omega + (\bar{L}-1-\bar{l})\bar{\omega}\}}, \tag{4.45}$$

and

$$\mathbf{Q}_t(\omega, \bar{\omega}) = \mathbf{P}_t - \mathbf{g}_t(\omega, \bar{\omega}) \mathbf{g}_t^H(\omega, \bar{\omega}) / (\bar{L}L), \tag{4.46}$$

with $\otimes$ denoting the Kronecker product [13], $L = N - M + 1$, and $\bar{L} = \bar{N} - \bar{M} + 1$. Then the APES estimate of $\alpha_t(\omega, \bar{\omega})$ is given by [3]

$$\hat{\alpha}_{\text{APES},t}(\omega, \bar{\omega}) = \frac{\mathbf{a}_{M,\bar{M}}^H(\omega, \bar{\omega}) \mathbf{Q}_t^{-1}(\omega, \bar{\omega}) \mathbf{g}_t(\omega, \bar{\omega})}{\bar{L}L \mathbf{a}_{M,\bar{M}}^H(\omega, \bar{\omega}) \mathbf{Q}_t^{-1}(\omega, \bar{\omega}) \mathbf{a}_{M,\bar{M}}(\omega, \bar{\omega})}, \tag{4.47}$$

where

$$\mathbf{a}_{M,\bar{M}}(\omega, \bar{\omega}) = \mathbf{a}_{\bar{M}}(\bar{\omega}) \otimes \mathbf{a}_M(\omega). \tag{4.48}$$

4.4.2 Efficient Initialization Method

Now we consider how to efficiently calculate $\hat{\alpha}_{\text{APES},t}(\omega,\bar{\omega})$ in a non-recursive way [12]. It can be used as the initialization method for the recursive algorithm that follows or as a separate time-varying APES estimator working at the batch mode.

Let

$$\mathbf{Z}_t = \begin{bmatrix} \mathbf{z}_{0,t-\bar{L}+1} & \cdots & \mathbf{z}_{L-1,t-\bar{L}+1} & \cdots & \mathbf{z}_{0,t} & \cdots & \mathbf{z}_{L-1,t} \end{bmatrix}, \tag{4.49}$$

$$b_t(\omega,\bar{\omega}) = \mathbf{a}_{M,\bar{M}}^H(\omega,\bar{\omega})\mathbf{P}_t^{-1}\mathbf{a}_{M,\bar{M}}(\omega,\bar{\omega}), \tag{4.50}$$

$$c_t(\omega,\bar{\omega}) = \mathbf{a}_{M,\bar{M}}^H(\omega,\bar{\omega})\mathbf{P}_t^{-1}\mathbf{Z}_t\mathbf{a}_{L,\bar{L}}^*(\omega,\bar{\omega}), \tag{4.51}$$

and

$$d_t(\omega,\bar{\omega}) = \mathbf{a}_{L,\bar{L}}^T(\omega,\bar{\omega})\mathbf{Z}_t^H\mathbf{P}_t^{-1}\mathbf{Z}_t\mathbf{a}_{L,\bar{L}}^*(\omega,\bar{\omega}). \tag{4.52}$$

By using

$$\mathbf{Q}_t^{-1}(\omega,\bar{\omega}) = \mathbf{P}_t^{-1} + \frac{\mathbf{P}_t^{-1}\mathbf{g}_t(\omega,\bar{\omega})\mathbf{g}_t^H(\omega,\bar{\omega})\mathbf{P}_t^{-1}}{\bar{L}L - \mathbf{g}_t^H(\omega,\bar{\omega})\mathbf{P}_t^{-1}\mathbf{g}_t(\omega,\bar{\omega})}, \tag{4.53}$$

we can then rewrite (4.47) as

$$\hat{\alpha}_{\text{APES},t}(\omega,\bar{\omega}) = \frac{c_t(\omega,\bar{\omega})}{b_t(\omega,\bar{\omega})\left[\bar{L}L - d_t(\omega,\bar{\omega})\right] + |c_t(\omega,\bar{\omega})|^2}. \tag{4.54}$$

The remaining problem is how to calculate $b_t(\omega,\bar{\omega})$, $c_t(\omega,\bar{\omega})$, and $d_t(\omega,\bar{\omega})$ efficiently. Let $\mathbf{D}_t$ be the Hermitian square root of $\mathbf{P}_t^{-1}$, i.e.,

$$\mathbf{P}_t^{-1} = \mathbf{D}_t\mathbf{D}_t^H. \tag{4.55}$$

Then $b_t(\omega,\bar{\omega})$, $c_t(\omega,\bar{\omega})$, and $d_t(\omega,\bar{\omega})$ can be computed via 2-D FFT in the following way

$$b_t(\omega,\bar{\omega}) = \|\mathbf{F}_\alpha(\omega,\bar{\omega})\|^2, \tag{4.56}$$

$$c_t(\omega,\bar{\omega}) = \mathbf{F}_\alpha(\omega,\bar{\omega})\mathbf{F}_\beta(\omega,\bar{\omega}), \tag{4.57}$$

$$d_t(\omega,\bar{\omega}) = \|\mathbf{F}_\beta(\omega,\bar{\omega})\|^2, \tag{4.58}$$

where

$$\mathbf{F}_\alpha(\omega,\bar{\omega}) = \mathbf{a}_{M,\bar{M}}^H(\omega,\bar{\omega})\mathbf{D}_t, \tag{4.59}$$

and

$$\mathbf{F}_\beta(\omega,\bar{\omega}) = \left(\mathbf{D}_t^H \mathbf{Z}_t\right) \mathbf{a}^*_{L,\bar{L}}(\omega,\bar{\omega}). \tag{4.60}$$

Note that $\mathbf{F}_\alpha(\omega,\bar{\omega})$ and $\mathbf{F}_\beta(\omega,\bar{\omega})$ can be calculated efficiently via 2-D FFT.

4.4.3 Recursive Implementation

Similarly to the 1-D case, $\hat{\alpha}_{\text{APES},t+1}(\omega,\bar{\omega})$ can also be updated from $\hat{\alpha}_{\text{APES},t}(\omega,\bar{\omega})$ by updating $b_{t+1}(\omega,\bar{\omega})$, $c_{t+1}(\omega,\bar{\omega})$, and $d_{t+1}(\omega,\bar{\omega})$, respectively, from $b_t(\omega,\bar{\omega})$, $c_t(\omega,\bar{\omega})$, and $d_t(\omega,\bar{\omega})$.

Let $\bar{b}_{t+1}(\omega,\bar{\omega})$, $\bar{c}_{t+1}(\omega,\bar{\omega})$, and $\bar{d}_{t+1}(\omega,\bar{\omega})$ be defined similarly to $b_{t+1}(\omega,\bar{\omega})$, $c_{t+1}(\omega,\bar{\omega})$, and $d_{t+1}(\omega,\bar{\omega})$, in (4.50), (4.51), and (4.52), respectively, except that the data sequence $\{\mathbf{z}_{l,t+1-\bar{l}}, l = 0,1\cdots,L-1,\bar{l} = 0,1,\cdots,\bar{L}-1\}$ is replaced by $\{\mathbf{z}_{l,t+1-\bar{l}}, l = 0,1\cdots,L,\bar{l} = 0,1,\cdots,\bar{L}\}$. We first consider how to update $\bar{b}_{t+1}(\omega,\bar{\omega})$, $\bar{c}_{t+1}(\omega,\bar{\omega})$, and $\bar{d}_{t+1}(\omega,\bar{\omega})$ from $b_t(\omega,\bar{\omega})$, $c_t(\omega,\bar{\omega})$, and $d_t(\omega,\bar{\omega})$, respectively.

Let

$$\tilde{\mathbf{Z}}_{t+1} = \begin{bmatrix} \mathbf{z}_{0,t+1} & \mathbf{z}_{1,t+1} & \cdots & \mathbf{z}_{L-1,t+1} \end{bmatrix}, \tag{4.61}$$

$$\bar{\mathbf{P}}_{t+1} = \mathbf{P}_t + \tilde{\mathbf{Z}}_{t+1}\tilde{\mathbf{Z}}_{t+1}^H, \tag{4.62}$$

and let the $L \times L$ matrix $\mathbf{G}_{t+1}$ be the Hermitian square root of $\left(\mathbf{I} + \tilde{\mathbf{Z}}_{t+1}^H \mathbf{P}_t^{-1} \tilde{\mathbf{Z}}_{t+1}\right)^{-1}$, i.e.,

$$\mathbf{G}_{t+1}\mathbf{G}_{t+1}^H = \left(\mathbf{I} + \tilde{\mathbf{Z}}_{t+1}^H \mathbf{P}_t^{-1} \tilde{\mathbf{Z}}_{t+1}\right)^{-1}. \tag{4.63}$$

Then $\bar{\mathbf{P}}_{t+1}^{-1}$ can be updated from $\mathbf{P}_t^{-1}$ as follows

$$\bar{\mathbf{P}}_{t+1}^{-1} = \mathbf{P}_t^{-1} - \mathbf{\Phi}_{t+1}\mathbf{\Phi}_{t+1}^H, \tag{4.64}$$

where

$$\mathbf{\Phi}_{t+1} = \mathbf{P}_t^{-1}\tilde{\mathbf{Z}}_{t+1}\mathbf{G}_{t+1}. \tag{4.65}$$

Let

$$\bar{\mathbf{Z}}_{t+1} = \begin{bmatrix} \mathbf{Z}_t & \tilde{\mathbf{Z}}_{t+1} \end{bmatrix}. \tag{4.66}$$

Using (4.64) we have

$$\bar{b}_{t+1}(\omega,\bar{\omega}) = b_t(\omega,\bar{\omega}) - \|\mathbf{f}_\Phi(\omega,\bar{\omega})\|^2, \tag{4.67}$$

$$\bar{c}_{t+1}(\omega,\bar{\omega}) = c_t(\omega,\bar{\omega}) + \mathbf{f}_\Phi^T(\omega,\bar{\omega})\left\{\mathbf{G}_{t+1}^H\mathbf{a}_L^*(\omega)e^{-j\bar{L}\bar{\omega}} - \mathbf{f}_\Psi(\omega,\bar{\omega})\right\}, \tag{4.68}$$

and

$$\begin{aligned}\bar{d}_{t+1}(\omega,\bar{\omega}) &= d_t(\omega,\bar{\omega}) + \mathbf{a}_L^T(\omega)\tilde{\mathbf{Z}}_{t+1}^H\bar{\mathbf{P}}_{t+1}^{-1}\tilde{\mathbf{Z}}_{t+1}\mathbf{a}_L^*(\omega) - \|\mathbf{f}_\Psi(\omega,\bar{\omega})\|^2 + \\ &\quad +2\mathrm{Re}\left\{\mathbf{f}_\Psi^H(\omega,\bar{\omega})\mathbf{G}_{t+1}^H\mathbf{a}_L^*(\omega)e^{-j\bar{L}\bar{\omega}}\right\},\end{aligned} \tag{4.69}$$

where

$$\mathbf{f}_\Phi(\omega,\bar{\omega}) = \mathbf{\Phi}_{t+1}^T\mathbf{a}_{M,\bar{M}}^*(\omega,\bar{\omega}), \tag{4.70}$$

and

$$\mathbf{f}_\Psi(\omega,\bar{\omega}) = \mathbf{\Psi}_{t+1}\mathbf{a}_{L,\bar{L}}^*(\omega,\bar{\omega}), \tag{4.71}$$

with

$$\mathbf{\Psi}_{t+1} = \mathbf{\Phi}_{t+1}^H\mathbf{Z}_t. \tag{4.72}$$

Note that (4.70) and (4.71) can be efficiently calculated via 2-D FFT and other terms appeared in Equations (4.67) through (4.69) can be calculated via 1-D FFT and simple matrix operations.

Now consider updating $b_{t+1}(\omega,\bar{\omega})$, $c_{t+1}(\omega,\bar{\omega})$ and $d_{t+1}(\omega,\bar{\omega})$ from $\bar{b}_{t+1}(\omega,\bar{\omega})$, $\bar{c}_{t+1}(\omega,\bar{\omega})$ and $\bar{d}_{t+1}(\omega,\bar{\omega})$, respectively. From

$$\mathbf{P}_{t+1} = \bar{\mathbf{P}}_{t+1} - \tilde{\mathbf{Z}}_{t-\bar{L}+1}\tilde{\mathbf{Z}}_{t-\bar{L}+1}^H, \tag{4.73}$$

we get

$$\mathbf{P}_{t+1}^{-1} = \bar{\mathbf{P}}_{t+1}^{-1} + \bar{\mathbf{\Phi}}_{t-\bar{L}+1}\bar{\mathbf{\Phi}}_{t-\bar{L}+1}^H, \tag{4.74}$$

where

$$\bar{\mathbf{\Phi}}_{t-\bar{L}+1} = \bar{\mathbf{P}}_{t+1}^{-1}\tilde{\mathbf{Z}}_{t-\bar{L}+1}\bar{\mathbf{G}}_{t-\bar{L}+1}, \tag{4.75}$$

with the $L\times L$ matrix $\bar{\mathbf{G}}_{t-\bar{L}+1}$ being the Hermitian square root of $\left(\mathbf{I} - \tilde{\mathbf{Z}}_{t-\bar{L}+1}^H\bar{\mathbf{P}}_{t+1}^{-1}\tilde{\mathbf{Z}}_{t-\bar{L}+1}\right)^{-1}$, i.e.,

$$\bar{\mathbf{G}}_{t-\bar{L}+1}\bar{\mathbf{G}}_{t-\bar{L}+1}^H = \left(\mathbf{I} - \tilde{\mathbf{Z}}_{t-\bar{L}+1}^H\bar{\mathbf{P}}_{t+1}^{-1}\tilde{\mathbf{Z}}_{t-\bar{L}+1}\right)^{-1}. \tag{4.76}$$

Hence we have

$$b_{t+1}(\omega,\bar{\omega}) = \bar{b}_{t+1}(\omega,\bar{\omega}) + \left\|\mathbf{f}_{\bar{\Phi}}(\omega,\bar{\omega})\right\|^2, \tag{4.77}$$

$$c_{t+1}(\omega,\bar{\omega}) = \left\{\bar{c}_{t+1}(\omega,\bar{\omega}) - \mathbf{f}_{\bar{\Phi}}^T(\omega,\bar{\omega})\left[\bar{\mathbf{G}}_{t-\bar{L}+1}^H \mathbf{a}_L^*(\omega) - \mathbf{f}_{\bar{\Psi}}(\omega,\bar{\omega})\right]\right\} e^{j\bar{\omega}}, \tag{4.78}$$

and

$$\begin{aligned} d_{t+1}(\omega,\bar{\omega}) &= \bar{d}_{t+1}(\omega,\bar{\omega}) + \mathbf{a}_L^T(\omega)\tilde{\mathbf{Z}}_{t-\bar{L}+1}^H \mathbf{P}_{t+1}^{-1}\tilde{\mathbf{Z}}_{t-\bar{L}+1}\mathbf{a}_L^*(\omega) + \left\|\mathbf{f}_{\bar{\Psi}}(\omega,\bar{\omega})\right\|^2 \\ &\quad -2\mathrm{Re}\left\{\mathbf{f}_{\bar{\Psi}}^H(\omega,\bar{\omega})\bar{\mathbf{G}}_{t-\bar{L}+1}^H \mathbf{a}_L^*(\omega)\right\}, \end{aligned} \tag{4.79}$$

where

$$\mathbf{f}_{\bar{\Phi}}(\omega,\bar{\omega}) = \bar{\mathbf{\Phi}}_{t-\bar{L}+1}^T \mathbf{a}_{M,\bar{M}}^*(\omega,\bar{\omega}), \tag{4.80}$$

and

$$\mathbf{f}_{\bar{\Psi}}(\omega,\bar{\omega}) = \bar{\mathbf{\Psi}}_{t-\bar{L}+1}\mathbf{a}_{L,\bar{L}+1}^*(\omega,\bar{\omega}), \tag{4.81}$$

with

$$\bar{\mathbf{\Psi}}_{t-\bar{L}+1} = \bar{\mathbf{\Phi}}_{t-\bar{L}+1}^H \bar{\mathbf{Z}}_{t+1}. \tag{4.82}$$

From the above derivations, we note again that $\hat{\alpha}_{t+1}(\omega,\bar{\omega})$ can be updated efficiently from $\hat{\alpha}_t(\omega,\bar{\omega})$ via FFT and simple matrix operations.

4.5 Numerical and Experimental Examples

Now we present several numerical and experimental examples to illustrate the performance of the proposed APES algorithm for 1-D and 2-D time-varying complex spectral analysis. In the following examples, we choose M and $\bar{M}$ of APES, respectively, to be the nearest integers of $N/2$ and $\bar{N}/2$. We compare the performance of APES with that of the STFT and the windowed STFT approaches. For the windowed STFT method, we use the Kaiser window with parameter 3. All 1-D and 2-D sequences are zero padded to 256 and 256×256, respectively, before using FFT.

EXAMPLE 1. First consider a 1-D simulated example. The data sequence is generated as follows:

$$y_t = \sum_{i=1}^{4} e^{j2\pi f_i(t)t} + w_t, \quad t = 0, 1, \cdots, 231, \tag{4.83}$$

where $f_1(t) = -0.08 - \frac{t}{4\times10^3}$ Hz, $f_2(t) = -0.02 + \frac{t}{4\times10^3}$ Hz, $f_3(t) = -0.02 - \frac{t}{4\times10^3}$ Hz, $f_4(t) = 0.08 + \frac{t}{4\times10^3}$ Hz, and w_t is the zero-mean white complex Gaussian noise with variance $\sigma^2 = 0.01$.

Figures 4.1(a) through (d) show, respectively, the true spectrum of the signal as a function of time t and its estimates obtained by using 1-D STFT, 1-D widowed STFT, and 1-D APES with maximally overlapping short-time data sequences of length $N = 32$. It can be noted that the APES method gives much better spectral estimates than the STFT and windowed STFT methods.

EXAMPLE 2. Now consider the ISAR imaging of a simulated fast rotating MIG-25 aircraft. The 32×512 data matrix was provided to us by the Naval Research Laboratory. The two examples out of the 481 time-varying ISAR images obtained by using 2-D STFT, 2-D windowed STFT, and 2-D APES with maximally overlapping short-time 32×32 data matrices are shown in Figures 4.2(a)(d), (b)(e), and (c)(f), respectively. Note again that APES outperforms the STFT and windowed STFT methods.

EXAMPLE 3. Finally consider the ISAR imaging of a moving helicopter. The 64×512 experimental data matrix was also provided to us by the Naval Research Laboratory. The one example out of the 449 time-varying ISAR images obtained by using 2-D STFT, 2-D windowed STFT, and 2-D APES with maximally overlapping short-time 64×64 data matrices are shown in Figures 4.3(a) through (c), respectively. The conclusions drawn from this experimental example are similar to those from the previous simulated examples. Note that there are horizontal blurred strips in the images, which are caused by the fast rotating rotors of the helicopter.

The above three examples have demonstrated the estimation performance of the recursive APES algorithm. Now we consider the computational benefit of the proposed recursive APES algorithm. Since both the non-recursive (referred to as the efficient initialization

discussed in Sections 4.3.2 and 4.4.2) and recursive APES algorithms proposed in this paper fully utilize the efficient FFT for the spectral estimation, they can save a lot of computations compared to those intuitive implementation schemes, such as the one used in [4]. The larger the number of frequencies of interest in the complex spectrum, the more significant the computational savings. Below we give some quantitative results for the speed-up ratios of the recursive versus non-recursive algorithms in different scenarioes obtained by running our MATLAB codes.

For the 1-D case, we assume that $N = 2M$ and the number of frequencies of interest in the complex spectrum is denoted by $N_{\max}$. For the 2-D case, we use $N_{\max}$ and $\bar{N}_{\max}$ as the dimensions of the ISAR images and choose $N_{\max} = \bar{N}_{\max}$, $N = 2M$, $\bar{N} = 2\bar{M}$, and $M = \bar{M}$. The speed-up ratios are listed in Table 4.1. Note that the speed-up ratio goes up as M becomes larger. Also, the smaller the $N_{\max}$, the larger the speed-up ratio. This can be explained as follows. The entire computational cost of the non-recursive algorithm is mainly composed of two parts, the matrix inversion and the FFT operations. The high efficiency of FFT operations is shared by both the non-recursive and recursive algorithms and the resursive algorithm is mainly used to reduce the computational overhead of the matrix inversion. When M becomes larger or $N_{\max}$ gets smaller, the computational cost of matrix inversion becomes the dominant part and hence the speed-up ratio goes up.

Finally, we remark that like many other adaptive least-square filtering algorithms discussed in [14], the roundoff errors due to the use of finite-precision arithmetic may pose a potential problem of numerical instability after a long recursive time interval. In this case, a periodic reinitialization procedure can be used to cure this problem.

4.6 Conclusions

We have presented a computationally efficient way of implementing APES recursively for time-varying spectral analysis, which involves only FFT and simple matrix operations and great computational savings can be achieved by fully exploiting the maximally overlapped short-time data sequences. Both numerical and experimental examples have shown that the

recursive APES method can perform much better than the short-time FFT methods for ISAR iamging and feature extraction of maneuvering targets.

Reference

[1] A. W. Rihaczek and S. J. Hershkowitz, *Radar Resolution and Complex-Image Analysis.* Artech House, Inc., 1996.

[2] J. Capon, "High resolution frequency-wavenumber spectrum analysis," *Proceedings of IEEE*, vol. 57, pp. 1408–1418, August 1969.

[3] J. Li and P. Stoica, "An adaptive filtering approach to spectral estimation and SAR imaging," *IEEE Transactions on Signal Processing*, vol. 44, pp. 1469–1484, June 1996.

[4] H. Li, J. Li, and P. Stoica, "Performance analysis of forward-backward matched-filterbank spectral estimators," *IEEE Transactions on Signal Processing*, vol. 46, no. 7, pp. 1954–1966, July 1998.

[5] W. G. Carrara, R. S. Goodman, and R. M. Majewski, *Spotlight Synthetic Aperture Radar: Signal Processing Algorithms.* Norwood, MA, USA: Artech House, Inc., 1995.

[6] V. C. Chen, "Time-frequency based ISAR image formation technique," *Proceedings of SPIE:Algorithms for Synthetic Aperture Radar Imagery IV*, pp. 43–54, April 1997.

[7] L. Cohen, *Time-Frequency Analysis.* Englewood Cliffs, NJ 07632: Prentice-Hall, Inc., 1995.

[8] F. Hlawatsch and G. F. Boudreaux-Bartels, "Linear and quadratic time-frequency signal representations," *IEEE Signal Processing Magazine*, pp. 21–67, April 1992.

[9] S. Qian and D. Chen, "Decomposition of Wigner-Ville distribution and time-frequency distribution series," *IEEE Transactions on Signal Processing*, vol. 42, no. 10, pp. 2836–2842, October 1994.

[10] Z. Bao, W. Deng, and J. Yang, "A new motion compensation scheme for ISAR imaging," *Proceedings of the 1991 CIE (Chinese Institute of Electronics) International Conference on Radar, Beijing, China*, pp. 327–330, October 1991.

[11] H. Wu, G. Y. Delisle, and D. G. Fang, "Translational motion compensation in ISAR image processing," *IEEE Transactions on Image Processing*, vol. 4, no. 11, pp. 1561–1571, November 1995.

[12] Z.-S. Liu, H. Li, and J. Li, "Efficient implementation of Capon and APES for spectral estimation," *IEEE Transactions on Aerospace and Electronic Systems*, no. 4, October 1998.

[13] A. Graham, *Kronecker Products and Matrix Calculus with Applications.* Chichester, UK: Ellis Horwood Ltd., 1981.

[14] S. Haykin, *Adaptive Filter Theory.* Englewood Cliffs, NJ: Prentice-Hall, 1986.

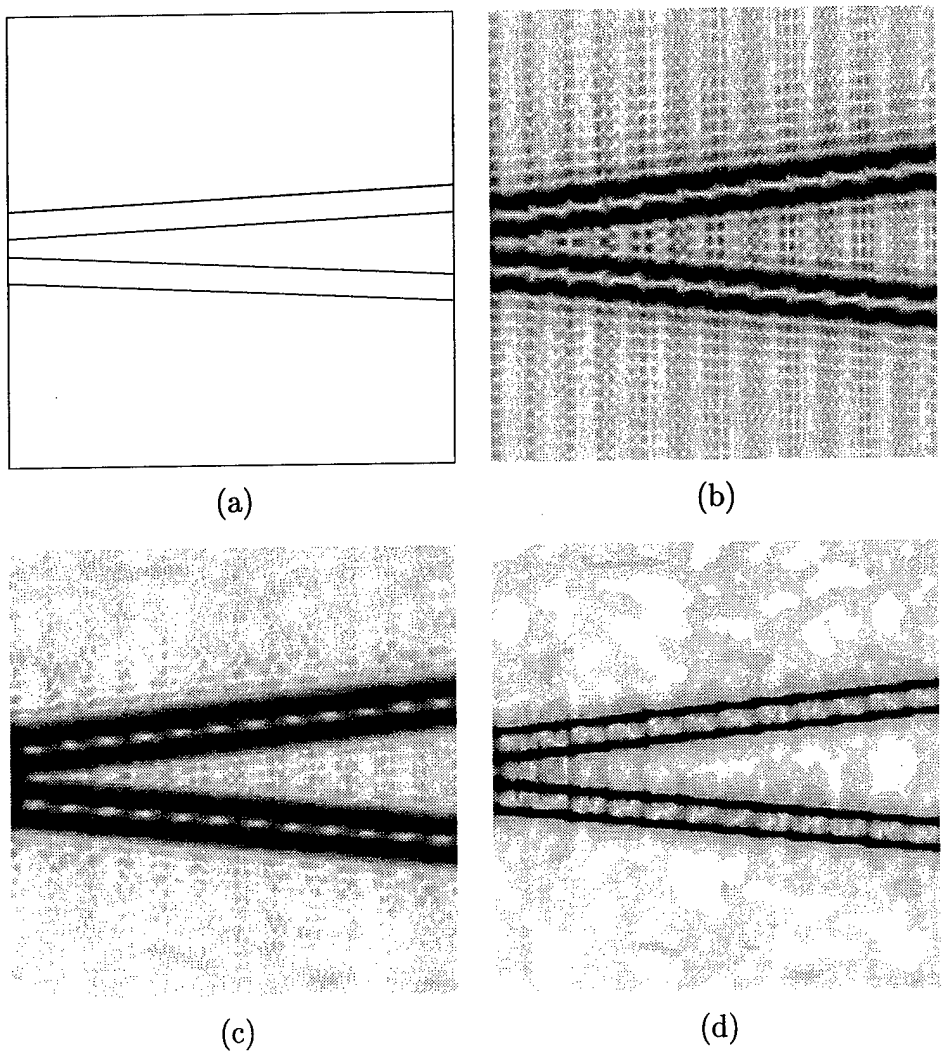


Figure 4.1: Comparison of (a) the true spectrum as a function of time with its estimates obtained by using (b) 1-D STFT, (c) 1-D windowed STFT, and (d) 1-D APES with maximally overlapping short-time data sequences of length $N = 32$.

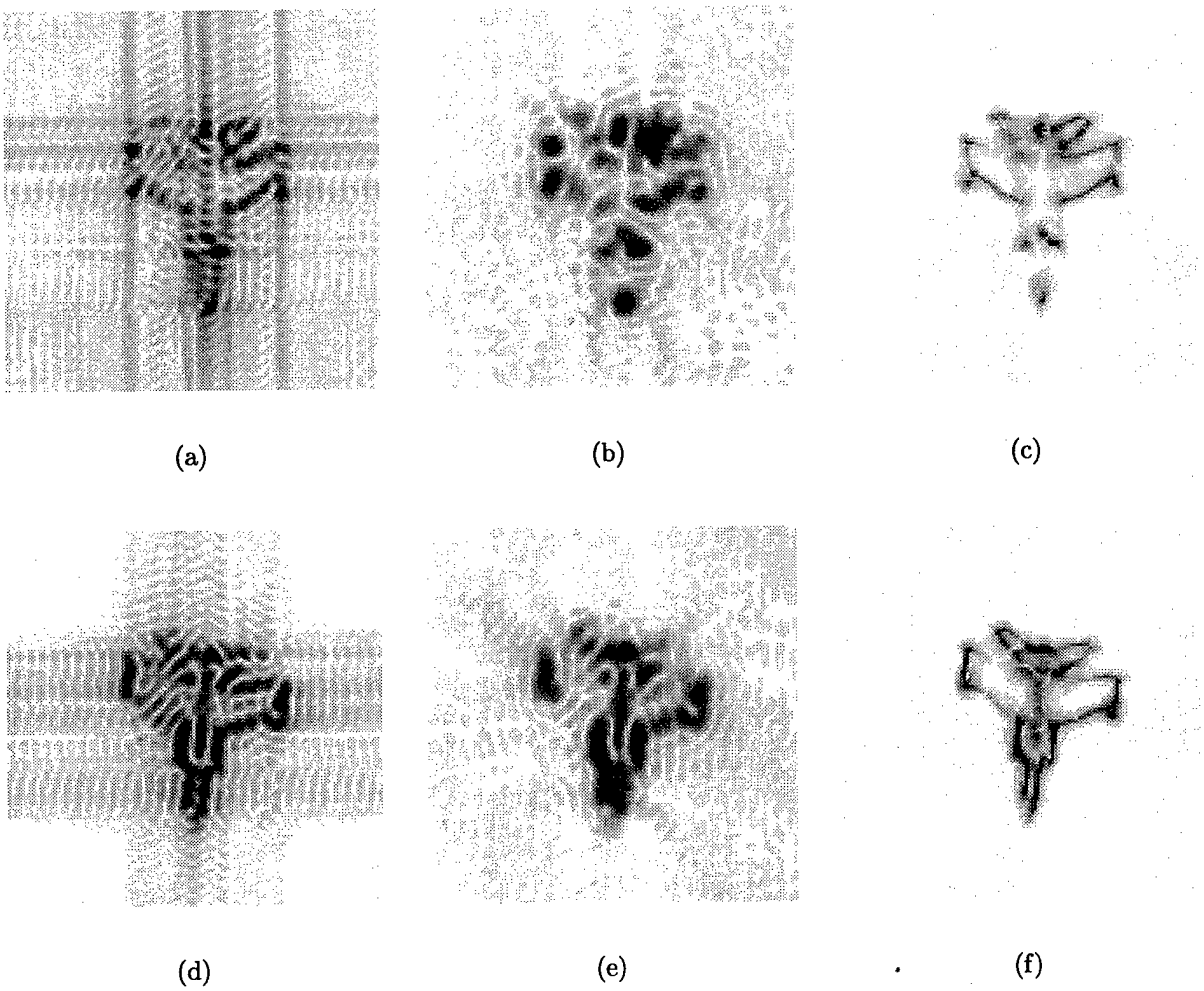
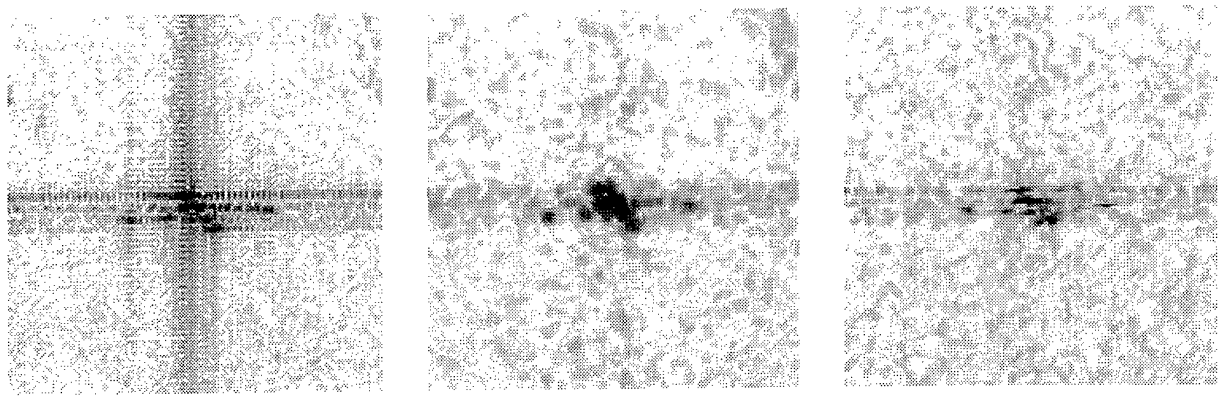


Figure 4.2: Two examples out of the 481 SAR images of a simulated moving MIG-25 airplane obtained by using (a)(d) 2-D STFT, (b)(e) 2-D windowed STFT, and (c)(f) 2-D APES with maximally overlapping short-time 32×32 data matrices.



(a) (b) (c)

Figure 4.3: One example out of the 449 SAR images of a moving helicopter obtained by using (a) 2-D STFT, (b) 2-D windowed STFT, and (c) 2-D APES with maximally overlapping short-time 64×64 data matrices.

	$N_{\max} = 32$	$N_{\max} = 64$	$N_{\max} = 128$	$N_{\max} = 256$
$M = 16$	22	17	13	10
$M = 32$		47	39	33
$M = 64$			99	88
$M = 128$				204

(a)

	$N_{\max} = 32$	$N_{\max} = 64$	$N_{\max} = 128$	$N_{\max} = 256$
$M = 16$	20	16	11	8
$M = 32$		43	39	29

(b)

Table 4.1: Speed-up ratios of recursive versus non-recursive APES algorithms for (a) the 1-D case and (b) the 2-D case.

5. A Semi-Parametric Spectral Estimation Approach to SAR Target Feature Extraction and Image Formation

5.1 Introduction

Synthetic aperture radar (SAR) image formation and target feature extraction play an important role in many applications including the battlefield awareness [1]. The conventional Fourier transform based methods are known to be computationally efficient and robust, but suffer from poor resolution, poor accuracy, and high sidelobes. Many modern spectral estimation methods have been devised and applied to target feature extraction and SAR image formation to improve resolution and accuracy, while reducing sidelobes. In [2, 3], many parametric and nonparametric spectral estimation methods are compared and discussed for their advantages and disadvantages for SAR image formation. The nonparametric methods that have been used for SAR image formation and target feature extraction include, for example, the reduced-rank variations of the Capon method [4, 5, 6], the adaptive sidelobe reduction approaches [7], and the matched-filter bank based complex spectral estimation methods [8] including the Capon [9] and APES (Amplitude and Phase EStimation) [10] methods. The parametric methods that have been considered include, for example, autoregressive (AR) model based methods [11, 12], eigendecomposition based methods [4, 5, 6, 13] including MUSIC [14] and ESPRIT [15], and nonlinear least squares (NLS) fitting based methods [16, 17]. In general, parametric algorithms outperform their nonparametric counterparts in resolution and accuracy but are more sensitive to data modeling errors.

Most of the parametric SAR target feature extraction algorithms are based on the two-dimensional (2-D) complex sinusoidal data model with constant amplitude and phase in both range and cross-range by assuming that the target consists of several trihedral corner reflectors (ideal point scatterers). This assumption is not always valid in practice. For example, for many man-made targets, especially vehicles and buildings, much of the returned energy is primarily caused by the dihedral, in addition to the trihedral, corner reflectors of the target [18]. In general, the constant amplitude and phase complex sinusoidal data model is basically valid in range. However, it is more difficult to establish a good parametric data

model in cross-range. An ideal trihedral corner reflector can be modeled as a complex sinusoid with a constant amplitude and phase in cross-range. An ideal dihedral corner reflector can be approximately modeled in cross-range as a complex sinusoid with amplitude described by a sinc ($\sin(x)/x$) function and a constant phase. In [17], a mixed data model using both constants and sinc functions in cross-range was considered and a parametric algorithm, referred to as the RELAX-NLS (RELAXation based NLS) algorithm, was presented for the feature extraction of targets consisting of both trihedral and dihedral corner reflectors. Due to the sophisticated data models used, RELAX-NLS is computationally demanding. In addition, like many other parametric spectral estimation algorithms, RELAX-NLS is not as robust as nonparametric approaches against data modeling errors.

In this chapter, instead of using the aforementioned approximate data models in cross-range, we use a more flexible data model, which models each target scatterer as a 2-D complex sinusoid with arbitrary amplitude and constant phase in cross-range and with constant amplitude and phase in range. Due to the arbitrary amplitude assumed in cross-range, the data model is essentially semi-parametric and the algorithm based on such a flexible data model is more robust against data modeling errors than parametric methods. A new algorithm, referred to as the SPAR (Semi-PARametric) algorithm, is presented for the SAR target feature extraction and high resolution image formation. By taking advantage of both parametric and nonparametric spectral estimation methods, SPAR exhibits better estimation and resolution performance over nonparametric approaches and is more robust against data modeling errors than parametric methods. By attempting to deal with one corner reflector at a time, SPAR can be used to effectively mitigate the artifact problem encountered in the high resolution SAR image formation due to the flexible data model. Another advantage of SPAR is that it can be used to obtain the initial conditions needed by other parametric algorithms, such as RELAX-NLS, to reduce the total amount of computations required to extract target features.

The remainder of this chapter is organized as follows. Section 5.2 formulates the problem of interest. Section 5.3 discusses the possible ambiguity problems of the flexible data model and their effects on SAR target feature extraction and high resolution image formation. In

Section 5.4, the SPAR algorithm is presented. A modified RELAX-NLS algorithm is presented in Section 5.5. Section 5.6 illustrates the performance of the proposed algorithm with both numerical and experimental examples. Finally, Section 5.7 contains our conclusions.

5.2 Problem Formulation

It is necessary to establish a proper data model for target scatterers to obtain super resolution SAR images of targets of interest. However, as pointed out in the previous section, it is difficult to establish a good parametric model in cross-range for target scatterers. Instead, we model herein the received signal reflected from a target scatterer as:

$$s(n,\bar{n}) = x(\bar{n})e^{j\phi}e^{j2\pi(fn+\bar{f}\bar{n})}, \quad n = 0,1,\cdots,N-1, \quad \bar{n} = 0,1,\cdots,\bar{N}-1, \tag{5.1}$$

where N and $\bar{N}$ denote the dimensions of the available data samples in range and cross-range, respectively; $x(\bar{n})$ is an arbitrary unknown real-valued function of $\bar{n}$ determined by the radar cross section (RCS) of the scatterer; ϕ is a constant phase; finally, $\{f,\bar{f}\}$ is the frequency pair proportional to the 2-D location (range and cross-range) of the scatterer. This data model is essentially semi-parametric since little parameterization is done in cross-range.

Assume that a target consists of K scatterers. Then the target data model in the presence of noise has the form:

$$y(n,\bar{n}) = \sum_{k=1}^{K} x_k(\bar{n})e^{j\phi_k}e^{j2\pi(f_k n+\bar{f}_k\bar{n})} + e(n,\bar{n}), \quad n = 0,1,\cdots,N-1, \quad \bar{n} = 0,1,\cdots,\bar{N}-1, \tag{5.2}$$

where $\{x_k(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}$ denotes the real-valued amplitude function of $\bar{n}$ for the kth scatterer; ϕ_k and $\{f_k,\bar{f}_k\}$, respectively, are the constant phase and the frequency pair of the kth scatterer; finally, $\{e(n,\bar{n})\}$ denotes the unknown 2-D noise and clutter sequence.

Since SAR images are often used in SAR applications [19, 20, 21], our problem of interest herein is to estimate the target parameters $\{\phi_k,\{x_k(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1},f_k,\bar{f}_k\}_{k=1}^{K}$ from the 2-D data sequence $\{y(n,\bar{n})\}$ and then to form high resolution SAR images with the estimated target parameters.

5.3 Effects of the Semi-Parametric Data Model on SAR Image Formation

Target feature extraction methods devised based on the data model in (5.2) are robust against data modeling errors due to the model flexibility. However, there are ambiguity problems associated with the semi-parametric data model. In this section, we first analyze the possible data model ambiguities and then illustrate their effects on SAR target feature extraction and image formation. The discussions below will motivate the introduction of the SPAR algorithm, which will be presented in detail in Section 5.4.

5.3.1 Model Ambiguities

Due to the flexibility of the data model in (5.2), there are various types of ambiguities that may impact the feature extraction of each scatterer. Below we list several types of the ambiguities inherent in the data model.

Type 1: Single scatterer

From (5.1), we note that ambiguity exists between ϕ and $x(\bar{n})$ since

$$x(\bar{n})e^{j\phi} = -x(\bar{n})e^{j(\phi+\pi)} \triangleq \tilde{x}(\bar{n})e^{j\tilde{\phi}}, \tag{5.3}$$

where $\tilde{x}(\bar{n}) = -x(\bar{n})$ and $\tilde{\phi} = \phi + \pi$. Ambiguity also exists between $\bar{f}$ and $x(\bar{n})$ since

$$x(\bar{n})e^{j2\pi\bar{f}\bar{n}} = (-1)^{\bar{n}}x(\bar{n})e^{j2\pi(\bar{f}-0.5)\bar{n}} \triangleq \tilde{x}(\bar{n})e^{j2\pi\tilde{\bar{f}}\bar{n}}, \tag{5.4}$$

where $\tilde{x}(\bar{n}) = (-1)^{\bar{n}}x(\bar{n})$ and $\tilde{\bar{f}} = \bar{f} - 0.5$. The above two types of ambiguities cannot be resolved.

Type 2: Two identical scatterers located in the same range

Let $\bar{f}_a$ and $\bar{f}_b$, respectively, denote the cross-range locations of two identical scatterers and let ϕ_a and ϕ_b, respectively, denote their phases. Then

$$x(\bar{n})\left[\mathrm{e}^{(j\phi_a+j2\pi\bar{f}_a\bar{n})} + \mathrm{e}^{(j\phi_b+j2\pi\bar{f}_b\bar{n})}\right] = 2x(\bar{n})\cos\left[\pi(\bar{f}_a - \bar{f}_b)\bar{n} + \frac{\phi_a - \phi_b}{2}\right]\mathrm{e}^{j\frac{\phi_a+\phi_b}{2}}\mathrm{e}^{j\pi(\bar{f}_a+\bar{f}_b)\bar{n}}, \tag{5.5}$$

which indicates that two identical corner reflectors (trihedrals or dihedrals) located in the same range but different cross-range positions $\bar{f}_a$ and $\bar{f}_b$, respectively, can be modeled by (5.1)

as one "scatterer" located at $(\bar{f}_a + \bar{f}_b)/2$ in cross-range with $x(\bar{n})$ modulated by $2\cos[\pi(\bar{f}_a - \bar{f}_b)\bar{n} + (\phi_a - \phi_b)/2]$. Thus the data model in (5.1) cannot be used to describe each of the two corner reflectors in this case.

Type 3: Two different scatterers located in the same range

Assume that two different corner reflectors with parameters $\{\phi_i, \{x_i(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}, f, \bar{f}_i\}_{i=1}^{2}$ are located in the same range. Then the target model in the absence of noise has the form:

$$r(n, \bar{n}) = \sum_{i=1}^{2} x_i(\bar{n}) e^{j\phi_i} e^{j2\pi(fn + \bar{f}_i \bar{n})}, \quad n = 0, 1, \cdots, N-1, \quad \bar{n} = 0, 1, \cdots, \bar{N}-1. \tag{5.6}$$

With straightforward calculations, we can rewrite (5.6) as

$$r(n, \bar{n}) = \sum_{i=1}^{2} \tilde{x}_i(\bar{n}) e^{j\tilde{\phi}_i} e^{j2\pi(\tilde{f}_i n + \tilde{\bar{f}}_i \bar{n})}, \tag{5.7}$$

where $\tilde{\phi}_2 = \tilde{\phi}_1 + \pi/2$ with $\tilde{\phi}_1$ denoting an arbitrary phase, $\tilde{f}_1 = \tilde{f}_2 = f$, $\tilde{\bar{f}}_1 = \tilde{\bar{f}}_2 = \tilde{\bar{f}}$ with $\tilde{\bar{f}}$ denoting an arbitrary cross-range location,

$$\tilde{x}_1(\bar{n}) = \sum_{i=1}^{2} x_i(\bar{n}) \cos[2\pi(\bar{f}_i - \tilde{\bar{f}})\bar{n} + (\phi_i - \tilde{\phi}_1)], \tag{5.8}$$

and

$$\tilde{x}_2(\bar{n}) = \sum_{i=1}^{2} x_i(\bar{n}) \sin[2\pi(\bar{f}_i - \tilde{\bar{f}})\bar{n} + (\phi_i - \tilde{\phi}_1)]. \tag{5.9}$$

Note from (5.7) that $\{\tilde{\phi}_i, \{\tilde{x}_i(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}, \tilde{f}_i, \tilde{\bar{f}}_i\}_{i=1}^{2}$ are the ambiguous features of $\{\phi_i, \{x_i(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}, f, \bar{f}_i\}_{i=1}^{2}$.

Type 4: Multiple scatterers located in the same range

When more than two scatterers are located in the same range, the data model in the absence of noise can still be written as (5.7) except that

$$\tilde{x}_1(\bar{n}) = \sum_{i=1}^{L} x_i(\bar{n}) \cos[2\pi(\bar{f}_i - \tilde{\bar{f}})\bar{n} + (\phi_i - \tilde{\phi}_1)], \tag{5.10}$$

and

$$\tilde{x}_2(\bar{n}) = \sum_{i=1}^{L} x_i(\bar{n}) \sin[2\pi(\bar{f}_i - \tilde{\bar{f}})\bar{n} + (\phi_i - \tilde{\phi}_1)], \tag{5.11}$$

where $L > 2$ denotes the number of scatterers located in the same range. Thus the L scatterers located in the same range are considered as two "scatterers" when using the data model in (5.2).

Before we discuss the impact of the model ambiguities on SAR image formation, we first describe how the image formation is done if we have the estimated model parameters.

5.3.2 Image Formation

Assume for now that we have extracted the target features based on any of the ambiguous data models. For notational convenience, we will use the notation used in (5.2). Since the target data model in range is a sum of several complex sinusoids with constant amplitudes and phases, we can use the estimated sinusoidal parameters to simulate a data matrix with a larger dimension in range and then use FFT to demonstrate the super resolution property of the feature extraction algorithm we shall present. Yet we cannot extrapolate the estimate $\{\hat{x}_k(\bar{n})\}$ of $\{x_k(\bar{n})\}$ since it is assumed to be an arbitrary unknown real-valued function of $\bar{n}$ and hence FFT cannot be used to obtain SAR images with enhanced resolution in cross-range. Instead, we use 1-D APES [8, 10, 22] in cross-range when forming SAR images. APES is a nonparametric complex spectral estimator making use of adaptive finite impulse response (FIR) filterbanks to suppress interference and noise. APES belongs to the class of matched filterbank spectral estimators and provides lower sidelobes, narrower spectral peaks, and more accurate spectral estimates than FFT.

Let $\{\hat{s}_s(n,\bar{n})\}$ denote the simulated data sequence with a larger dimension in range based on the estimated target features $\{\hat{\phi}_k, \{\hat{x}_k(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}, \hat{f}_k, \hat{\bar{f}}_k\}_{k=1}^{\hat{K}}$ of $\{\phi_k, \{x_k(\bar{n})\}_{\bar{n}=0}^{\bar{N}-1}, f_k, \bar{f}_k\}_{k=1}^{\hat{K}}$ where $\hat{K}$ denotes the estimate of the scatterer number K. Then

$$\hat{s}_s(n,\bar{n}) = \sum_{k=1}^{\hat{K}} \hat{x}_k(\bar{n}) e^{j\hat{\phi}_k} e^{j2\pi(\hat{f}_k n + \hat{\bar{f}}_k \bar{n})}, \quad n = 0, 1, \cdots, \zeta N - 1, \quad \bar{n} = 0, 1, \cdots, \bar{N} - 1, \quad (5.12)$$

where ζ denotes an extrapolation factor ($\zeta > 1$) and is a parameter of user choice. Note that the super resolution property of the so-formed SAR images is determined by the feature extraction algorithm and $\zeta > 1$ is only used to demonstrate the super resolution property of

the target feature extraction algorithm. The estimated noise and clutter data matrix is

$$\hat{e}(n,\bar{n}) = y(n,\bar{n}) - \hat{s}_s(n,\bar{n}), \quad n = 0,1,\cdots,N-1, \quad \bar{n} = 0,1,\cdots,\bar{N}-1, \tag{5.13}$$

which is also important in many SAR applications since, for example, important target information such as the target shadow information is contained in $\hat{e}(n,\bar{n})$. We cannot extrapolate $\hat{e}(n,\bar{n})$ in either range or cross-range since no parametric data model is available for $\hat{e}(n,\bar{n})$.

To obtain SAR images with low sidelobes in range via 1-D FFT, we apply 1-D windows to $\hat{s}_s(n,\bar{n})$ and $\hat{e}(n,\bar{n})$ in range. We obtain a new $(\zeta N) \times \bar{N}$ data matrix $\tilde{\mathbf{Y}}$ as follows:

$$\begin{cases} \tilde{y}(n,\bar{n}) = \hat{s}_s(n,\bar{n})w_s(n) + \zeta\hat{e}(n,\bar{n})w_e(n), & n = 0,1,\cdots,N-1, \quad \bar{n} = 0,1,\cdots,\bar{N}-1, \\ \tilde{y}(n,\bar{n}) = \hat{s}_s(n,\bar{n})w_s(n), & n = N,N+1,\cdots,\zeta N-1, \quad \bar{n} = 0,1,\cdots,\bar{N}-1, \end{cases} \tag{5.14}$$

where $\tilde{y}(n,\bar{n})$ denotes the $(n,\bar{n})$th element of $\tilde{\mathbf{Y}}$ and $w_s(n)$ and $w_e(n)$ are 1-D windows of lengths ζN and N, respectively, satisfying

$$\sum_{n=0}^{\zeta N-1} w_s(n) = \zeta N, \tag{5.15}$$

and

$$\sum_{n=0}^{N-1} w_e(n) = N. \tag{5.16}$$

The window functions $w_s(n)$ and $w_e(n)$ are selected according to the desired sidelobe levels. Note that scaling $\hat{e}(n,\bar{n})$ in $\tilde{y}(n,\bar{n})$ by a factor of ζ is necessary since the range dimension of $\hat{s}_s(n,\bar{n})$ is ζ times of that of $\hat{e}(n,\bar{n})$. The steps needed for SAR image formation are as follows:

Step (1): Form $\tilde{\mathbf{Y}}$ from $\tilde{y}(n,\bar{n})$ by using (5.12), (5.13), and (5.14).

Step (2): Apply the normalized 1-D FFT to each column of $\tilde{\mathbf{Y}}$ to obtain an intermediate matrix and then apply 1-D APES to each row of the intermediate matrix. (See [22] for the efficient implementation of APES.) Note that the normalized 1-D FFT has the form

$$\frac{1}{\zeta N}\sum_{n=0}^{\zeta N-1} \tilde{y}(n,\bar{n})e^{-j2\pi fn}, \quad \bar{n} = 0,1,\cdots,\bar{N}-1. \tag{5.17}$$

5.3.3 Model Ambiguity Effects on SAR Image Formation

All of the aforementioned types of ambiguities will have no effect on the SAR image formation if no parameter estimation errors exist since the scatterers will be perfectly reconstructed by using any of the possible ambiguous data models. For example, when there are two identical scatterers located in the same range, the data model in (5.2) can still be used for SAR image formation since the two scatterers are now described as one "scatterer" with (5.5), which still fits the data model of (5.2) with $K = 1$. Hence the original SAR image can still be reconstructed by using the parameters of the one "scatterer" described by the right side of (5.5).

In the presence of parameter estimation errors due to the presence of noise and clutter, however, Types 1 and 2 ambiguities discussed in Section 5.3.1 will have little effect on SAR image formation, whereas Types 3 and 4 ambiguities can result in artifact problems for the high resolution SAR image formation. Generally speaking, the higher the signal-to-noise ratio (SNR), the more accurate the parameter estimates and hence the less significant the artifact problem. The effect of Type 3 ambiguity on SAR image formation in the presence of estimation errors is demonstrated by comparing Figures 5.1 and 5.2. (The effects are similar for Type 4 ambiguity.) Figure 5.1 is obtained by assuming no parameter estimation errors. Figure 5.1(a) shows the FFT image of a target consisting of two dihedrals of different lengths located in the same range. We use

$$x_k(\bar{n}) = \alpha_k \text{sinc}[b_k \pi(\bar{n} - \tau_k)], \quad k = 1, 2, \quad \bar{n} = 0, 1, \cdots 31, \tag{5.18}$$

to simulate the dihedrals, where α_k and b_k, respectively, are proportional to the maximal RCS and the length of the kth dihedral corner reflector and τ_k denotes the peak location of the data sequence and is determined by the orientation of the kth dihedral. The size of the simulated data matrix is 32×32 (i.e., $N = \bar{N} = 32$). The parameters for the two dihedrals are given in Table 5.1. An ambiguous set of target features can be obtained by choosing $\tilde{f}_1 = \tilde{f}_2 = 0.1$, $\tilde{\bar{f}}_1 = \tilde{\bar{f}}_2 = \tilde{\bar{f}} = 0.2$, $\tilde{\phi}_1 = 0$, and $\tilde{\phi}_2 = \pi/2$ in (5.8) and (5.9). The windowed FFT SAR images of the two "scatterers" are shown in Figures 5.1(b) and (c), respectively, which differ considerably from the two dihedral scatterers in Figure 5.1(a). The combined

k	α_k	ϕ_k	f_k	$\bar{f}_k$	b_k	τ_k
k=1	9.6	0	0.1	-0.3	0.3	18.6
k=2	6.4	0	0.1	0.1	0.2	18.6

Table 5.1: True parameters of the two dihedrals used in Figures 5.1 and 5.2.

SAR image of the two "scatterers" is given in Figure 5.1(d), which is exactly the same as the true image shown in Figure 5.1(a). However, due to the presence of noise and clutter, parameter estimation errors are inevitable. The errors in range are the main cause of the artifact problem in the high resolution SAR image formation. In Figure 5.2, we assume that all parameters are accurate except that $\hat{\tilde{f}}_2 = \tilde{f}_2 + 0.01 = 0.11$. Figures 5.2(a) and (b), respectively, show the windowed FFT images of the two aforementioned "scatterers" in the presence of estimation errors and Figure 5.2(c) shows the combined SAR image. By comparing Figures 5.1(a) and 5.2(c), we note that an extra line (artifact) shows up next to the short dihedral. The reason is that due to the estimation errors, $\hat{\tilde{f}}_1 \neq \hat{\tilde{f}}_2$. Hence the two "scatterers" in Figures 5.2(a) and (b) are not exactly in the same range and cannot be "combined" perfectly to obtain the two dihedral lines in Figure 5.1(a). This problem becomes even worse when SAR images are formed via data extrapolation in range. The larger the extrapolation factor ζ, the more significant the artifact problem since the difference between $\hat{\tilde{f}}_1$ and $\hat{\tilde{f}}_2$ is exaggerated ζ times. Figures 5.2(d) shows the SAR image obtained with $\zeta = 2$. By comparing Figures 5.2(d) and (c) (here $\zeta = 1$ and hence no data extrapolation), we note that the artifact next to the short dihedral becomes more significant. Severe artifacts may exist at low SNR since the accuracy of the parameter estimates is poor. The SPAR algorithm we present below attempts to avoid this problem by using windows to isolate the multiple scatterers located in the same range.

5.4 The SPAR Algorithm

Figure 5.3 shows the block diagram for the SPAR algorithm, which can be summarized by the following two steps:

Step 1: Scatterer Isolation based Target Feature Extraction: See Section 5.4.1 below for details.

Step 2: SAR Image Formation: See Section 5.3.2 for details, where the estimated target features are obtained by using Step 1.

5.4.1 Target Feature Extraction

The basic idea behind SPAR is to extract the features of each scatterer separately. Before we present the target feature extraction algorithm, we first summarize the steps needed for the feature extraction of a single scatterer as a preparation. The generalized Akaike information criterion is also introduced to estimate K, the number of scatterers, at the end of this subsection.

Feature Extraction of a Single Scatterer

The data model of a single scatterer in the presence of noise has the form:

$$y_s(n,\bar{n}) = s(n,\bar{n}) + e_s(n,\bar{n}), \quad n = 0,1,\cdots,N-1, \quad \bar{n} = 0,1,\cdots,\bar{N}-1, \tag{5.19}$$

where $s(n,\bar{n})$ is given in (5.1) and $\{e_s(n,\bar{n})\}$ denotes the unknown 2-D noise and clutter sequence. Let

$$\boldsymbol{\omega}_N(f) = \begin{bmatrix} 1 & e^{j2\pi f} & \cdots & e^{j2\pi f(N-1)} \end{bmatrix}^T, \tag{5.20}$$

and

$$\boldsymbol{\omega}_{\bar{N}}(\bar{f}) = \begin{bmatrix} 1 & e^{j2\pi \bar{f}} & \cdots & e^{j2\pi \bar{f}(\bar{N}-1)} \end{bmatrix}^T, \tag{5.21}$$

where $(\cdot)^T$ denotes the transpose. Let $\mathbf{D}(\bar{f})$ denote the following diagonal matrix:

$$\mathbf{D}(\bar{f}) = \text{diag}\left\{ 1, \quad e^{j2\pi \bar{f}}, \quad \cdots, \quad e^{j2\pi \bar{f}(\bar{N}-1)} \right\}. \tag{5.22}$$

Define

$$\mathbf{x} = \begin{bmatrix} x(0) & x(1) & \cdots & x(\bar{N}-1) \end{bmatrix}^T. \tag{5.23}$$

Let $\mathbf{Y}_s$ be an $N \times \bar{N}$ matrix with its $(n, \bar{n})$th element being $y_s(n, \bar{n})$. Then we can rewrite (5.19) as:

$$\mathbf{Y}_s = e^{j\phi}\mathbf{G}(\mathbf{x}, f, \bar{f}) + \mathbf{E}_s, \tag{5.24}$$

where

$$\mathbf{G}(\mathbf{x}, f, \bar{f}) = \boldsymbol{\omega}_N(f)\mathbf{x}^T\mathbf{D}(\bar{f}), \tag{5.25}$$

and $\mathbf{E}_s$ denotes an $N \times \bar{N}$ matrix with $e_s(n, \bar{n})$ being its $(n, \bar{n})$th element. Let $\mathbf{y}_{s\bar{n}}$, $\bar{n} = 0, 1, \cdots, \bar{N}-1$, denote the $\bar{n}$th column of $\mathbf{Y}_s$ and define

$$\bar{\mathbf{y}}_s(f) = \mathbf{Y}_s^T\boldsymbol{\omega}_N^*(f), \tag{5.26}$$

where $(\cdot)^*$ denotes the complex conjugate. Then the NLS estimates $\left\{\hat{\mathbf{x}}, \hat{\phi}, \hat{f}, \hat{\bar{f}}\right\}$ of $\left\{\mathbf{x}, \phi, f, \bar{f}\right\}$ are (see Appendix A for the detailed derivations):

$$\hat{\mathbf{x}} = \frac{1}{N}\mathrm{Re}\left[e^{-j\phi}\bar{\mathbf{y}}_s(f) \odot \boldsymbol{\omega}_{\bar{N}}^*(\bar{f})\right], \tag{5.27}$$

where $\mathrm{Re}(x)$ denotes the real part of x and $\odot$ denotes the Hadamard matrix product or the element-wise product of two matrices;

$$\hat{\phi} = \frac{1}{2}\arg\left\{\sum_{\bar{n}=0}^{\bar{N}-1}\left[\mathbf{y}_{s\bar{n}}^T\boldsymbol{\omega}_N^*(f)\right]^2 e^{-j2\pi(2\bar{f})\bar{n}}\right\}, \tag{5.28}$$

where $\arg(x)$ denotes the argument of a complex variable x; finally,

$$\{\hat{f}, \hat{\bar{f}}\} = \arg\max_{f,\bar{f}} C_4(f, \bar{f}), \tag{5.29}$$

where

$$C_4(f, \bar{f}) = \sum_{\bar{n}=0}^{\bar{N}-1}\left|\mathbf{y}_{s\bar{n}}^T\boldsymbol{\omega}_N^*(f)\right|^2 + \left|\sum_{\bar{n}=0}^{\bar{N}-1}\left[\mathbf{y}_{s\bar{n}}^T\boldsymbol{\omega}_N^*(f)\right]^2 e^{-j2\pi(2\bar{f})\bar{n}}\right|. \tag{5.30}$$

The steps needed to obtain the NLS estimates of a single scatterer are summarized as follows:

Step (I): Use (5.26) to obtain $\bar{\mathbf{y}}_s(f)$ and obtain the cost function $C_4(f, \bar{f})$ according to (5.30). Determine $\{\hat{f}, \hat{\bar{f}}\}$ by maximizing $C_4(f, \bar{f})$ using the method given in Appendix A.

Step (II): Calculate $\hat{\phi}$ according to (5.28) with $\{f, \bar{f}\}$ replaced by $\{\hat{f}, \hat{\bar{f}}\}$ obtained in Step (I).

Step (III): Calculate $\hat{\mathbf{x}}$ via (5.27) with $\{\phi, f, \bar{f}\}$ replaced by $\{\hat{\phi}, \hat{f}, \hat{\bar{f}}\}$ determined in Steps (I) and (II), respectively.

Feature Extraction of Multiple Scatterers

When a target consists of multiple scatterers, we can obtain the NLS estimates of the target features based on (5.2) by using a relaxation-based optimization approach. Let

$$\mathbf{x}_k = \begin{bmatrix} x_k(0) & x_k(1) & \cdots & x_k(\bar{N}-1) \end{bmatrix}^T, \tag{5.31}$$

and let $\mathbf{Y}$ and $\mathbf{E}$ be $N \times \bar{N}$ matrices with their $(n, \bar{n})$th elements being $y(n, \bar{n})$ and $e(n, \bar{n})$, respectively. Then we can rewrite (5.2) as

$$\mathbf{Y} = \sum_{k=1}^{K} e^{j\phi_k} \mathbf{G}_k(\mathbf{x}_k, f_k, \bar{f}_k) + \mathbf{E}, \tag{5.32}$$

where $\mathbf{G}_k(\mathbf{x}_k, f_k, \bar{f}_k)$ has the same form as the $\mathbf{G}(\mathbf{x}, f, \bar{f})$ in (5.25) except that $\mathbf{x}$, f, and $\bar{f}$ are replaced by $\mathbf{x}_k$, f_k, and $\bar{f}_k$, respectively. Let $\mathbf{y}_{\bar{n}}$, $\bar{n} = 0, 1, \cdots, \bar{N}-1$, be the $\bar{n}$th column of $\mathbf{Y}$. Then the estimates $\{\hat{\phi}_k, \hat{\mathbf{x}}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=1}^{\hat{K}}$ of $\{\phi_k, \mathbf{x}_k, f_k, \bar{f}_k\}_{k=1}^{K}$ can be obtained by minimizing the following NLS cost function:

$$C_5\left(\{\phi_k, \mathbf{x}_k, f_k, \bar{f}_k\}_{k=1}^{K}\right) = \left\| \mathbf{Y} - \sum_{k=1}^{K} e^{j\phi_k} \mathbf{G}_k(\mathbf{x}_k, f_k, \bar{f}_k) \right\|_F^2, \tag{5.33}$$

where $\|\cdot\|_F$ denotes the Frobenius norm [23]. The minimization of $C_5\left(\{\phi_k, \mathbf{x}_k, f_k, \bar{f}_k\}_{k=1}^{K}\right)$ in (5.33) is a very complicated optimization problem. The proposed SPAR algorithm performs a complete relaxation-based search by letting only the parameters of one scatterer vary at a time while freezing the parameter values of all other scatterers ($\bar{K}-1$ in number) at their most recently determined values for each assumed number of scatterers $\bar{K}$. Let

$$\mathbf{Y}_k = \mathbf{Y} - \sum_{i=1, i\neq k}^{\bar{K}} e^{j\hat{\phi}_i} \mathbf{G}_i(\hat{\mathbf{x}}_i, \hat{f}_i, \hat{\bar{f}}_i), \tag{5.34}$$

and assume that $\{\hat{\phi}_i, \hat{\mathbf{x}}_i, \hat{f}_i, \hat{\bar{f}}_i\}_{i=1, i\neq k}^{\bar{K}}$ are given. Then the NLS estimates $\{\hat{\phi}_k, \hat{\mathbf{x}}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=1}^{\hat{K}}$ of $\{\phi_k, \mathbf{x}_k, f_k, \bar{f}_k\}_{k=1}^{K}$ can be obtained by minimizing $C_6(\phi_k, \mathbf{x}_k, f_k, \bar{f}_k)$, where

$$C_6(\phi_k, \mathbf{x}_k, f_k, \bar{f}_k) = \left\| \mathbf{Y}_k - e^{j\phi_k} \mathbf{G}_k(\mathbf{x}_k, f_k, \bar{f}_k) \right\|_F^2, \tag{5.35}$$

and using the method presented in the previous subsection for the feature extraction of a single scatterer. However, when multiple scatterers are located in the same range, the minimization of $C_6(\phi_k, \mathbf{x}_k, f_k, \bar{f}_k)$ has numerous ambiguous solutions that may lead to the artifact problem in the high resolution SAR image formation.

SPAR attempts to avoid the ambiguity problem by isolating out the most dominant scatterer in $\mathbf{Y}_k$ by using a 2-D rectangular window, which is determined from and applied to the 2-D FFT of $\mathbf{Y}_k$. The isolation process has the following steps:

Step (i): Obtain $\mathbf{V}_k$, the 2-D FFT of $\mathbf{Y}_k$, without zero padding.

Step (ii): Determine the 2-D window $w(n, \bar{n})$ from $\mathbf{V}_k$. We first locate the peak location $(n^+, \bar{n}^+)$ of the magnitude of $\mathbf{V}_k$. We then fix $\bar{n}$ to $\bar{n}^+$ and search for the interval $n_1 \leq n^+ \leq n_2$ so that the magnitude of $\mathbf{V}_k$ is above a certain threshold, say T_i, within the interval. Similarly, we can fix n to n^+ and search for the interval $\bar{n}_1 \leq \bar{n}^+ \leq \bar{n}_2$. Then the $N \times \bar{N}$ rectangular window $w(n, \bar{n})$ has unit value for $n_1 \leq n \leq n_2$ and $\bar{n}_1 \leq \bar{n} \leq \bar{n}_2$ and zero value elsewhere. The threshold T_i we use in our numerical and experimental examples is 10% of the peak value of the magnitude of $\mathbf{V}_k$.

Step (iii): Determine $\hat{\mathbf{Y}}_k$ by applying 2-D inverse FFT (IFFT) to $\mathbf{V}_k \odot \mathbf{W}$, where the $(n, \bar{n})$th element of $\mathbf{W}$ is $w(n, \bar{n})$.

Instead of minimizing $C_6(\phi_k, \mathbf{x}_k, f_k, \bar{f}_k)$, we now minimize

$$C_7(\phi_k, \mathbf{x}_k, f_k, \bar{f}_k) == \left\| \hat{\mathbf{Y}}_k - e^{j\phi_k} \mathbf{G}_k(\mathbf{x}_k, f_k, \bar{f}_k) \right\|_F^2, \tag{5.36}$$

where $\hat{\mathbf{Y}}_k$ is used to replace $\mathbf{Y}_k$ in $C_6(\phi_k, \mathbf{x}_k, f_k, \bar{f}_k)$, by using the method presented in Section 5.4.1.

With the above preparations, now we provide the steps of the scatterer isolation and relaxation based optimization algorithm, which are the substeps of Step 1 of SPAR.

Step I: Assume $\bar{K} = 1$. Calculate $\hat{\mathbf{Y}}$ from $\mathbf{Y}$ by using the isolation process. Obtain $\{\hat{\phi}_k, \hat{\mathbf{x}}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=1}$ from $\hat{\mathbf{Y}}$.

Step II: Assume $\bar{K} = 2$. Compute $\mathbf{Y}_2$ with (5.34) by using $\{\hat{\phi}_k, \hat{\mathbf{x}}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=1}$ obtained in Step I. Calculate $\hat{\mathbf{Y}}_2$ from $\mathbf{Y}_2$. Obtain $\{\hat{\phi}_k, \hat{\mathbf{x}}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=2}$ from $\hat{\mathbf{Y}}_2$. Next compute $\mathbf{Y}_1$ with (5.34) by using $\{\hat{\phi}_k, \hat{\mathbf{x}}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=2}$, calculate $\hat{\mathbf{Y}}_1$ from $\mathbf{Y}_1$, and then redetermine $\{\hat{\phi}_k, \hat{\mathbf{x}}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=1}$ from $\hat{\mathbf{Y}}_1$.

Iterate the previous two substeps until "practical convergence" is achieved (to be discussed later on).

Step III: Assume $\bar{K} = 3$. Compute $\mathbf{Y}_3$ with (5.34) by using $\{\hat{\phi}_k, \hat{\mathbf{x}}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=1}^{2}$ obtained in Step II. Calculate $\hat{\mathbf{Y}}_3$ from $\mathbf{Y}_3$. Obtain $\{\hat{\phi}_k, \hat{\mathbf{x}}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=3}$ from $\hat{\mathbf{Y}}_3$. Next, compute $\mathbf{Y}_1$ with (5.34) by using $\{\hat{\phi}_k, \hat{\mathbf{x}}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=2}^{3}$, calculate $\hat{\mathbf{Y}}_1$ from $\mathbf{Y}_1$, and then redetermine $\{\hat{\phi}_k, \hat{\mathbf{x}}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=1}$ from $\hat{\mathbf{Y}}_1$. Then compute $\mathbf{Y}_2$ with (5.34) by using $\{\hat{\phi}_k, \hat{\mathbf{x}}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=1,3}$, calculate $\hat{\mathbf{Y}}_2$ from $\mathbf{Y}_2$, and then redetermine $\{\hat{\phi}_k, \hat{\mathbf{x}}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=2}$ from $\hat{\mathbf{Y}}_2$.

Iterate the previous three substeps until "practical convergence".

Remaining Steps: Continue until $\bar{K}$ is equal to the desired or estimated number of scatterers.

The "practical convergence" in the iterations of the above relaxation-based optimization algorithm may be determined by checking the relative change ϵ of the cost function $C_5\left(\{\hat{\phi}_k, \hat{\mathbf{x}}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=1}^{\bar{K}}\right)$ in (5.33) between two consecutive iterations. Our numerical and experimental examples show that the algorithm usually converges in a few iterations.

We can determine K, the number of scatterers in (5.2), by extending the generalized Akaike information criterion (see [24] for details). By assuming that the noise is white, the estimate $\hat{K}$ of K is determined as an integer that minimizes the following cost function:

$$\mathrm{GAIC}_{\check{K}} = N\bar{N}\ln\left(\sum_{n=0}^{N-1}\sum_{\bar{n}=0}^{\bar{N}-1}|\hat{e}_{\check{K}}(n,\bar{n})|^2\right) + \gamma\ln[\ln(N\bar{N})][\check{K}(\bar{N}+3)+1], \tag{5.37}$$

where γ is a constant of user choice and is usually determined by empirical experience,

$$\hat{e}_{\check{K}}(n,\bar{n}) = y(n,\bar{n}) - \sum_{k=1}^{\check{K}}\hat{x}_k(n)e^{j\hat{\phi}_k}e^{j2\pi(\hat{f}_k n + \hat{\bar{f}}_k \bar{n})}, \quad n = 0,1,\cdots,N-1, \quad \bar{n} = 0,1,\cdots,\bar{N}-1, \tag{5.38}$$

and $\breve{K}(\bar{N}+3)+1$ is the total number of real-valued unknown parameters (of which $\breve{K}(\bar{N}+3)$ is for the scatterers and 1 is for the white noise variance).

Note that the NLS estimates of $\{\phi_k, \mathbf{x}_k, f_k, \bar{f}_k\}_{k=1}^K$ can also be obtained from $\{\mathbf{Y}_k\}_{k=1}^K$ determined in (5.34), rather than $\{\hat{\mathbf{Y}}_k\}_{k=1}^K$, via the above relaxation-based optimization algorithm. We refer to this approach as the hybrid method. When multiple scatterers are located in the same range, the hybrid method may be computationally more efficient than SPAR since the hybrid method does not isolate the scatterers so that multiple scatterers located in the same range can be more efficiently described as at most two "scatterers". However, the hybrid SAR images may suffer from more severe artifact problem than SPAR, especially at low SNR. When no multiple scatterers are located in the same range, SPAR and the hybrid method perform similarly.

5.5 Modified RELAX-NLS Algorithm

RELAX-NLS [17] is a parametric approach for the feature extraction of targets consisting of both trihedrals and dihedrals. It is based on a mixed data model in which $x(\bar{n})$ is modeled as a real-valued constant for a trihedral or a sinc function of $\bar{n}$ for a dihedral. Like SPAR, RELAX-NLS extracts the target features by minimizing an NLS cost function via a relaxation-based approach. However, RELAX-NLS is computationally expensive since a 4-D search over the parameter space is required for dihedral corner reflectors. Since SPAR is more robust and computationally more efficient than RELAX-NLS, the former can be used to provide the initial conditions needed by the latter.

Let $\{\hat{\phi}_k, \hat{\mathbf{x}}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=1}^{\hat{K}}$ denote the parameter estimates obtained via SPAR according to the data model in (5.2), where $\hat{K}$ is the estimated number of scatterers obtained via the aforementioned generalized Akaike information criterion. The SPAR estimates $\{\hat{\phi}_k, \hat{\mathbf{x}}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=1}^{\hat{K}}$ cannot be used directly as initial conditions for RELAX-NLS. The initial conditions are obtained by applying the first step of RELAX-NLS [17] to each $\mathbf{U}_k, \quad k = 1, 2, \cdots, \hat{K}$, where

$$\mathbf{U}_k = \mathbf{Y} - \sum_{i=1, i\neq k}^{\hat{K}} e^{j\hat{\phi}_i} \mathbf{G}_i(\hat{\mathbf{x}}_i, \hat{f}_i, \hat{\bar{f}}_i), \quad k = 1, 2, \cdots, \hat{K}. \tag{5.39}$$

Once we have the initial conditions, we can use the last step of RELAX-NLS [17] to obtain the dihedral and trihedral parameter estimates, which are then used for SAR image formation [25]. We refer to this approach as the modified RELAX-NLS algorithm.

5.6 Numerical and Experimental Results

We demonstrate and compare the SAR image formation performances of SPAR and the modified RELAX-NLS with both numerical and experimental examples. The algorithms are also compared with the hybrid method and RELAX-NLS. In the following examples, the dimensions of the original SAR phase history data matrix are $N = \bar{N} = 32$ and the generalized Akaike information criterion with $\gamma = 5.5$ is used to determine $\hat{K}$ for the relaxation-based feature extraction algorithms of SPAR and the hybrid method. The threshold T_i used in the isolation process of SPAR is 10% of the peak value. The maximization of $C_4(f, \bar{f})$ in (5.30) is done in two steps. First, initial frequency estimates $\hat{f}$ and $\hat{\bar{f}}$ are obtained via 1-D FFT with zero-padding to a total length of 128 in range and to a total length of 64 in cross-range. Next, these initial estimates are refined by using the FMIN function in MATLAB alternately, i.e., by updating $\hat{f}$ while fixing $\hat{\bar{f}}$ at its most recently determined value and vice versa, until "practical convergence", which is determined by checking the relative change of the cost function $C_4(\hat{f}, \hat{\bar{f}})$. We have used 10^{-3} to determine the convergence of this fine search as well as the relaxation-based algorithm. The extrapolation factor $\zeta = 8$ is used in range for SPAR and the hybrid method and in both range and cross-range for both RELAX-NLS and the modified RELAX-NLS algorithm. Both 1-D and 2-D Kaiser windows with shape parameter $\beta = 6$ are used whenever needed. (We will be happy to provide the MATLAB codes to the interested readers.)

First consider a numerical example with high SNR. The SAR phase history data matrix is simulated by assuming that there are four trihedrals and three dihedrals in the presence of zero-mean white complex Gaussian noise with variance $\sigma_n^2 = 0.6$. The amplitude functions for the four trihedrals are generated as follows:

$$x_k(\bar{n}) = 1, \quad k = 1, 2, 3, \quad \bar{n} = 0, 1, \cdots, \bar{N} - 1, \tag{5.40}$$

and

$$x_4(\bar{n}) = 2, \quad \bar{n} = 0, 1, \cdots, \bar{N} - 1. \tag{5.41}$$

The amplitude functions for the three dihedrals are

$$x_5(\bar{n}) = 9.6\text{sinc}[0.3\pi(\bar{n} - 18.6)], \quad \bar{n} = 0, 1, \cdots, \bar{N} - 1, \tag{5.42}$$

and

$$x_k(\bar{n}) = 6.4\text{sinc}[0.2\pi(\bar{n} - 18.6)], \quad k = 6, 7, \quad \bar{n} = 0, 1, \cdots, \bar{N} - 1, \tag{5.43}$$

where $\text{sinc}(x) = \sin(x)/x$. Figure 5.4(a) shows the modulus of the true SAR image. Note that two of the dihedrals are located in the same cross-range and are closely spaced in range and two of them are located in the same range. Of the four trihedrals, two of them are closely spaced in range and the other two are located in the same range. In this example, the hybrid method and SPAR have the same estimated number of scatterers, $\hat{K} = 7$. Figure 5.4(b) shows the windowed 2-D FFT SAR image obtained by applying the normalized 2-D FFT to the windowed data matrix. SAR images formed via the hybrid method, SPAR, RELAX-NLS, and the modified RELAX-NLS algorithm are shown in Figures 5.4(c) through 5.4(f), respectively. We note that at high SNR, the hybrid image is similar to the SPAR image. Both of the parametric RELAX-NLS and the modified RELAX-NLS algorithms outperform their semi-parametric counterparts SPAR and the hybrid method since the data model used by the parametric methods is correct rather than approximate. For this example, our simulations show that the ratios between the MATLAB flops needed by the hybrid method, SPAR, the modified RELAX-NLS, and RELAX-NLS over the flops needed by the windowed FFT method are 27.4, 28.4, 50.1, and 70.8, respectively. Note that both SPAR and the hybrid method are computationally more efficient than RELAX-NLS and the modified RELAX-NLS algorithm, with the modified RELAX-NLS being more efficient than RELAX-NLS.

Consider next a numerical example with low SNR. The SAR phase history data are the same as in the above example except that the noise variance is increased to $\sigma_n^2 = 6$. SAR images obtained by using the windowed 2-D FFT, the hybrid method, SPAR, RELAX, and the modified RELAX-NLS algorithm are shown in Figures 5.5(a) through 5.5(e), respectively.

Note that the artifact problem starts to show up in the SAR image obtained via the hybrid method in Figure 5.5(b) due to large parameter estimation errors. By comparing Figures 5.5(c) with (b), it can be seen that SPAR can effectively mitigate the artifact problem.

Finally, consider an experimental example of SAR image formation by using the Moving and Stationary Target Acquisition and Recognition (MSTAR) Slicy data collected by imaging an object consisting of both trihedral and dihedral corner reflectors, which is shown in Figure 5.6. The data were collected by the Sandia National Laboratory using the STARLOS sensor. The field data were collected by a spotlight-mode SAR with a carrier frequency of 9.559 GHz and bandwidth of 0.591 GHz. The radar was about 5 kilometers away from the ground object. The data were collected when the object was illuminated by the radar from approximately the azimuth angle 0° and elevation angle 30°. To cross-validate the experimental results given below, XPATCH [26], a high frequency electromagnetic scattering prediction code for complex 3-D objects, was used to generate very high resolution phase history data for the object shown in Figure 5.6. Note that the computer-aided design (CAD) model used in XPATCH may have slightly different dimensions as the object used to collect the Slicy data. The data generated by XPATCH has a resolution of 0.038 meters in both range and cross-range, and the corresponding windowed FFT SAR image is shown in Figure 5.7(a). (We have used the log scale for all of the images shown in Figure 5.7.) The original experimental Slicy data have a resolution of 0.3 meters in range and 0.32 meters in cross-range. The 32×32 data matrix we used to demonstrate the performance of our algorithms has a spoiled resolution of 0.51 meters in range and 0.54 meters in cross-range. The windowed 2-D FFT SAR image of this data matrix is shown in Figure 5.7(b). Figures 5.7(c) and (d), respectively, show the SAR images obtained via the hybrid method and SPAR with $\hat{K} = 7$ (obtained via the generalized Akaike information criterion). Figures 5.7(e) and (f) show the SAR images obtained via RELAX-NLS and the modified RELAX-NLS algorithm with $\hat{K} = 7$, respectively. Note that the hybrid method has a more severe artifact problem than SPAR. Note also that the SPAR image shown in Figure 5.7(d) appears to fit Figure 5.7(a) and the characteristics of the object in Figure 5.6 well. However, the parametric algorithms are not as robust as SPAR since the parametric images shown in Figures 5.7(e) and (f) do not fit Figure 5.7(a) as well

with one of the scatterers mis-identified and mis-located. For this experimental example, the ratios between the MATLAB flops needed by the hybrid method, SPAR, RELAX-NLS, and the modified RELAX-NLS over the flops needed by the windowed FFT method are 29.2, 16.7, 32.8, and 43.1, respectively. Note that SPAR can sometimes be faster than the hybrid method!

5.7 Conclusions

We have presented a semi-parametric spectral estimation algorithm, referred to as SPAR, for SAR target feature extraction and image formation based on a flexible data model. SPAR can be used to effectively mitigate the artifact problem encountered by the hybrid algorithm for SAR image formation due to the flexibility of the data model. SPAR can also be used to provide initial conditions needed by other parametric algorithms to reduce the total amount of computations needed to form SAR images. Due to the flexible data model used by SPAR, this semi-parametric algorithm is more robust and computationally more efficient than the existing parametric algorithms.

Acknowledgement

The authors would like to thank Mr. M. Koets for providing us with the XPATCH data.

Appendix A: Feature Extraction of a Single Scatterer via SPAR

The estimates of $\left\{\phi, \mathbf{x}, f, \bar{f}\right\}$ can be obtained by minimizing the following NLS cost function:

$$\begin{aligned} C_1\left(\phi, \mathbf{x}, f, \bar{f}\right) &= \left\|\mathbf{Y}_s - e^{j\phi}\mathbf{G}(\mathbf{x}, f, \bar{f})\right\|_F^2 \\ &= \sum_{\bar{n}=0}^{\bar{N}-1} \left\|\mathbf{y}_{s_{\bar{n}}} - x(\bar{n})e^{j(2\pi\bar{f}\bar{n}+\phi)}\boldsymbol{\omega}_N(f)\right\|^2 . \end{aligned} \tag{5.44}$$

After simple calculations, we can rewrite (5.44) as

$$\begin{aligned} C_2\left(\phi, \mathbf{x}, f, \bar{f}\right) &= \sum_{\bar{n}=0}^{\bar{N}-1} \left\{ \|\mathbf{y}_{s_{\bar{n}}}\|^2 + N\left[x(\bar{n}) - \frac{1}{N}\mathrm{Re}\left(e^{-j(2\pi\bar{f}\bar{n}+\phi)}\mathbf{y}_{s_{\bar{n}}}^T \boldsymbol{\omega}_N^*(f)\right)\right]^2 \right. \\ &\quad \left. -\frac{1}{N}\mathrm{Re}^2\left[e^{-j(2\pi\bar{f}\bar{n}+\phi)}\mathbf{y}_{s_{\bar{n}}}^T \boldsymbol{\omega}_N^*(f)\right]\right\}. \end{aligned} \tag{5.45}$$

Minimizing (5.45) with respect to $x(\bar{n})$ yields

$$\hat{x}(\bar{n}) = \frac{1}{N}\mathrm{Re}\left[e^{-j(2\pi\bar{f}\bar{n}+\phi)}\mathbf{y}_{s_{\bar{n}}}^T \boldsymbol{\omega}_N^*(f)\right], \quad \bar{n} = 0, 1, \cdots, \bar{N}-1. \tag{5.46}$$

Hence

$$\begin{aligned} \hat{\mathbf{x}} &= \frac{1}{N}\mathrm{Re}\left\{e^{-j\phi}\left[\mathbf{Y}_s^T \boldsymbol{\omega}_N^*(f)\right] \odot \boldsymbol{\omega}_{\bar{N}}^*(\bar{f})\right\} \\ &= \frac{1}{N}\mathrm{Re}\left[e^{-j\phi}\bar{\mathbf{y}}_s(f) \odot \boldsymbol{\omega}_{\bar{N}}^*(\bar{f})\right]. \end{aligned} \tag{5.47}$$

Inserting (5.46) into (5.45), we obtain the NLS estimates $\{\hat{\phi}, \hat{f}, \hat{\bar{f}}\}$ of $\{\phi, f, \bar{f}\}$ by equivalently maximizing the following cost function:

$$\begin{aligned} C_3(\phi, f, \bar{f}) &= \sum_{\bar{n}=0}^{\bar{N}-1} \left\{\mathrm{Re}\left[\mathbf{y}_{s_{\bar{n}}}^T \boldsymbol{\omega}_N^*(f) e^{-j(2\pi\bar{f}\bar{n}+\phi)}\right]\right\}^2 \\ &= \frac{1}{2}\sum_{\bar{n}=0}^{\bar{N}-1} \left\{|\mathbf{y}_{s_{\bar{n}}}^T \boldsymbol{\omega}_N^*(f)|^2 + \mathrm{Re}\left[\left(\mathbf{y}_{s_{\bar{n}}}^T \boldsymbol{\omega}_N^*(f)\right)^2 e^{-j[2\pi(2\bar{f})\bar{n}+2\phi]}\right]\right\}. \end{aligned} \tag{5.48}$$

Let $\bar{y}_s(\bar{n}, f)$ denote the $\bar{n}$th element of $\bar{\mathbf{y}}_s(f), \bar{n} = 0, 1, \cdots, \bar{N}-1$. Then $\hat{\phi}$ is given by

$$\hat{\phi} = \frac{1}{2}\arg\left\{\sum_{\bar{n}=0}^{\bar{N}-1}\left[\mathbf{y}_{s_{\bar{n}}}^T \boldsymbol{\omega}_N^*(f)\right]^2 e^{-j2\pi(2\bar{f})\bar{n}}\right\}. \tag{5.49}$$

Inserting (5.49) into (5.48) and ignoring the scaling factor, we can simplify (5.48) to:

$$\begin{aligned} C_4(f, \bar{f}) &= \sum_{\bar{n}=0}^{\bar{N}-1}\left|\mathbf{y}_{s_{\bar{n}}}^T \boldsymbol{\omega}_N^*(f)\right|^2 + \left|\sum_{\bar{n}=0}^{\bar{N}-1}\left[\mathbf{y}_{s_{\bar{n}}}^T \boldsymbol{\omega}_N^*(f)\right]^2 e^{-j2\pi(2\bar{f})\bar{n}}\right| \\ &= \|\bar{\mathbf{y}}_s(f)\|^2 + \left|\sum_{\bar{n}=0}^{\bar{N}-1}\bar{y}_s^2(\bar{n}, f)e^{-j2\pi(2\bar{f})\bar{n}}\right|. \end{aligned} \tag{5.50}$$

Then the NLS estimates $\{\hat{f}, \hat{\bar{f}}\}$ of $\{f, \bar{f}\}$ are determined by

$$\{\hat{f}, \hat{\bar{f}}\} = \arg\max_{f, \bar{f}} C_4(f, \bar{f}). \tag{5.51}$$

Note that $\bar{\mathbf{y}}_s(f)$ in (5.26) can be obtained by applying 1-D FFT to each column of $\mathbf{Y}_s$ and the second term in (5.50) can also be readily obtained by applying 1-D FFT to the sequence $\{\bar{y}_s^2(\bar{n}, f)\}_{\bar{n}=0}^{\bar{N}-1}$ with $2\bar{f}$ as the frequency variable. Hence $\{\hat{f}, \hat{\bar{f}}\}$ can be obtained via a 2-D search for the location corresponding to the peak of $C_4(f, \bar{f})$, which can be computed efficiently via 1-D FFTs. Note also that padding with zeros for the 1-D FFTs is necessary to achieve high accuracy for the frequency estimates. An alternative approach is to find an approximate location corresponding to the global maximum with 1-D FFT without much zero-padding and then use the approximate location as the initial condition to find a more accurate position via, for example, alternately using the FMIN function in MATLAB.

Reference

[1] M. T. Fennell and R. P. Wishner, "Battlefield awareness via synergistic SAR and MTI exploitation," *IEEE Aerospace and Electronic Systems Magazine*, vol. 13, pp. 39–45, February 1998.

[2] S. R. DeGraaf, "SAR imaging via modern 2-D spectral estimation methods," *SPIE Proceedings on Optical Engineering in Aerospace Sensing*, vol. 2230, pp. 36–47, Orlando, FL, April 1994.

[3] S. R. DeGraaf, "SAR imaging via modern 2-D spectral estimation methods," *IEEE Transactions on Image Processing*, vol. 7, pp. 729–761, May 1998.

[4] G. R. Benitz, "Adaptive high-definition imaging," *SPIE Proceedings on Optical Engineering in Aerospace Sensing,* Orlando, FL, pp. 106–119, April 1994.

[5] G. R. Benitz, "High definition vector imaging for synthetic aperture radar," *Proceedings of the 31st Asilomar Conference on Signals, Systems and Computers,* Pacific Grove, CA, November 1997.

[6] G. R. Benitz, "High-definition vector imaging," *MIT Lincoln Laboratory Journal - Special Issue on Superresolution*, vol. 10, no. 2, pp. 147–170, 1997.

[7] S. R. DeGraaf, "Sidelobe reduction via adaptive FIR filtering in SAR imagery," *IEEE Transactions on Image Processing*, vol. 3, pp. 292–301, May 1994.

[8] H. Li, J. Li, and P. Stoica, "Performance analysis of forward-backward matched-filterbank spectral estimators," *IEEE Transactions on Signal Processing*, vol. 46, pp. 1954–1966, July 1998.

[9] J. Capon, "High resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, pp. 1408–1418, August 1969.

[10] J. Li and P. Stoica, "An adaptive filtering approach to spectral estimation and SAR imaging," *IEEE Transactions on Signal Processing*, vol. 44, pp. 1469–1484, June 1996.

[11] A. Farina, F. Prodi, and F. Vinelli, "Application of superresolution techinques to radar imaging," *Chinese Journal of Systems Engineering and Electronics*, vol. 5, no. 1, pp. 1–14, January 1994.

[12] A. Farina, A. Forte, F. Prodi, and F. Vinelli, "Superresolution capabilities in 2D direct and inverse SAR processing," *International Symposium on Noise and Clutter Rejection in Radars and Imaging Sensors*, pp. 151–156, Kanagawa Science Park, Japan, November 1994.

[13] S. Barbarossa, L. Marsili, and G. Mungari, "SAR super-resolution imaging by signal subspace projection techniques," *AEU International Journal of Electronics and Communications*, vol. 50, no. 2, pp. 133–138, March 1996.

[14] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. AP-34, pp. 276–280, March 1986.

[15] R. Roy and T. Kailath, "ESPRIT – Estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-37, pp. 984–995, July 1989.

[16] J. Li and P. Stoica, "Efficient mixed-spectrum estimation with applications to target feature extraction," *IEEE Transactions on Signal Processing*, vol. 44, pp. 281–295, February 1996.

[17] Z.-S. Liu and J. Li, "Feature extraction of SAR targets consisting of trihedral and dihedral corner reflectors," *IEE Proceedings on Radar, Sonar and Navigation*, vol. 145, pp. 161–172, March 1998.

[18] V. Larson and L. Novak, "Polarimetric subspace target detector for SAR data based on the Huynen dihedral model," *Proceedings of SPIE*, vol. 2487, pp. 235–250, Orlando, Florida, April 1995.

[19] L. M. Novak, G. R. Benitz, G. J. Owirka, and L. A. Bessette, "ATR performance using enhanced resolution SAR," *SPIE Proceedings on Optical Engineering in Aerospace Sensing,* Orlando, FL, pp. 332–337, April 1996.

[20] L. M. Novak, G. R. Benitz, and G. J. Owirka, "Classifier performance using enhanced resolution SAR data," *Proceedings of the IEE International Conference on Radar*, pp. 634–638, Edinburgh, UK, October 1997.

[21] L. M. Novak, G. J. Owirka, B. William S, and A. L. Weaver, "The automatic target-recognition system in SAIP," *MIT Lincoln Laboratory Journal - Special Issue on Superresolution*, vol. 10, no. 2, pp. 187–202, 1997.

[22] Z.-S. Liu, H. Li, and J. Li, "Efficient implementation of Capon and APES for spectral estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, pp. 1314–1319, October 1998.

[23] G. H. Golub and C. F. Van Loan, *Matrix Computations.* Baltimore, MD: Johns Hopkins University Press, 1984.

[24] T. Söderström and P. Stoica, *System Identification.* London, U.K.: Prentice-Hall International, 1989.

[25] Z. Bi, J. Li, and Z.-S. Liu, "Super resolution SAR image formation via parametric spectral estimation methods," to appear in *IEEE Transactions on Aerospace and Electronic Systems*, January 1999.

[26] D. J. Andersh, M. Hazlett, S. W. Lee, D. D. Reeves, D. P. Sullivan, and Y. Chu, "XPATCH: a high-frequency electromagnetic scattering prediction code and environment for complex three-dimensional objects," *IEEE Antennas and Propagation Magazine*, vol. 36, pp. 65–69, February 1994.

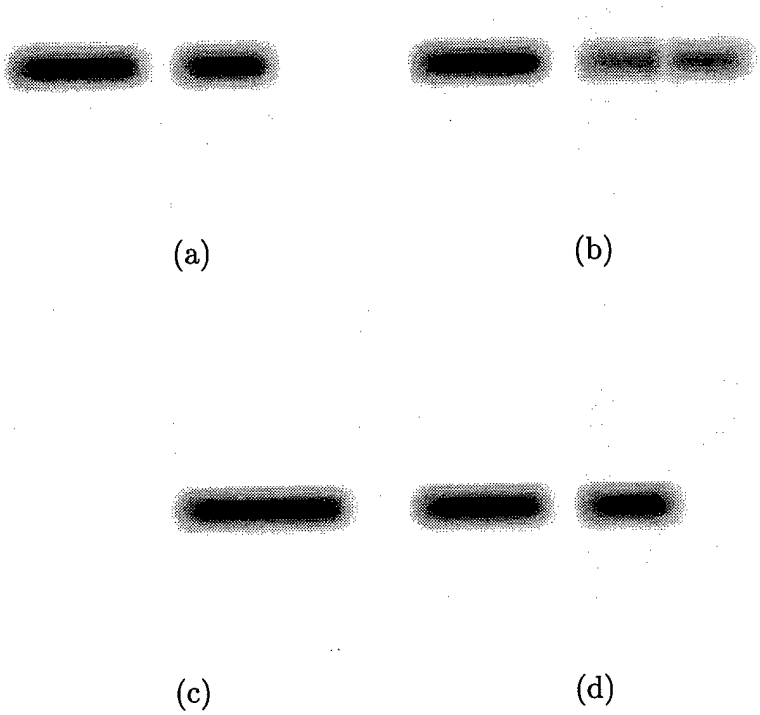


Figure 5.1: Ambiguity effect on the SAR image formation in the absence of range estimation errors. (a) True windowed FFT SAR image. (b) Windowed FFT image of the first "scatterer". (c) Windowed FFT image of the second "scatterer". (d) Combined windowed FFT image of the two "scatterers". (The vertical and horizontal axes are for range and cross-range, respectively.)

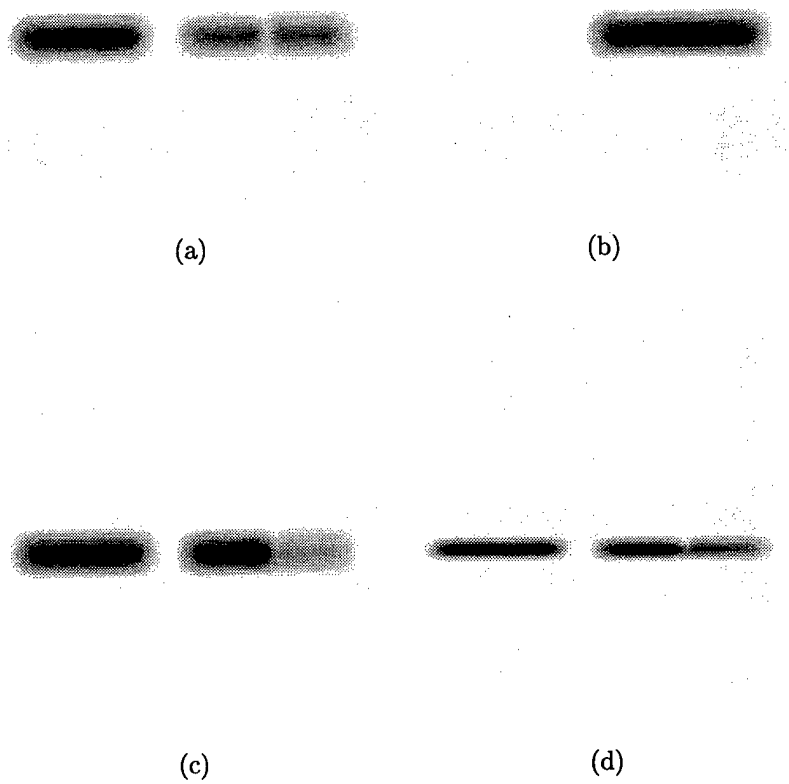


Figure 5.2: Ambiguity effect on the SAR image formation in the presence of range estimation errors. (a) Windowed FFT image of the first "scatterer". (b) Windowed FFT image of the second "scatterer". (c) Combined windowed FFT image of the two "scatterers" with $\zeta = 1$ (without extrapolation). (d) Combined windowed FFT image with $\zeta = 2$. (The vertical and horizontal axes are for range and cross-range, respectively.)

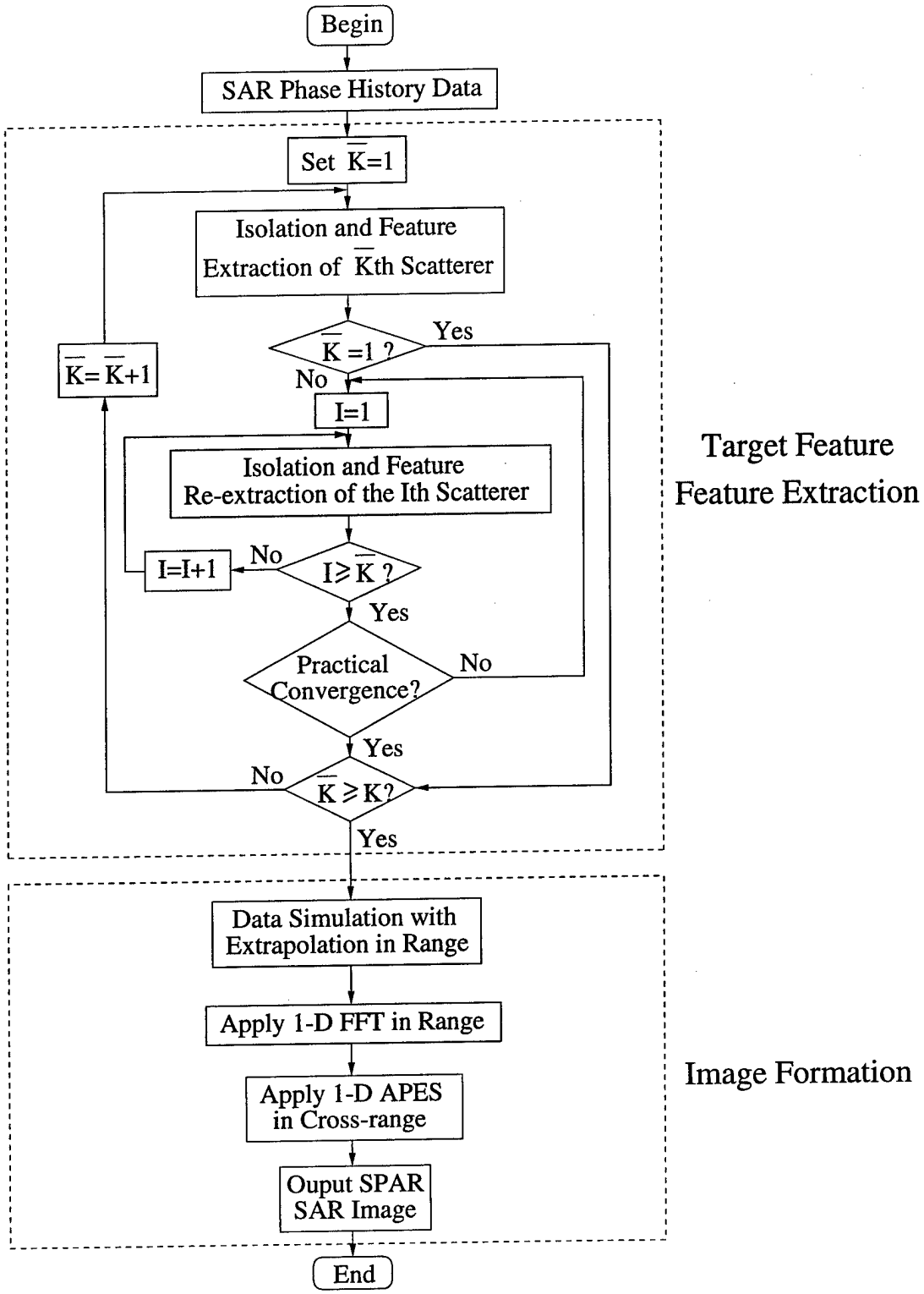


Figure 5.3: Block diagram for SPAR.

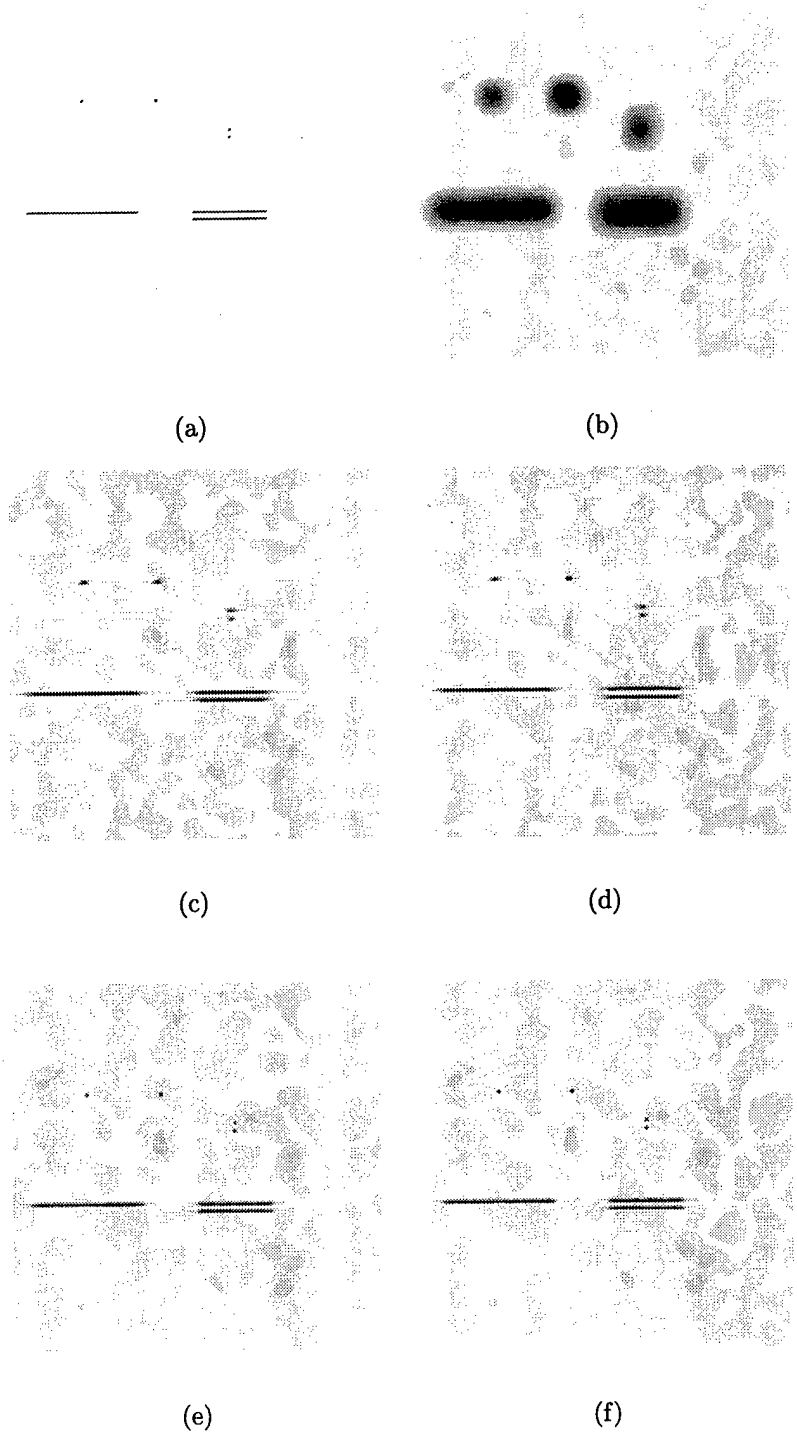


Figure 5.4: Comparison of SAR images formed using different algorithms for simulated data at high SNR ($\sigma_n^2 = 0.6$). (a) True SAR image. (b) Windowed 2-D FFT SAR image. (c) The hybrid SAR image. (d) SPAR SAR image. (e) RELAX-NLS SAR image. (f) Modified RELAX-NLS SAR image. (The vertical and horizontal axes are for range and cross-range, respectively.)

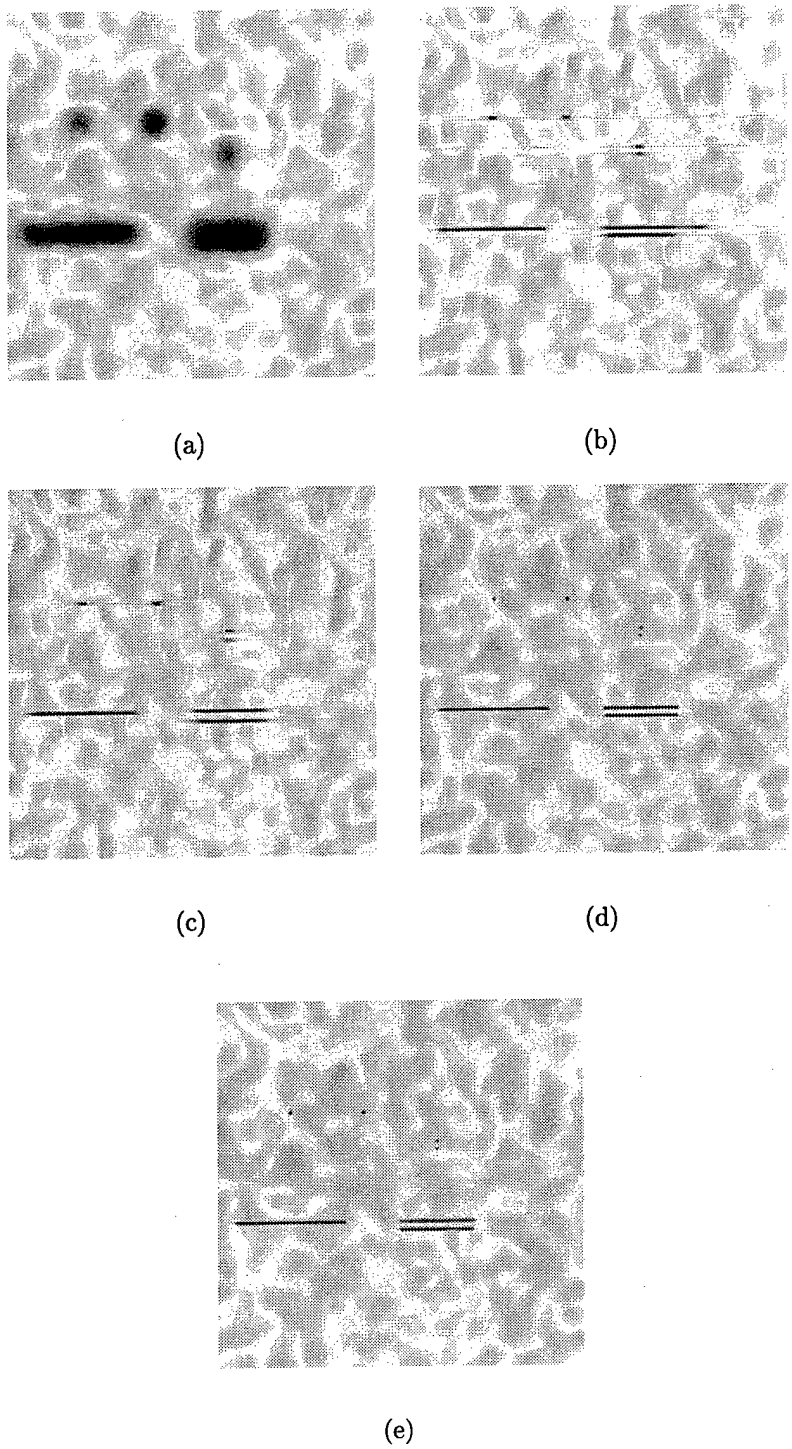


Figure 5.5: Comparison of SAR images formed using different algorithms for simulated data at low SNR ($\sigma_n^2 = 6$). (a) Windowed 2-D FFT SAR image. (b) The hybrid SAR image. (c) SPAR SAR image. (d) RELAX-NLS SAR image. (e) Modified RELAX-NLS SAR image. (The vertical and horizontal axes are for range and cross-range, respectively.)

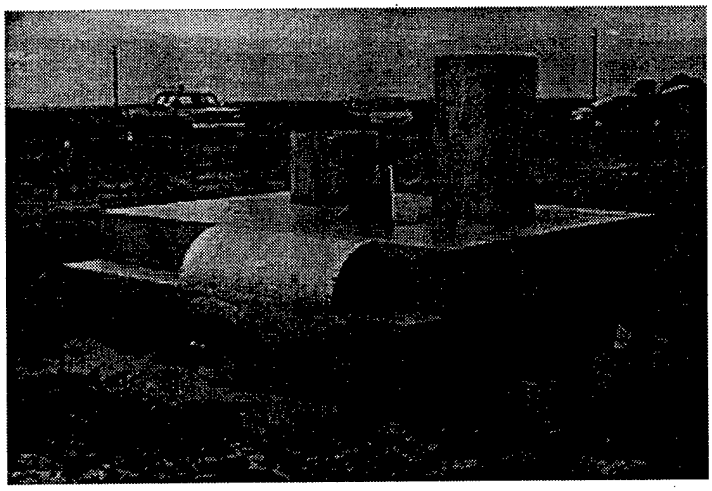


Figure 5.6: Target photo taken at 45° azimuth angle.

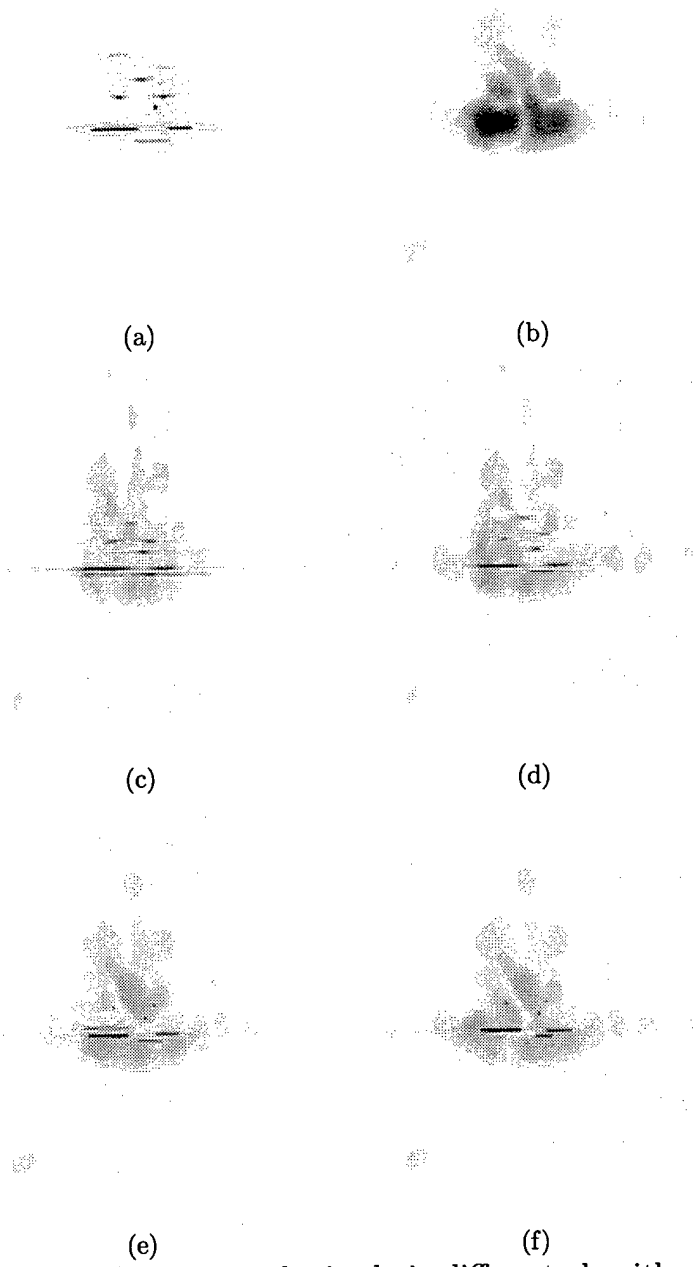


Figure 5.7: Comparison of SAR images obtained via different algorithms for the Slicy data hb15533.015 (0° azimuth and 30° elevation angles). (a) Windowed 2-D FFT SAR image from the data generated by XPATCH with resolution 0.038×0.038 meters. (b) Windowed 2-D FFT SAR image from the Slicy data with resolution 0.51×0.54 meters. (c) The hybrid SAR image obtained from the data used in (b). (d) SPAR SAR image obtained from the data used in (b). (e) RELAX-NLS SAR image obtained from the data used in (b). (f) Modified RELAX-NLS SAR image obtained from the data used in (b). (The vertical and horizontal axes are for range and cross-range, respectively.)

6. Synthetic Aperture Radar Motion Compensation and Feature Extraction via a Relaxation Based Algorithm

6.1 Introduction

Synthetic aperture radar (SAR) can be used to form radar target images and extract target features with high resolution in both range and cross-range directions. For an airborne SAR system, phase errors occur along the synthetic aperture (cross-range direction) due to uncompensated platform motion. These errors can significantly degrade the SAR image quality. Many algorithms have been proposed to compensate these unknown phase errors [1]. Most existing motion compensation algorithms are non-parametric and hence are robust. Among these algorithms, the phase-gradient autofocus (PGA) algorithm [1, 2, 3, 4, 5] is one of the most well-known and competitive algorithms.

In many applications including automatic target classification, however, it is convenient to describe a small radar target via several parameters, which are the features of the target. High resolution parametric algorithms, such as the RELAX algorithm, have been derived to extract the target features [6]. Yet these algorithms assume that the motion errors do not exist. It appears that extracting target features via parametric methods in the presence of motion errors has not been addressed before.

In this chapter, we establish a data model for the feature extraction of point scatterers in the presence of uncompensated aperture motion errors and unknown noise. We also propose a parametric relaxation-based algorithm to estimate the target features as well as the motion errors based on the data model. This algorithm is referred to as the *motion compensation RELAX* algorithm or *MCRELAX* in this chapter. MCRELAX minimizes a complicated nonlinear least-squares cost function and is performed by an alternating procedure, which iteratively updates the estimates of the target features by fixing the estimates of the phase errors and then updates the estimates of the phase errors by fixing the estimates of the target features. The initial estimates of the phase errors in the MCRELAX algorithm can be obtained by using the PGA algorithm to speed up the convergence rate of the former. MCRELAX can also used for motion compensation only and is still computationally simple

since it requires a sequence FFTs (fast Fourier transforms) and vector products and hence can be implemented in hardware easily.

The remainder of the chapter is organized as follows. In Section 6.2, we describe the data model and formulate the problem of interest. Section 6.3 presents the MCRELAX algorithm. In Section 6.4, we derive the Cramér-Rao bound (CRB) for the target features and phase errors. Section 6.5 shows the results of several numerical and experimental examples. Section 6.6 contains our conclusions. Finally, to make this chapter self-contained, the computational steps of the PGA algorithm are summarized in the appendix.

6.2 Data Model and Problem Formulation

We first describe how one can obtain 1-D target features via a high range resolution radar. The range resolution of a radar is determined by the radar bandwidth. To achieve high resolution in range, the radar must transmit wideband pulses, which are often linear frequency modulated (FM) chirp pulses. A normalized chirp pulse can be written as

$$s(t) = \cos\left[(2\pi f_0 t + \gamma t^2)\right], \quad |t| \leq T_0/2, \tag{6.1}$$

where f_0 denotes the carrier frequency, 2γ denotes the FM rate, and T_0 denotes the width of the pulse. We assume that f_0, γ, and T_0 are known. The signal returned by a scatterer of a target has the form (after mixed with the cosine and sine terms)

$$r(t) = \delta_\tau \exp\left\{-j[2\pi f_0(t-\tau) + \gamma(t-\tau)^2]\right\}, \tag{6.2}$$

where δ_τ is determined by the radar cross section (RCS) of the scatterer and τ denotes the round-trip time delay. The demodulated signal $\tilde{d}(t)$ is obtained by mixing $r(t)$ with $\exp\{-j\,[2\pi f_0(t-\tau_0) + \gamma(t-\tau_0)^2]\}$ for some given τ_0 (to be defined later on), where $(\cdot)^*$ denotes the complex conjugate,

$$\tilde{d}(t) = \delta_\tau \exp\left[j(2\pi f_0 - 2\gamma\tau_0)(\tau-\tau_0)\right] \exp\left[-j\gamma(\tau-\tau_0)^2\right] \exp\left[j2\gamma(\tau-\tau_0)t\right]. \tag{6.3}$$

The term $\exp\left[-j\gamma(\tau-\tau_0)^2\right]$ in (6.3) is usual close to a constant and can also be partially removed [7] as follows. Let $\tilde{D}(\omega)$ denote the Fourier transform of $\tilde{d}(t)$. Then the inverse

Fourier transform of $\tilde{D}(\omega)\exp\left(j\frac{\omega^2}{4\gamma}\right)$ will have the term $\exp\left[-j\gamma(\tau-\tau_0)^2\right]$ removed. Yet this removal can only be approximate since $\tilde{d}(t)$ is not known for all t and hence $\tilde{D}(\omega)$ is not known exactly. The closer $\exp\left[-j\gamma(\tau-\tau_0)^2\right]$ is to a constant for $\tau_{\min} \leq \tau \leq \tau_{\max}$, where $\tau_{\max}$ and $\tau_{\min}$ correspond to the maximum and minimum values, respectively, of the round-trip time delays between the scatterers of a target and the radar, the better its removal. With this removal, we have

$$\bar{d}(t) = \delta_\tau \exp\left[j(2\pi f_0 - 2\gamma\tau_0)(\tau-\tau_0)\right]\exp\left[j2\gamma(\tau-\tau_0)t\right], \tag{6.4}$$

which is a sinusoidal signal with frequency $2\gamma(\tau-\tau_0)$ and complex amplitude $\delta_\tau \exp\left[j(2\pi f_0 - 2\gamma\tau_0)(\tau-\tau_0)\right]$. We know $\tau_{\max}$ and $\tau_{\min}$ approximately since we assume that the altitude, antenna beamwidth, and grazing angle of the radar are known. We also assume that $(\tau_{\max}-\tau_{\min}) \ll T_0$. Then for $-T_0/2+\tau_{\max} \leq t \leq T_0/2+\tau_{\min}$, the scatterers of the target at different ranges correspond to different frequencies of the signal $\bar{d}(t)$, while the RCS's of the scatterers are proportional to the amplitudes of the corresponding sinusoids. The ranges and RCS's of the target scatterers are the one-dimensional (1-D) target features.

We now describe how one can obtain two-dimensional (2-D) target features via a spotlight-mode SAR. The cross-range resolution of an ordinary ranging radar is limited by its antenna beamwidth. For an airborne or spaceborne system, a narrow antenna beamwidth requires an antenna that may be too large to be carried on board of the airplane or the spacecraft. Spotlight-mode SAR avoids this requirement by collecting coherent radar returns while viewing a target from many different angles [7]. By properly processing the return signals, we can also achieve high resolution in cross-range.

A broadside data collection geometry in a spotlight-mode SAR is shown in Figure 6.1. The XYZ coordinate system is centered on a small patch of ground, where a target is located. The angle θ and ϕ denote the azimuth and elevation angles, respectively, of the radar relative to the XYZ coordinate system. The distance between the radar and the coordinate origin of the XYZ coordinate system is R_0. The ground is illuminated by a narrow radio frequency (RF) beam from the moving radar that moves along the θ direction but with ϕ and R_0 fixed. In Figure 6.1, R denotes the distance between the radar and a

scatterer at the position (x, y, z). We assume that θ, ϕ, and R_0 are known.

The range R of the scatterer located at (x, y, z) can be written as

$$R = \left[(R_0 \cos\theta \cos\phi - x)^2 + (R_0 \sin\theta \cos\phi - y)^2 + (R_0 \sin\phi - z)^2\right]^{1/2}. \tag{6.5}$$

Under the conditions $\frac{x}{R_0} \ll 1$, $\frac{y}{R_0} \ll 1$, and $\frac{z}{R_0} \ll 1$, we have

$$\begin{aligned} R &= R_0 \left[1 - 2\tfrac{x}{R_0}\cos\theta\cos\phi - 2\tfrac{y}{R_0}\sin\theta\cos\phi - 2\tfrac{z}{R_0}\sin\phi + \tfrac{x^2+y^2+z^2}{R_0^2}\right]^{1/2} \\ &\approx R_0 \left[1 - \tfrac{x}{R_0}\cos\theta\cos\phi - \tfrac{y}{R_0}\sin\theta\cos\phi - \tfrac{z}{R_0}\sin\phi + \tfrac{x^2+y^2+z^2-(x\cos\theta\cos\phi+y\sin\theta\cos\phi+z\sin\phi)^2}{2R_0^2}\right] \\ &= R_0 - x\cos\theta\cos\phi - y\sin\theta\sin\phi - z\sin\phi + \tfrac{x^2+y^2+z^2-(x\cos\theta\cos\phi+y\sin\theta\cos\phi+z\sin\phi)^2}{2R_0}. \end{aligned} \tag{6.6}$$

For the broadside data collection geometry, θ is very small [31, 7] . For very small θ, we have $\sin 2\theta \approx 2\sin\theta$ and $\cos\theta \approx 1$. Then

$$\begin{aligned} & x^2 + y^2 + z^2 - (x\cos\theta\cos\phi + y\sin\theta\cos\phi + z\sin\phi)^2 \\ \approx\ & x^2\sin^2\phi + y^2 + z^2\cos^2\phi - xy\sin 2\theta\cos^2\phi - xz\cos\theta\sin 2\phi - yz\sin\theta\sin 2\phi \\ \approx\ & \left(\tfrac{x^2\sin^2\phi+y^2+z^2\cos^2\phi-xz\sin 2\phi}{\cos\phi}\right)\cos\theta\cos\phi - 2\left(xy\cos\phi + yz\sin\phi\right)\sin\theta\cos\phi, \end{aligned} \tag{6.7}$$

and

$$z\sin\phi \approx (z\tan\phi)\cos\theta\cos\phi. \tag{6.8}$$

Then

$$R \approx R_0 - \tilde{x}\cos\theta\cos\phi - \tilde{y}\sin\theta\cos\phi, \tag{6.9}$$

where

$$\tilde{x} = x + z\tan\phi - \frac{x^2\sin^2\phi + y^2 + z^2\cos^2\phi - xz\sin 2\phi}{2R_0\cos\phi}, \tag{6.10}$$

and

$$\tilde{y} = y + \frac{xy\cos\phi + yz\sin\phi}{R_0}. \tag{6.11}$$

Note that the second term of the right side of (6.10) is due to the range layover of the scatterer with non-zero height z [5]. The third term of the right side of (6.10) and the second term of the right side of (6.11) are due to the range curvature effect [5].

Let $\tau_0 = \frac{2R_0}{c}$. Since $\tau = \frac{2R}{c}$, then from (6.4), we have

$$d(t, \theta) = \delta_{x,y,z} \exp\left[j(\tilde{x}t_x + \tilde{y}t_y)\right], \tag{6.12}$$

where $\delta_{x,y,z}$ is proportional to the RCS of the scatterer located at (x, y, z),

$$t_x = -\frac{4[\pi f_0 + \gamma(t - \tau_0)]\cos\phi}{c}\cos\theta, \tag{6.13}$$

and

$$t_y = -\frac{4[\pi f_0 + \gamma(t - \tau_0)]\cos\phi}{c}\sin\theta. \tag{6.14}$$

Note that $d(t, \theta)$ in (6.12) is a 2-D sinusoidal signal. The frequency pair of the 2-D sinusoid corresponds to the 2-D location $\{\tilde{x}, \tilde{y}\}$ of the scatterer, while the amplitude is proportional to its RCS. Note that $(\tilde{x}, \tilde{y})$ is the location due to the layover and range curvature effects [5] and is not the true location (x, y, z) of the scatterer. The range curvature effect can be neglected for large R_0. However, this SAR system cannot distinguish the scatterers located at (x, y, z) and $(\tilde{x}, \tilde{y}, 0)$ and combines these scatterers into one scatterer. The scatterers we referred to below are those resulted from such combinations. When a target has multiple scatterers with distinct $(\tilde{x}, \tilde{y})$, $d(t, \theta)$ in (6.12) will be a sum of sinusoids. The 2-D locations and RCS's of the target scatterers are the 2-D target features. Since usually the samples on the t and θ axes are uniformly spaced, the samples of t_x and t_y occur at the points of a polar grid. Hence Polar-to-Cartesian interpolation is needed for the samples of t_x and t_y to occur at rectangular grid points [7].

Thus after Polar-to-Cartesian interpolation and sampling, the signal reflected by a radar target that consists of K 2-D point scatterers in an ideal SAR system can be described as:

$$\tilde{s}(m, \bar{m}) = \sum_{k=1}^{K} \tilde{\alpha}_k \exp\left[j2\pi(mf_k + \bar{m}\tilde{f}_k)\right], \quad m = 0, 1, \cdots, M-1, \quad \bar{m} = 0, 1, \cdots, \bar{M}-1, \tag{6.15}$$

where the complex amplitude $\tilde{\alpha}_k$ and the 2-D frequency pair $\{f_k, \tilde{f}_k\} \in [-0.5 \quad 0.5]$ Hz, respectively, are proportional to the radar cross section (RCS) and the 2-D location (range and cross-range) of the kth scatterer of the target, and M and $\bar{M}$ denote the numbers of available data samples.

In the derivations of the above data model, we have assumed that the radar moves along the θ direction with R_0 known exactly. For a practical airborne SAR system, however, the distance between the moving radar and the coordinate origin of the XYZ coordinate system

may not be known exactly due to platform position uncertainty [5]. Also, the round-trip time delay between the radar and the target may be somewhat random due to atmospheric turbulence [5] and the randomness has the same effect on SAR imaging as when R_0 is in error. For a given θ, let $\Delta R_0(\theta)$ denote the distance error between the radar and the coordinate origin. Then the time delay error has the form $\Delta\tau(\theta) = \frac{2\Delta R_0(\theta)}{c}$. As we can see from (6.4), this $\Delta\tau(\theta)$ mainly causes a phase error $\exp\left[j2\pi f_0 \Delta\tau(\theta)\right]$ because $\exp\left[j2\gamma\Delta\tau(\theta)(t-\tau_0)\right] \approx 1$ for $|t-\tau_0| < T_0$, which is due to $\gamma T_0 \ll \pi f_0$. Since θ is usually very small [31, 7] , we neglect the phase errors caused by the errors in θ and ϕ. Note that the phase errors are independent of t. Hence, by neglecting the effect of Polar-to-Cartesian interpolation, the signal obtained by a realistic SAR system can be described as

$$y(m,\bar{m}) = \tilde{s}(m,\bar{m}) \exp\left(j\tilde{\psi}_{\bar{m}}\right) + e(m,\bar{m}), \tag{6.16}$$

where $\{\tilde{\psi}_{\bar{m}}\}_{\bar{m}=0}^{\bar{M}-1}$ are the phase errors due to the uncompensated aperture motion and $e(m,\bar{m}), \quad m = 0,1,\cdots,M-1, \quad \bar{m} = 0,1,\cdots,\bar{M}-1$, denote the unknown noise.

We note that for any real scalars β_1 and β_2, Equation (6.16) still holds when $\{\tilde{\alpha}_k, f_k, \tilde{f}_k\}_{k=1}^{K}$ and $\{\psi_{\bar{m}}\}_{\bar{m}=0}^{\bar{M}-1}$, respectively, are replaced by $\{\tilde{\alpha}_k \exp(j\beta_1), f_k, \tilde{f}_k + \beta_2\}_{k=1}^{K}$ and $\{\psi_{\bar{m}} - \beta_1 - 2\pi\beta_2\bar{m}\}_{\bar{m}=0}^{\bar{M}-1}$. Hence the data model in (6.16) has ambiguities. That is, we cannot uniquely determine $\left\{\tilde{f}_k\right\}_{k=1}^{K}$ and the phases of $\{\tilde{\alpha}_k\}_{k=1}^{K}$. To avoid these ambiguities, let

$$\alpha_k = \tilde{\alpha}_k \exp\left(j\tilde{\psi}_0\right), \quad k = 1,2,\cdots,K, \tag{6.17}$$

$$\psi_{\bar{m}} = \tilde{\psi}_{\bar{m}} - \tilde{\psi}_0 - (\tilde{\psi}_1 - \tilde{\psi}_0)\bar{m}, \quad \bar{m} = 0,1,\cdots,\bar{M}-1, \tag{6.18}$$

and

$$\bar{f}_k = \tilde{f}_k + \left(\frac{\tilde{\psi}_1 - \tilde{\psi}_0}{2\pi}\right). \tag{6.19}$$

Note that $\psi_0 = \psi_1 = 0$. Hence we can rewrite (6.16) as

$$y(m,\bar{m}) = s(m,\bar{m}) \exp\left(j\psi_{\bar{m}}\right) + e(m,\bar{m}), \tag{6.20}$$

where

$$s(m,\bar{m}) = \sum_{k=1}^{K} \alpha_k \exp\left[j2\pi(mf_k + \bar{m}\bar{f}_k)\right]. \tag{6.21}$$

Let $\mathbf{Y}$ and $\mathbf{E}$ denote the matrices whose $m\bar{m}$th elements are $y(m,\bar{m})$ and $e(m,\bar{m})$, respectively. Let

$$\mathbf{P} = \text{diag}\left\{ 1, \ 1, \quad exp\,(j\psi_2)\,, \ \cdots, \ \exp\,(j\psi_{\bar{M}-1}) \right\}, \tag{6.22}$$

and

$$\boldsymbol{\omega}_M(f_k) = \left[1 \quad \exp\,(j2\pi f_k) \quad \cdots \quad \exp\,[j2\pi(M-1)f_k] \right]^T, \tag{6.23}$$

where $(\cdot)^T$ denotes the transpose. Then (6.20) can be written in the following matrix form:

$$\mathbf{Y} = \sum_{k=1}^{K} \alpha_k \boldsymbol{\omega}_M(f_k)\boldsymbol{\omega}_{\bar{M}}^T(\bar{f}_k)\mathbf{P} + \mathbf{E}. \tag{6.24}$$

Our problem of interest herein is to estimate $\{\alpha_k, f_k, \bar{f}_k\}_{k=1}^K$ and $\{\psi_{\bar{m}}\}_{\bar{m}=2}^{\bar{M}-1}$ from $\mathbf{Y}$ in (6.24).

6.3 The MCRELAX Algorithm

The MCRELAX algorithm obtains the estimates $\left\{\hat{\alpha}_k, \hat{f}_k, \hat{\bar{f}}_k\right\}_{k=1}^K$ and $\left\{\hat{\psi}_{\bar{m}}\right\}_{\bar{m}=2}^{\bar{M}-1}$, respectively, of $\left\{\alpha_k, f_k, \bar{f}_k\right\}_{k=1}^K$ and $\{\psi_{\bar{m}}\}_{\bar{m}=2}^{\bar{M}-1}$ by minimizing the following NLS criterion:

$$C_1\left(\left\{\alpha_k, f_k, \bar{f}_k\right\}_{k=1}^K, \{\psi_{\bar{m}}\}_{\bar{m}=2}^{\bar{M}-1}\right) = \left\| \mathbf{Y} - \sum_{k=1}^{K} \alpha_k \boldsymbol{\omega}_M(f_k)\boldsymbol{\omega}_{\bar{M}}^T(\bar{f}_k)\mathbf{P} \right\|_F^2, \tag{6.25}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. When the noise $e(m,\bar{m})$ is the zero-mean white Gaussian random process, the NLS estimates of the unknown parameters obtained with MCRELAX coincide with the maximum likelihood (ML) estimates of the parameters. When the noise is colored, the NLS estimates are no longer the ML estimates, but they are still statistically very accurate. The minimization of (6.25) is a very complicated optimization problem. Before we present MCRELAX, let us consider the following preparations.

Assume first that $\left\{\hat{\alpha}_k, \hat{f}_k, \hat{\bar{f}}_k\right\}_{k=1}^K$ are given. Let $\hat{\mathbf{S}}$ denote the matrix whose $m\bar{m}$th element is $\hat{s}(m,\bar{m})$, where $\hat{s}(m,\bar{m})$ is the same as $s(m,\bar{m})$ in (6.21) except that $\{\alpha_k, f_k, \bar{f}_k\}_{k=1}^K$ are replaced by $\{\hat{\alpha}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=1}^K$. Let $\mathbf{y}_{\bar{m}}$ and $\hat{\mathbf{s}}_{\bar{m}}$, respectively, denote the $\bar{m}$th columns of $\mathbf{Y}$ and

$\hat{\mathbf{S}}$. Then minimizing C_1 in (6.25) with respect to $\{\psi_{\bar{m}}\}_{\bar{m}=2}^{\bar{M}-1}$ becomes minimizing

$$C_2\left(\{\psi_{\bar{m}}\}_{\bar{m}=2}^{\bar{M}-1}\right) = \sum_{\bar{m}=2}^{\bar{M}-1} \left\|\mathbf{y}_{\bar{m}} - \hat{\mathbf{s}}_{\bar{m}} \exp\left(j\psi_{\bar{m}}\right)\right\|^2, \tag{6.26}$$

where $\|\cdot\|$ denotes the Euclidean norm, which gives the estimates of $\{\psi_{\bar{m}}\}_{\bar{m}=2}^{\bar{M}-1}$:

$$\hat{\psi}_{\bar{m}} = \text{angle}\left\{\mathbf{s}_{\bar{m}}^H \mathbf{y}_{\bar{m}}\right\}, \quad \bar{m} = 2, 3, \cdots, \bar{M}-1, \tag{6.27}$$

where angle(x) denotes the phase of x and $(\cdot)^H$ denotes the complex conjugate transpose.

Assuming next that $\{\hat{\psi}_{\bar{m}}\}_{\bar{m}=2}^{\bar{M}-1}$ are given. Let $\mathbf{Z}$ be the data matrix motion compensated by using $\hat{\mathbf{P}}$, where $\hat{\mathbf{P}}$ is the same as $\mathbf{P}$ in (6.22) except that $\{\psi_{\bar{m}}\}_{\bar{m}=2}^{\bar{M}-1}$ are replaced by $\{\hat{\psi}_{\bar{m}}\}_{\bar{m}=2}^{\bar{M}-1}$, i.e.,

$$\mathbf{Z} = \mathbf{Y}\hat{\mathbf{P}}^{-1}. \tag{6.28}$$

Since $\hat{\mathbf{P}}$ is a unitary matrix, matrices $\mathbf{Z}$ and $\mathbf{Y}$ have the same Frobenius norm. Thus we can equivalently minimize the following cost function to obtain the estimate $\left\{\hat{\alpha}_k, \hat{f}_k, \hat{\bar{f}}_k\right\}_{k=1}^{K}$ of $\left\{\alpha_k, f_k, \bar{f}_k\right\}_{k=1}^{K}$:

$$C_3\left(\left\{\alpha_k, f_k, \bar{f}_k\right\}_{k=1}^{K}\right) = \left\|\mathbf{Z} - \sum_{k=1}^{K} \alpha_k \boldsymbol{\omega}_M(f_k)\boldsymbol{\omega}_{\bar{M}}^T(\bar{f}_k)\right\|_F^2. \tag{6.29}$$

Let

$$\mathbf{Z}_k = \mathbf{Z} - \sum_{i=1, i\neq k}^{\bar{K}} \hat{\alpha}_i \boldsymbol{\omega}_M(\hat{f}_i)\boldsymbol{\omega}_{\bar{M}}^T(\hat{\bar{f}}_i). \tag{6.30}$$

Then minimizing $\left\|\mathbf{Z}_k - \alpha_k \boldsymbol{\omega}_M(f_k)\boldsymbol{\omega}_{\bar{M}}^T(\bar{f}_k)\right\|_F^2$ with respect to α_k, f_k, and $\bar{f}_k$ yields [6]

$$(\hat{f}_k, \hat{\bar{f}}_k) = \max_{(f_k, \bar{f}_k)} \left|\boldsymbol{\omega}_M^H(f_k)\mathbf{Z}_k\boldsymbol{\omega}_{\bar{M}}^*(\bar{f}_k)\right|^2, \tag{6.31}$$

and

$$\hat{\alpha}_k = \left.\frac{\boldsymbol{\omega}_M^H(f_k)\mathbf{Z}_k\boldsymbol{\omega}_{\bar{M}}^*(\bar{f}_k)}{M\bar{M}}\right|_{f_k=\hat{f}_k, \bar{f}_k=\hat{\bar{f}}_k}. \tag{6.32}$$

Note that $\hat{f}_k$ and $\hat{\bar{f}}_k$ in (6.31) can be obtained as the location of the dominant peak of the 2-D periodogram $\left|\boldsymbol{\omega}_M^H(f_k)\mathbf{Z}_k\boldsymbol{\omega}_{\bar{M}}^*(\bar{f}_k)\right|^2/(M\bar{M})$, which can be efficiently computed by using a 2-D FFT (fast Fourier transform) with the data matrix $\mathbf{Z}_k$ padded with zeros. Then $\hat{\alpha}_k$ is easily computed from the complex height of the peak of $\boldsymbol{\omega}_M^H(f_k)\mathbf{Z}_k\boldsymbol{\omega}_{\bar{M}}^*(\bar{f}_k)/(M\bar{M})$.

It has been shown in [6] that the minimization of C_3 in (6.29) can be efficiently achieved by using the RELAX algorithm which requires a sequence of FFTs. Let $\bar{K}$ denote the intermediate number of scatterers. Then RELAX is described by the following steps:

Step (1): Assume $\bar{K}$=1. Obtain $\left\{\hat{f}_1, \hat{\tilde{f}}_1\right\}$ and $\hat{\alpha}_1$ from $\mathbf{Z}$ (instead of $\mathbf{Z}_k$) by using (6.31) and (6.32), respectively.

Step (2): Assume $\bar{K} = 2$. Obtain $\mathbf{Z}_2$ with (6.30) by using $\hat{f}_1$, $\hat{\tilde{f}}_1$, and $\hat{\alpha}_1$ obtained in Step (1). Obtain $\left\{\hat{f}_2, \hat{\tilde{f}}_2\right\}$ and $\hat{\alpha}_2$ from $\mathbf{Z}_2$ by using (6.31) and (6.32), respectively. Next, compute $\mathbf{Z}_1$ with (6.30) by using $\hat{f}_2$, $\hat{\tilde{f}}_2$, and $\hat{\alpha}_2$ and redetermine $\left\{\hat{f}_1, \hat{\tilde{f}}_1\right\}$ and $\hat{\alpha}_1$ from $\mathbf{Z}_1$.

Iterate the previous two substeps until "practical convergence" is achieved (to be discussed later on).

Step (3): Assume $\bar{K} = 3$. Compute $\mathbf{Z}_3$ with (6.30) by using $\{\hat{f}_i, \hat{\tilde{f}}_i, \hat{\alpha}_i\}_{i=1}^{2}$ obtained in Step (2). Obtain $\left\{\hat{f}_3, \hat{\tilde{f}}_3\right\}$ and $\hat{\alpha}_3$ from $\mathbf{Z}_3$. Next, compute $\mathbf{Z}_1$ with by using $\{\hat{f}_i, \hat{\tilde{f}}_i, \hat{\alpha}_i\}_{i=2}^{3}$ and redetermine $\left\{\hat{f}_1, \hat{\tilde{f}}_1\right\}$ and $\hat{\alpha}_1$ from $\mathbf{Z}_1$. Then compute $\mathbf{Z}_2$ by using $\{\hat{f}_i, \hat{\tilde{f}}_i, \hat{\alpha}_i\}_{i=1,3}$ and redetermine $\left\{\hat{f}_2, \hat{\tilde{f}}_2\right\}$ and $\hat{\alpha}_2$ from $\mathbf{Z}_2$.

Iterate the previous three substeps until "practical convergence".

Remaining Steps: Continue similarly until $\bar{K} = K$. (Whenever K is unknown, it can be estimated from the available data, for instance, by using generalized AIC rules which are particularly tailored to the RELAX method of parameter estimation. See, e.g., [6].)

The "practical convergence" in the iterations of the above RELAX algorithm may be determined by checking the relative change of the cost function $C_3\left(\left\{\hat{f}_k, \hat{\tilde{f}}_k, \hat{\alpha}_k\right\}_{k=1}^{K}\right)$ in (6.29) between two consecutive iterations. In our numerical examples, we terminate the iterative process in each of the above steps when the aforementioned relative change is less than $\varepsilon_1 = 10^{-3}$. Our numerical examples show that the iterations, with this convergence criterion, usually converge in a few steps.

To speed up the convergence rate of MCRELAX, we use the PGA algorithm (see the appendix) to provide the initial estimates of the motion errors. A flow chart of the MCRELAX algorithm is shown in Figure 6.2. MCRELAX can be described with the following steps:

Step 1: Obtain the initial estimates of $\{\hat{\psi}_{\bar{m}}\}_{\bar{m}=2}^{\bar{M}}$ by using the PGA algorithm.

Step 2: Compensate the motion errors by using (6.28).

Step 3: Estimate $\{\hat{\alpha}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=1}^{K}$ from $\mathbf{Z}$ by using the RELAX algorithm.

Step 4: Redetermine $\{\hat{\psi}_{\bar{m}}\}_{\bar{m}=2}^{\bar{M}}$ by using (6.27).

Step 5: Check for practical convergence (see later on). If so, stop; otherwise go to Step 2, but only perform Step (K) of the RELAX algorithm in Step 3.

The practical convergence of the above iterations may be determined by checking the relative change of the cost function C_1 in (6.25) between two consecutive iterations. In our numerical examples, we terminate the iterative process when the aforementioned relative change is less than $\varepsilon_2 = 10^{-3}$. Our numerical examples show that the iterations, with this convergence criterion, also usually converge in a few steps.

Since a minimization is performed at every iteration, the value of the cost function C_1 in (6.25) cannot increase. As a result, under mild conditions, the MCRELAX algorithm is bound to converge to a local minimum of C_1 [8]. Depending on the data parameters, the local minimum may or may not be the global one. To achieve better performance, we could also go to Step (1) instead of Step (K) of Step 3 for the first few iterations of MCRELAX. Doing so does not change the convergence property of MCRELAX.

When only the first three steps of MCRELAX are used for parameter estimation, we refer to the approach as *PGA-RELAX*. We will show in Section 5 that the parameter estimates obtained by using MCRELAX is more accurate than those by using PGA-RELAX.

6.4 The CRB of the Parameter Estimates

We sketch below the derivation of the CRBs for the parameter estimates of the data model in (6.24) when the additive noise is assumed to be a zero-mean colored Gaussian random process with an unknown covariance matrix.

Let

$$\mathbf{y} = \text{vec}\{\mathbf{Y}\}, \tag{6.33}$$

and

$$\mathbf{e} = \text{vec}\{\mathbf{E}\}, \tag{6.34}$$

where vec[$\mathbf{X}$] denotes the vector $[\ \mathbf{x}_1^T\ \ \mathbf{x}_2^T\ \ \cdots\ \ \mathbf{x}_K^T\ ]^T$ with $\{\mathbf{x}_k\}_{k=1}^K$ being the columns of $\mathbf{X}$. Then (6.24) can be written as

$$\mathbf{y} = \sum_{k=1}^{K} \alpha_k \left[\mathbf{P}\boldsymbol{\omega}_{\bar{M}}(\bar{f}_k)\right] \otimes \boldsymbol{\omega}_M(f_k) + \mathbf{e} \triangleq \boldsymbol{\Omega}\boldsymbol{\alpha} + \mathbf{e}, \tag{6.35}$$

where $\otimes$ denotes the Kronecker product,

$$\boldsymbol{\Omega} = \left[\ \left\{\mathbf{P}\boldsymbol{\omega}_{\bar{M}}(\bar{f}_1)\right\} \otimes \boldsymbol{\omega}_M(f_1)\ \ \cdots\ \ \left\{\mathbf{P}\boldsymbol{\omega}_{\bar{M}}(\bar{f}_K)\right\} \otimes \boldsymbol{\omega}_M(f_K)\ \right], \tag{6.36}$$

and

$$\boldsymbol{\alpha} = \left[\ \alpha_1\ \ \alpha_2\ \ \cdots\ \ \alpha_K\ \right]^T. \tag{6.37}$$

Let $\mathbf{Q} = E\{\mathbf{e}\mathbf{e}^H\}$ be the covariance matrix of $\mathbf{e}$, where $E\{\cdot\}$ denotes the expectation. The unknown variables in the likelihood function of $\mathbf{y}$ are the elements of $\mathbf{Q}$, the real and imaginary parts of the amplitudes, the frequency pairs, and the phase errors. The extended Slepian-Bangs' formula for the ijth element of the Fisher information matrix has the form [9, 10]:

$$\{\mathrm{FIM}\}_{ij} = \mathrm{tr}\left(\mathbf{Q}^{-1}\mathbf{Q}_i'\mathbf{Q}^{-1}\mathbf{Q}_j'\right) + 2\mathrm{Re}\left[\left(\boldsymbol{\alpha}^H\boldsymbol{\Omega}^H\right)_i' \mathbf{Q}^{-1}\left(\boldsymbol{\Omega}\boldsymbol{\alpha}\right)_j'\right], \tag{6.38}$$

where $\mathbf{X}_i'$ denotes the derivative of $\mathbf{X}$ with respect to the ith unknown parameter, tr($\mathbf{X}$) denotes the trace of $\mathbf{X}$, and Re($\mathbf{X}$) denotes the real part of $\mathbf{X}$. Note that FIM is a block diagonal matrix since $\mathbf{Q}$ does not depend on the parameters in $(\boldsymbol{\Omega}\boldsymbol{\alpha})$, and $(\boldsymbol{\Omega}\boldsymbol{\alpha})$ does not depend on the elements of $\mathbf{Q}$. Hence the CRB of the estimates of the target features and phase errors can be determined from the second term of the right side of (6.38).

Let

$$\boldsymbol{\eta} = \left[\ \mathrm{Re}^T(\boldsymbol{\alpha})\ \ \mathrm{Im}^T(\boldsymbol{\alpha})\ \ \mathbf{f}^T\ \ \bar{\mathbf{f}}^T\ \ \boldsymbol{\psi}^T\ \right]^T, \tag{6.39}$$

where Im($\mathbf{X}$) denotes the imaginary part of $\mathbf{X}$,

$$\mathbf{f} = \left[\ f_1\ \ f_2\ \ \cdots\ \ f_K\ \right]^T, \tag{6.40}$$

$$\bar{\mathbf{f}} = \left[\ \bar{f}_1\ \ \bar{f}_2\ \ \cdots\ \ \bar{f}_K\ \right]^T, \tag{6.41}$$

and

$$\boldsymbol{\psi} = \begin{bmatrix} \psi_2 & \psi_3 & \cdots & \psi_{\bar{M}-1} \end{bmatrix}^T. \tag{6.42}$$

Let

$$\mathbf{F} = \begin{bmatrix} \boldsymbol{\Omega} & j\boldsymbol{\Omega} & \mathbf{D}_f & \mathbf{D}_{\bar{f}} & \mathbf{D}_\psi \end{bmatrix}, \tag{6.43}$$

where the kth columns of $\mathbf{D}_f$ and $\mathbf{D}_{\bar{f}}$, respectively, are $\alpha_k \partial \left\{ \left[\mathbf{P}\boldsymbol{\omega}_{\bar{M}}(\bar{f}_k) \right] \otimes \boldsymbol{\omega}_M(f_k) \right\} / \partial f_k$ and $\alpha_k \partial \left\{ \left[\mathbf{P}\boldsymbol{\omega}_{\bar{M}}(\bar{f}_k) \right] \otimes \boldsymbol{\omega}_M(f_k) \right\} / \partial \bar{f}_k$, $k = 1, 2, \cdots, K$, and the $\bar{m}$th column of $\mathbf{D}_\psi$ is $\partial \left\{ \sum_{k=1}^{K} \alpha_k [\mathbf{P}\boldsymbol{\omega}_{\bar{M}}(\bar{f}_k)] \otimes \boldsymbol{\omega}_M(f_k) \right\} / \partial \psi_{\bar{m}}$, $\bar{m} = 2, 3, \cdots, \bar{M} - 1$. Then

$$\mathrm{CRB}(\boldsymbol{\eta}) = \left[2\mathrm{Re}(\mathbf{F}^H \mathbf{Q}^{-1} \mathbf{F}) \right]^{-1}. \tag{6.44}$$

6.5 Numerical and Experimental Examples

We first present an experimental example comparing the performance of the MCRELAX algorithm with that of the PGA algorithm. The data matrix is 256×256, which is a portion of the data collected by one of the two apertures of the ERIM's (Environmental Research Institute of Michigan's) DCS IFSAR (interferometric SAR). These data have already been motion compensated by some unknown means. Figure 6.3(a) shows the modulus of the image (in dB and scaled to be between 0 and 255) obtained by applying FFT with Kaiser window and shape parameter 6 to the original data matrix. To test the performance of the MCRELAX and the PGA algorithms, we added phase errors to the data matrix. The phase errors are shown in Figure 6.4 and are generated by an 8th-order polynomial and are similar to the phase errors used in [1]. The SAR image obtained from the data with the added phase errors is shown in Figure 6.3(b). Figures 6.3(c) and (d) show the images obtained after motion compensation by PGA and MCRELAX, respectively. In the MCRELAX algorithm, we assume that there are 50 dominant point scatterers in the image. (We have also tried 30 and the results are similar.) The images in Figures 6.3(c) and (d) are very close to the original one in Figure 6.3(a), which shows that PGA indeed works very well. The comparison of the added phase error estimates obtained by PGA and MCRELAX with the true values is shown in Figure 6.4, which again shows that the phase errors are well estimated by PGA and MCRELAX. However, for perhaps rare cases where strong closely spaced point

scatterers occur in the same range, PGA may not work as well as MCRELAX since PGA is a nonparametric spectral estimation method while MCRELAX is a parametric one and has better resolution than nonparametric methods. To illustrate this point, we embedded three strong point scatterers in one range. The powers of the added scatterers relative to the power of the strongest pixel in Figure 6.3(a) are 14.0, 14.2, and 14.0 dB. We then added the same motion error as in Figure 6.3(b). The so-obtained images with and without the added motion errors are shown in Figures 6.5(a) and (b), respectively. Figures 6.5(c) and (d), respectively, show the images obtained after motion compensation by applying the PGA algorithm and the MCRELAX algorithm to the data used in Figure 6.5(b). We note that the image obtained with the MCRELAX algorithm is better than that obtained with PGA. The comparison of the added phase error estimates obtained by PGA and MCRELAX with the true values is shown in Figure 6.6. We see that the phase errors estimated by using MCRELAX are better than those obtained by using PGA. Yet the number of MATLAB flops required by MCRELAX is about 2×10^3 times as much as that required by PGA. Moreover, if we remove the range line where the strong embedded scatterers are located, PGA works as well as in Figure 6.3, which shows that PGA is indeed a very good motion compensation algorithm.

We next consider an example of a simulated tank. The true features of the simulated tank are shown in Table 6.1, which are obtained from an XPATCH [11] simulated tank. Figure 6.7(a) depicts the modulus of the true RCS's of the scatterers as a function of range and cross-range. The motion errors are identically and independently distributed random variables with uniform distribution between 0 and 2π. The noise sequence is zero-mean circularly symmetric white Gaussian random processes that are uncorrelated with the target parameters and have variance $\sigma^2 = 20$. The modulus of the RCS's of the scatterers obtained via FFT without windowing but with zero-padding in the presence of the phase errors and noise are shown in Figure 6.7(b). Figure 6.7(c) shows the modulus of the RCS's obtained by applying FFT to the data that is motion compensated by PGA. Figures 6.7(d) and (e), respectively, show the modulus of the RCS's obtained by PGA-RELAX and MCRELAX. We note that PGA can reduce the motion errors significantly and is very robust. Note also

that both PGA-RELAX and MCRELAX can correctly resolve all eight scatterers, including those that cannot be resolved by FFT. Moreover, the parameter estimates obtained with MCRELAX are closer to the true ones than those obtained with PGA-RELAX. (Note, for example, the two largest scatterers.)

Let us now consider the CRBs of the target parameters. In target classification applications, it is often the relative positions of the scatterers that are more important than a common shift in range or cross-range. Let δ be a common shift of $\{f_k\}_{k=1}^K$, i.e.,

$$\breve{f}_k = f_k - \delta, \tag{6.45}$$

such that

$$\sum_{k=1}^{K} \breve{f}_k = 0. \tag{6.46}$$

Then

$$\delta = \frac{1}{K}\sum_{k=1}^{K} f_k. \tag{6.47}$$

Hence $\breve{f} = [\ \breve{f}_1 \ \cdots \ \breve{f}_K\ ]^T$ is related to $\mathbf{f}$ by the following linear transformation

$$\breve{\mathbf{f}} = \mathbf{T}\mathbf{f}, \tag{6.48}$$

where

$$\mathbf{T} = \frac{1}{K}\begin{bmatrix} K-1 & -1 & \cdots & -1 \\ -1 & K-1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & K-1 \end{bmatrix}. \tag{6.49}$$

Similarly, let

$$\breve{\bar{\mathbf{f}}} = \mathbf{T}\bar{\mathbf{f}}, \tag{6.50}$$

and

$$\bar{\delta} = \frac{1}{K}\sum_{k=1}^{K} \bar{f}_k. \tag{6.51}$$

Then the CRB matrix of $\breve{\mathbf{f}}$ is related to $\mathbf{f}$ by

$$\mathrm{CRB}(\breve{\mathbf{f}}) = \mathbf{T}\mathrm{CRB}(\mathbf{f})\mathbf{T}^T, \tag{6.52}$$

and

$$\mathrm{CRB}(\delta) = \frac{1}{K^2}\mathbf{1}^T\mathrm{CRB}(\mathbf{f})\mathbf{1}, \tag{6.53}$$

where

$$\mathbf{1} = [\ 1 \quad 1 \quad \cdots \quad 1\ ]^T. \tag{6.54}$$

Similarly,

$$\mathrm{CRB}(\breve{\bar{\mathbf{f}}}) = \mathbf{T}\mathrm{CRB}(\bar{\mathbf{f}})\mathbf{T}^T, \tag{6.55}$$

and

$$\mathrm{CRB}(\bar{\delta}) = \frac{1}{K^2}\mathbf{1}^T\mathrm{CRB}(\bar{\mathbf{f}})\mathbf{1}. \tag{6.56}$$

Table 6.2 compares the CRBs of $\{\hat{\alpha}_k\}_{k=1}^K$, $\left\{\hat{f}_k\right\}_{k=1}^K$, $\left\{\hat{\breve{f}}_k\right\}_{k=1}^K$, $\left\{\hat{\bar{f}}_k\right\}_{k=1}^K$, $\left\{\hat{\breve{\bar{f}}}_k\right\}_{k=1}^K$, $\hat{\delta}$, and $\hat{\bar{\delta}}$ for the aforementioned simulated tank example for both the case of known and the case of unknown motion errors. Note that the case of known motion errors is equivalent to the case of no motion errors. We note that as compared to the case of known motion errors, the unknown motion errors have the most significant impact on the CRBs of $\left\{\hat{\bar{f}}_k\right\}_{k=1}^K$ and $\hat{\bar{\delta}}$, which are almost the same. The unknown motion errors have little effect on other target parameters, including the relative cross-range positions $\left\{\hat{\breve{\bar{f}}}_k\right\}_{k=1}^K$. Hence when the motion errors are unknown, the main errors in $\left\{\hat{\bar{f}}_k\right\}_{k=1}^K$ are due to the error of a common shift.

Finally, Table 6.3 shows the comparisons of the CRBs and the mean-squared errors (MSEs) of the parameter estimates obtained by using PGA-RELAX and MCRELAX for the simulated tank example. Note that only relative positions and the common shifts are considered. The MSEs are obtained from 100 Monte-Carlo trials. We note that the MSEs of the parameter estimates obtained with MCRELAX are very close to the corresponding CRBs, while those obtained with PGA-RELAX are not. Thus, iterating Steps 2 through 5 in MCRELAX can improve the accuracy of the parameter estimates. Yet the number of MATLAB flops required by MCRELAX is only about twice as much as that required by PGA-RELAX for this example.

6.6 Conclusions

We have studied the problem of extracting target features via SAR in the presence of uncompensated aperture motion errors. A parametric data model of a spotlight-mode SAR system has been established. The Cramér-Rao bounds for the parameters of the data model have also been derived. The CRB analysis shows that the unknown motion errors can significantly affect the accuracy of a common shift of the scatterer positions in cross-range direction, but have little effect on other target parameters including the accuracy of the relative positions in the range direction. A relaxation-based MCRELAX algorithm for estimating both target features and motion errors has been devised. Simulation results have shown that the MSEs of the parameter estimates can approach the corresponding CRBs. We have also shown with a couple of examples that MCRELAX can simply be used for motion compensation only and can give good motion compensation results.

Appendix - The PGA Algorithm

To make this paper self-contained, we briefly describe the PGA algorithm. The algorithm we describe below is slightly different from the original PGA algorithm presented in [4] in that we avoid the ambiguity problems described in Section 2 by imposing $\psi_0 = \psi_1 = 0$. The PGA algorithm can be summarized with the following steps (with Steps 2 to 4 being iterative):

Step 1: Obtain $\mathbf{V}$, the 2-D FFT of the data matrix $\mathbf{Y}$.

Step 2: For each row $\mathbf{v}_m$ of $\mathbf{V}$, $m = 0, 1, \cdots, M-1$, perform the following operations:

(a) Select the dominant peak of the modulus $|\mathbf{v}_m|$ of $\mathbf{v}_m$ and shift it to the origin (to remove its frequency offset). More specifically, let

$$|v_{m,\bar{m}^\dagger}| = \max\left\{ \ |v_{m,0}|, \ \ |v_{m,1}|, \ \ \cdots, \ \ |v_{m,\bar{M}-1}| \ \right\}, \tag{6.57}$$

where $\bar{m}^\dagger$ denote the cross-range position of the dominant peak. Then the shifted vector has

the form

$$\tilde{\mathbf{v}}_m = \left[\begin{array}{cccc} \tilde{v}_{m,0} & \tilde{v}_{m,1} & \cdots & \tilde{v}_{m,\bar{M}-1} \end{array} \right] = \left[\begin{array}{ccccccccc} v_{m,\bar{m}^\dagger} & v_{m,\bar{m}^\dagger+1} & \cdots & v_{m,\bar{M}-1} & v_{m,0} & v_{m,1} & \cdots & v_{m,\bar{m}^\dagger-1} \end{array} \right]. \tag{6.58}$$

(b) Window the circularly shifted imagery. Let $2d^\dagger + 1$ be the window length. Then the windowed data vector has the form

$$\breve{\mathbf{v}}_m = \left[\begin{array}{cccc} \breve{v}_{m,0} & \breve{v}_{m,1} & \cdots & \breve{v}_{m,\bar{M}-1} \end{array} \right] = \left[\begin{array}{ccccccccc} \tilde{v}_{m,0} & \cdots & \tilde{v}_{m,d^\dagger} & 0 & \cdots & 0 & \tilde{v}_{m,\bar{M}-d^\dagger} & \cdots & \tilde{v}_{m,\bar{M}-1} \end{array} \right]. \tag{6.59}$$

Here the window length is identical for all range bins $\{\tilde{\mathbf{v}}_m\}_{m=0}^{M-1}$ and is determined from

$$s_{\bar{m}} = \sum_{m=0}^{M-1} |\tilde{v}_{m,\bar{m}}|^2, \quad \bar{m} = 0, 1, \cdots, \bar{M} - 1, \tag{6.60}$$

by thresholding $s_{\bar{m}}$ at the point 10 dB down from its peak, s_0, then increasing this width by 50% [4].

(c) Form an $M \times \bar{M}$ matrix $\mathbf{G}$ whose mth row contains the 1-D inverse FFT of the windowed data vector $\breve{\mathbf{v}}_m$. Let $\mathbf{G} = [\ \mathbf{g}_1, \ \cdots, \ \mathbf{g}_{\bar{M}-1}\ ]$, where $\mathbf{g}_{\bar{m}}$ is the $\bar{m}$th column of $\mathbf{G}$.

Step 3: Compute the difference estimates $\Delta\hat{\phi}_{\bar{m}} = \hat{\phi}_{\bar{m}} - \hat{\phi}_{\bar{m}-1}$ of the motion phase errors $\{\phi_{\bar{m}}\}_{\bar{m}=2}^{\bar{M}-1}$ (of the current iteration) according to

$$\Delta\hat{\phi}_{\bar{m}} = \Delta\tilde{\phi}_{\bar{m}} - \Delta\tilde{\phi}_1, \quad \bar{m} = 1, 2, \cdots, \bar{M} - 1, \tag{6.61}$$

where

$$\Delta\tilde{\phi}_{\bar{m}} = \text{angle}\left\{\mathbf{g}_{\bar{m}-1}^H \mathbf{g}_{\bar{m}}\right\}. \tag{6.62}$$

Obtain $\hat{\phi}_{\bar{m}}$ by using $\hat{\phi}_0 = 0, \quad \bar{m} = 1, 2, \cdots, \bar{M} - 1$. (Note that we guarantee $\hat{\phi}_0 = \hat{\phi}_1 = 0$.)

Step 4: Compensate for the motion errors (of the current iteration)

$$\hat{y}(m, \bar{m}) = y(m, \bar{m}) \exp\left(-j\hat{\phi}_{\bar{m}}\right), \quad m = 0, 1, \cdots, M - 1, \bar{m} = 0, 1, \cdots, \bar{M} - 1. \tag{6.63}$$

Step 5: Iterate Steps 1 through 4 with $y(m, \bar{m})$ being replaced by $\hat{y}(m, \bar{m})$ until convergence.

Note that the estimate $\hat{\psi}_{\bar{m}}$ of the motion error $\psi_{\bar{m}}$, $\bar{m} = 2, 3, \cdots, \bar{M} - 1$, is obtained by summing the $\hat{\phi}_{\bar{m}}$ obtained from all iterations. We remark that the PGA algorithm is robust since it searches for a single peak for each row vector $\mathbf{v}_m$. Even if a peak location is mistakenly determined, it only causes a constant shift in $\Delta\tilde{\phi}_m$ in (6.62), which may be corrected by (6.61).

ACKNOWLEDGMENTS

The authors thank Maj. T. Burns for providing us with the ERIM data. The authors are also grateful to Dr. P. H. Eichel for his helpful comments on the PGA algorithm.

Reference

[1] P. H. Eichel, D. C. Ghiglia, and C. V. Jakowatz, Jr., "Speckle processing methods for synthetic-aperture -radar phase correction," *Optics Letters*, vol. 14, pp. 1–3, January 1989.

[2] P. H. Eichel and C. V. Jakowatz, Jr., "Phase-gradient algorithm as an optimal estimator of the phase derivative," *Optics Letters*, vol. 14, pp. 1101–1103, October 1989.

[3] C. V. Jakowatz, Jr. and D. E. Wahl, "Eigenvector method for maximum-likelihood estimation of phase errors in synthetic-aperture-radar imagery," *Journal of the Optical Society of America*, vol. 10, pp. 2539–2546, December 1993.

[4] D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and C. V. Jakowatz, Jr., "Phase gradient autofocus-a robust tool for high resolution SAR phase correction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, pp. 827–835, July 1994.

[5] C. V. Jakowatz, Jr., D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. A. Thompson, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach.* Norwell, MA: Kluwer Academic Publishers, 1996.

[6] J. Li and P. Stoica, "Efficient mixed-spectrum estimation with applications to target feature extraction," *IEEE Transactions on Signal Processing*, vol. 44, pp. 281–295, February 1996.

[7] D. C. Munson, Jr., J. D. O'Brien, and W. K. Jenkins, "A tomographic formulation of spotlight-mode synthetic aperture radar," *Proceedings of the IEEE*, vol. 71, pp. 917 –925, August 1983.

[8] V. G. Karmanov, *Programmation Mathematique.* Editions Mir, Moscow, 1977.

[9] W. Bangs, *Array processing with generalized beamformers.* Ph.D. dissertation, Yale University, New Haven, CT, 1971.

[10] P. Stoica and R. L. Moses, *Introduction to Spectral Analysis.* Englewood Cliffs, NJ: Prentice-Hall, 1997.

[11] D. J. Andersh, M. Hazlett, S. W. Lee, D. D. Reeves, D. P. Sullivan, and Y. Chu, "XPATCH: a high-frequency electromagnetic scattering prediction code and environment for complex three-dimensional objects," *IEEE Antennas and Propagation Magazine*, vol. 36, pp. 65–69, February 1994.

kth scatterer	1	2	3	4	5	6	7	8
$\mathrm{Re}\{\alpha_k\}$	9.2643	-3.5000	4.4339	-2.1447	-1.4363	1.7064	1.1555	1.2068
$\mathrm{Im}\{\alpha_k\}$	-4.9633	-9.6967	-1.4916	0.3269	-1.2229	-0.6923	-0.7064	-0.6122
f_k	0.1016	0.1406	0.1016	0.0078	0.1016	0.0312	-0.1328	0.0938
$\tilde{f}_k$	-0.0293	0.0488	-0.0059	-0.0059	0.1191	-0.0215	-0.0371	-0.0684

Table 6.1: True parameter values of the simulated tank.

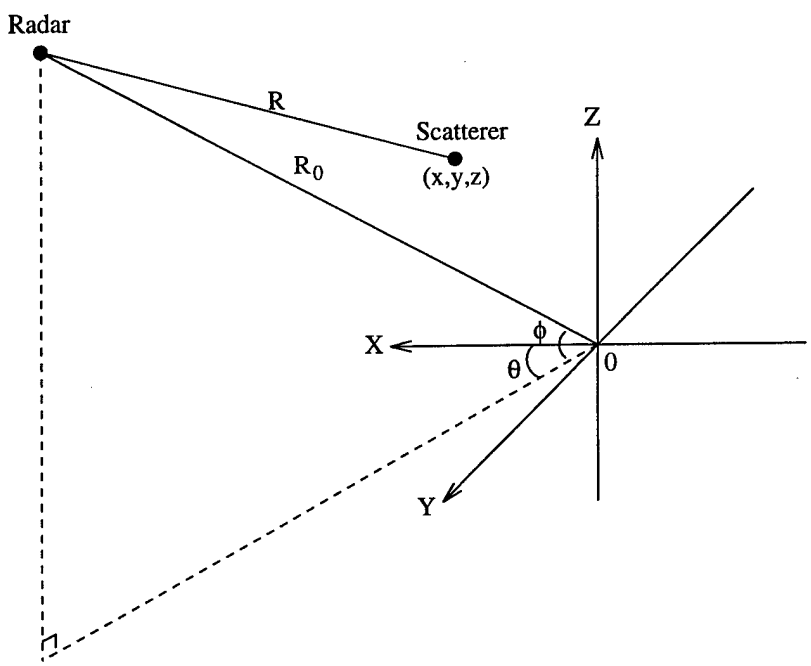


Figure 6.1: Data collection geometry in a spotlight-mode SAR.

kth scatterer	1	2	3	4	5	6	7	8
CRB1	-2.13	-7.78	-2.28	-9.58	-9.85	-9.48	-11.15	-8.62
CRB2	-3.32	-11.19	-3.13	-9.72	-11.04	-9.58	-11.24	-9.54

(a)

kth scatterer	1	2	3	4	5	6	7	8
CRB1	-74.71	-75.49	-67.94	-58.38	-60.77	-56.66	-57.97	-57.61
CRB2	-74.80	-75.53	-67.98	-58.39	-60.85	-56.67	-57.98	-57.92
CRB3	-68.31	-68.84	-66.75	-58.70	-61.30	-57.23	-58.92	-58.54
CRB4	-68.43	-68.97	-66.88	-58.71	-61.38	-57.24	-58.94	-58.81

(b)

kth scatterer	1	2	3	4	5	6	7	8
CRB1	-44.20	-44.25	-44.16	-44.14	-43.87	-44.09	-44.07	-43.85
CRB2	-67.11	-75.58	-59.51	-60.03	-60.52	-58.46	-57.98	-55.27
CRB3	-65.83	-68.27	-57.61	-60.41	-60.19	-59.01	-58.81	-54.72
CRB4	-66.10	-68.97	-58.71	-60.55	-61.23	-59.07	-58.88	-56.02

(c)

CRB1	CRB2	CRB3	CRB4
-69.75	-69.86	-44.22	-69.88

(d)

Table 6.2: Comparison of the CRBs (in dB) of the target parameters for the simulated tank example. (a) CRBs of $\{\alpha_k\}$ obtained by assuming the motion errors unknown (CRB1) and known (CRB2). (b) CRBs of $\{f_k\}$ and $\left\{\breve{f}_k\right\}$, obtained by assuming the motion errors unknown (CRB1 and CRB3, respectively) and known (CRB2 and CRB4, respectively). (c) CRBs of $\left\{\bar{f}_k\right\}$ and $\left\{\breve{\bar{f}}_k\right\}$, obtained by assuming the motion errors unknown (CRB1 and CRB3, respectively) and known (CRB2 and CRB4, respectively). (d) CRBs of δ and $\bar{\delta}$, obtained by assuming the motion errors unknown (CRB1 and CRB3, respectively) and known (CRB2 and CRB4, respectively).

kth scatterer	1	2	3	4	5	6	7	8
MSE1	5.03	6.58	1.76	-5.64	-2.29	-5.36	-8.44	-7.86
MSE2	-2.95	-6.71	-2.56	-9.39	-8.72	-9.45	-11.50	-9.20
CRB	-2.13	-7.78	-2.28	-9.58	-9.85	-9.48	-11.15	-8.62

(a)

kth scatterer	1	2	3	4	5	6	7	8
MSE1	-58.67	-58.53	-59.18	-54.83	-52.04	-44.88	-56.04	-57.68
MSE2	-69.72	-70.14	-67.11	-58.65	-61.44	-56.85	-59.05	-58.45
CRB	-68.31	-68.84	-66.75	-58.70	-61.30	-57.23	-58.92	-58.54

(b)

kth scatterer	1	2	3	4	5	6	7	8
MSE1	-57.04	-56.37	-54.64	-54.81	-51.48	-47.00	-54.50	-49.15
MSE2	-65.82	-68.49	-57.77	-60.86	-60.12	-58.56	-59.21	-54.92
CRB	-65.83	-68.27	-57.61	-60.41	-60.19	-59.01	-58.81	-54.72

(c)

	δ	$\bar{\delta}$
MSE1	-59.21	-10.54
MSE2	-63.69	-42.36
CRB	-69.75	-44.22

(d)

Table 6.3: Comparison of the CRBs (in dB) of the parameters of the simulated tank with the MSEs obtained by PGA-RELAX (MSE1) and MCRELAX (MSE2). (a) For $\{\alpha_k\}$. (b) For $\left\{\breve{f}_k\right\}$. (c) For $\left\{\breve{\bar{f}}_k\right\}$. (d) For δ and $\bar{\delta}$.

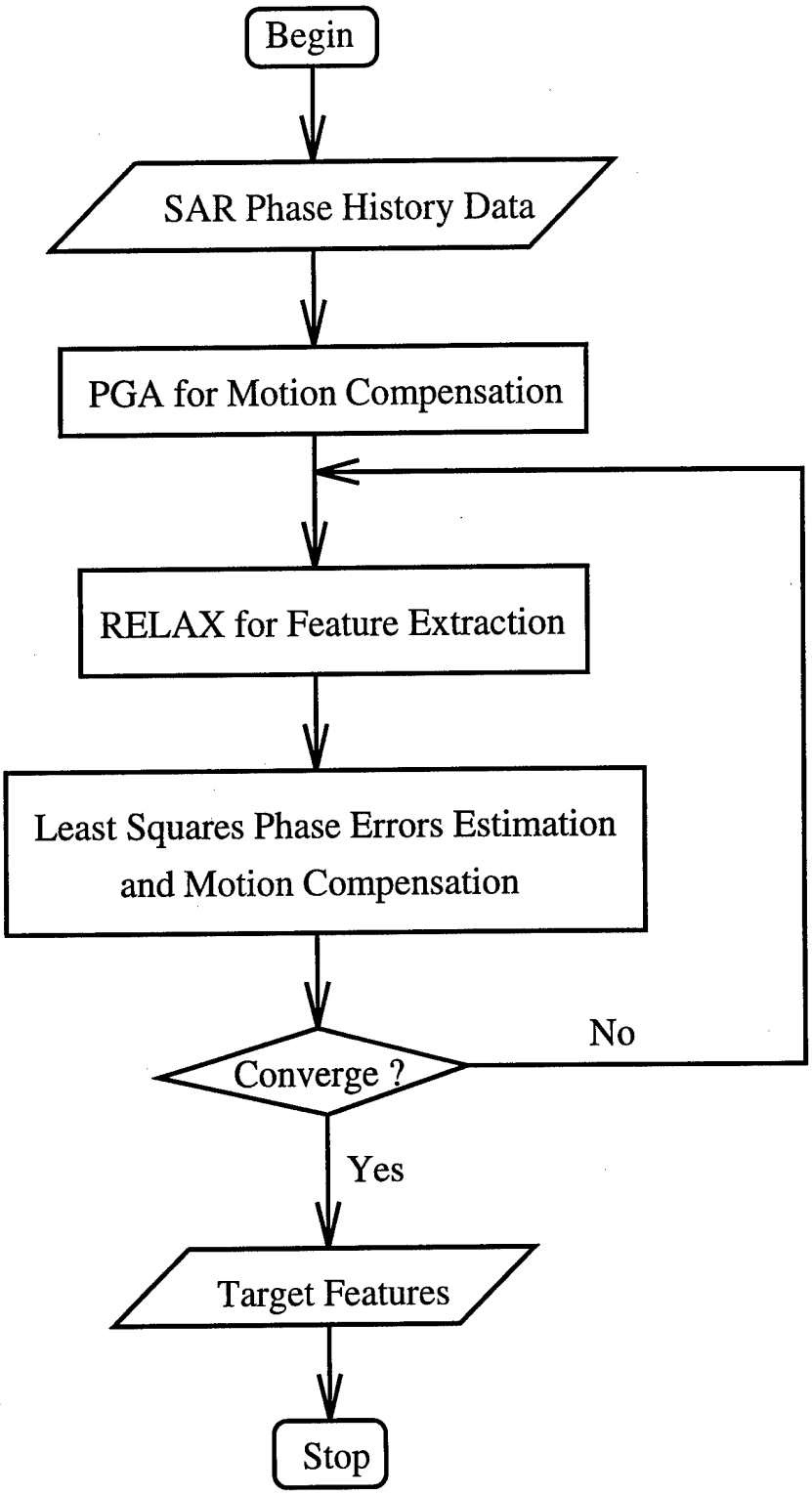
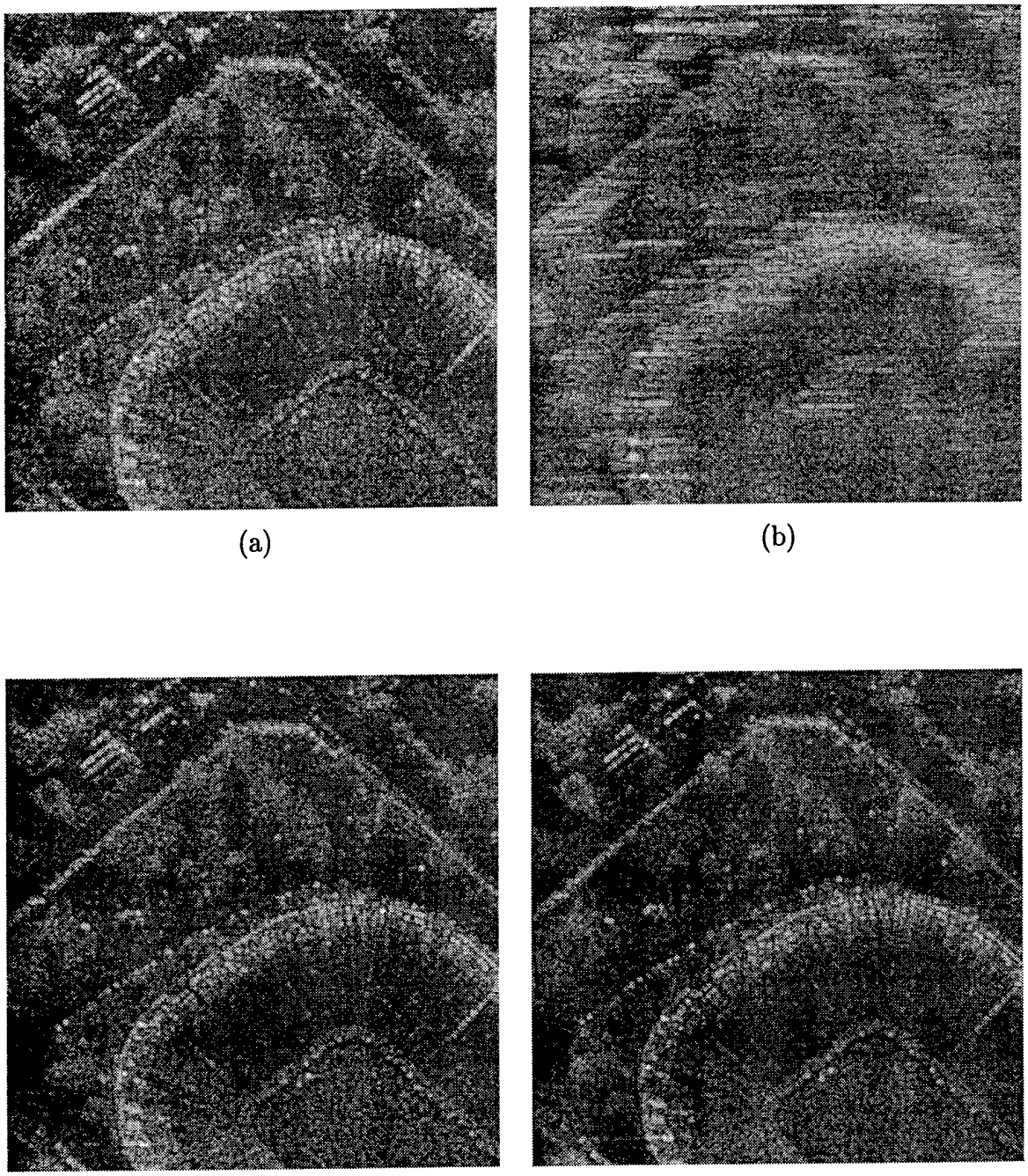


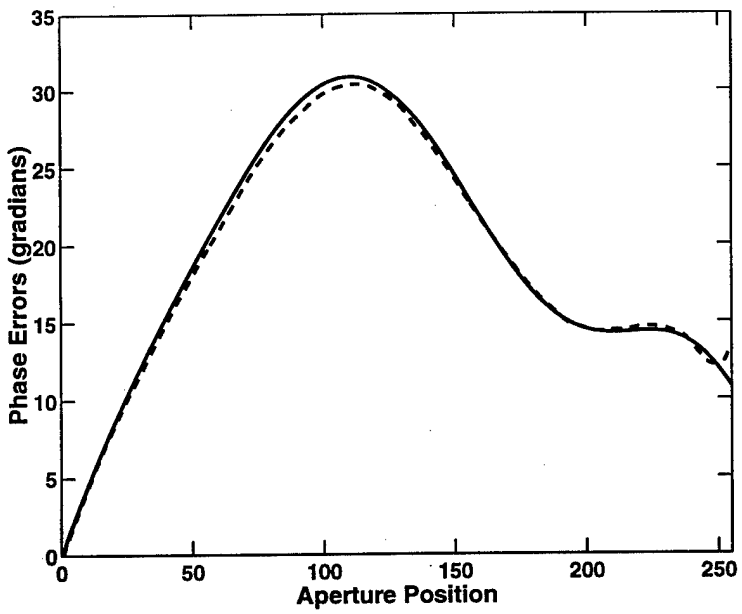
Figure 6.2: The flow chart of the MCRELAX algorithm.



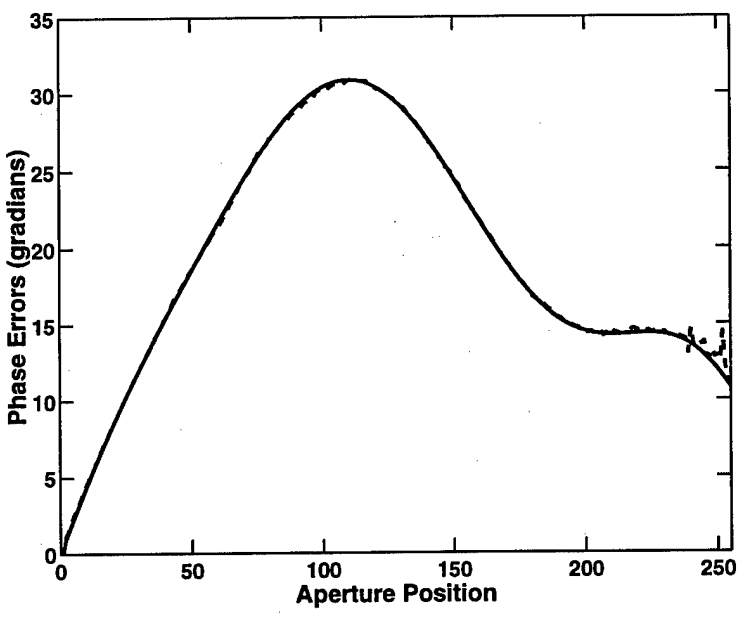
(a) (b)

(c) (d)

Figure 6.3: Comparison of (a) the original image with (b) the image with added motion errors before motion compensation and the images with added motion errors after motion compensation obtained by (c) the PGA algorithm and (d) the MCRELAX algorithm by using the dashed lines in Figure 4.

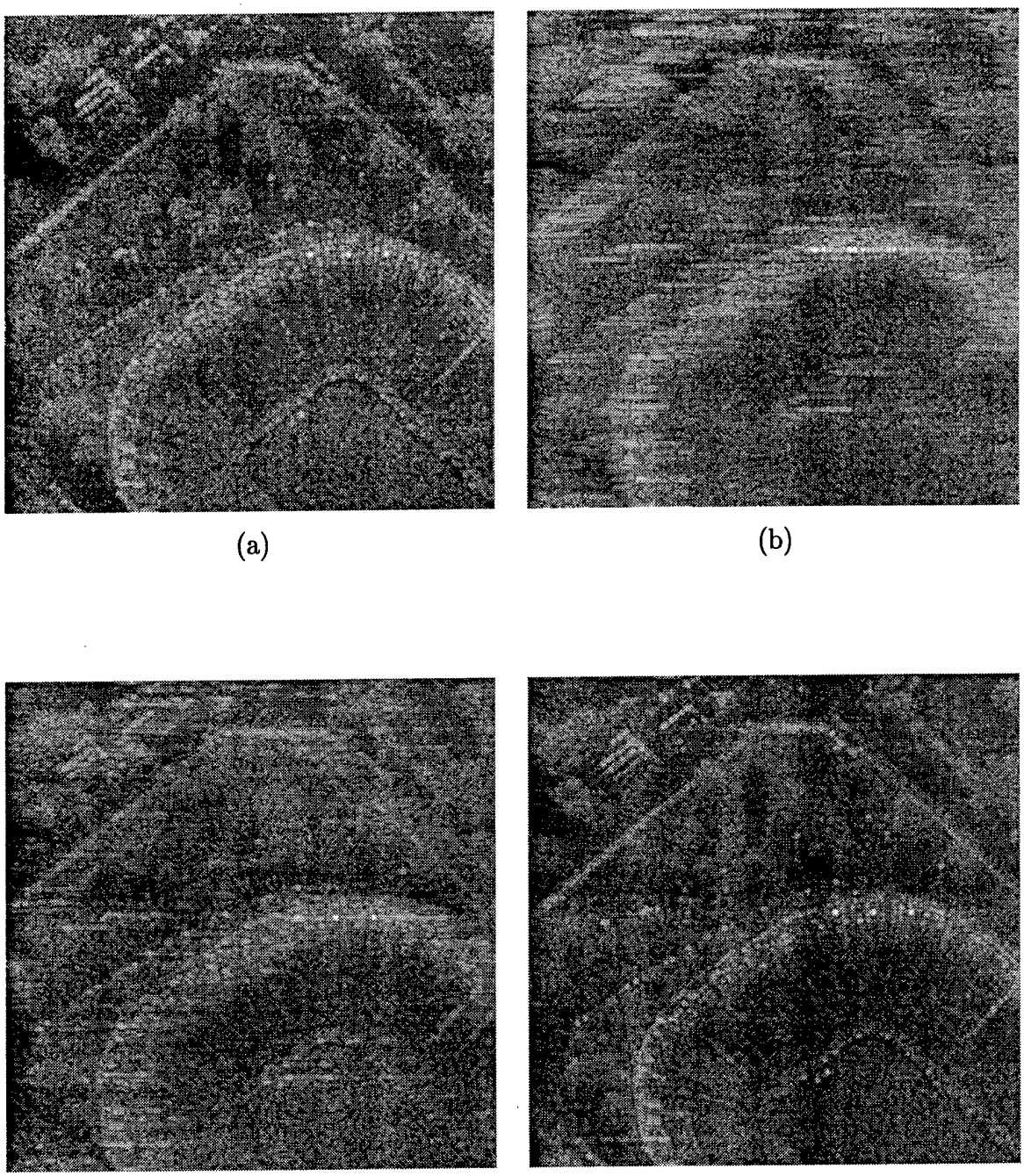


(a)



(b)

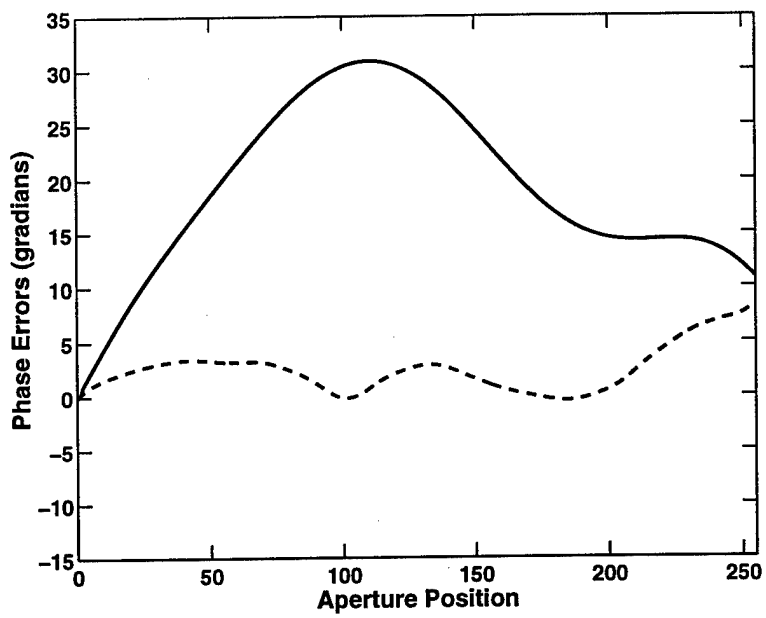
Figure 6.4: Comparison of the true phase errors (solid line) with their estimates (dashed line) obtained by (a) PGA and (b) MCRELAX.



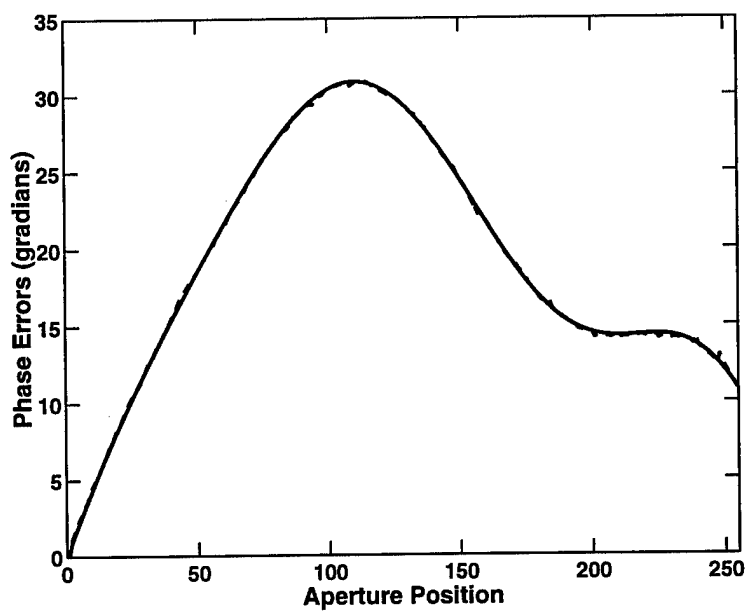
(a) (b)

(c) (d)

Figure 6.5: Comparison of (a) the original image (with three embedded point scatterers) with (b) the image with added motion errors before motion compensation and the images with added motion errors after motion compensation obtained by (c) the PGA algorithm and (d) the MCRELAX algorithm by using the dashed line in Figure 6.



(a)



(b)

Figure 6.6: Comparison of the true phase errors (solid line) with their estimates (dash line) obtained by (a) PGA and (b) MCRELAX.

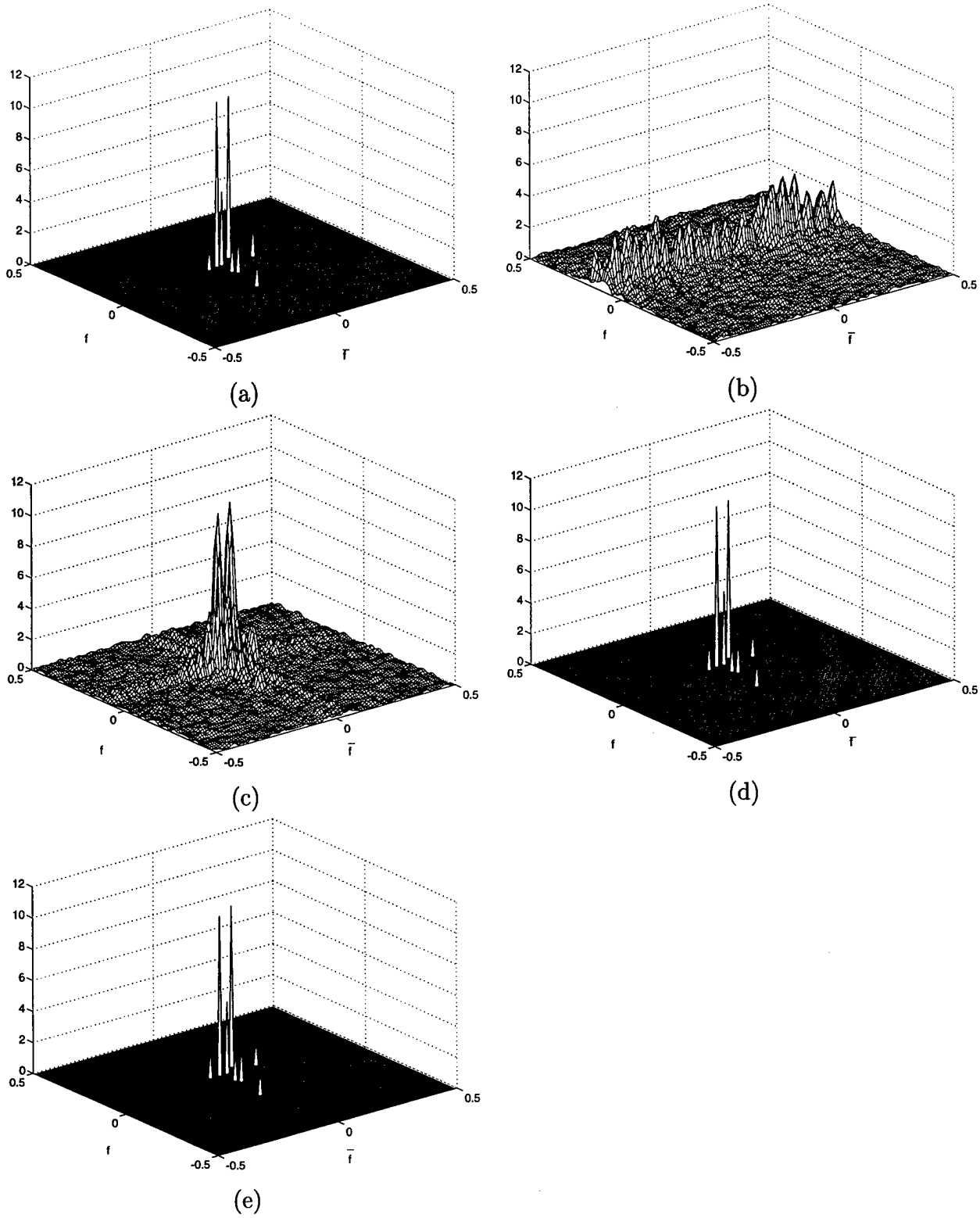


Figure 6.7: The simulated tank example with $M = \bar{M} = 32$ and $\sigma^2 = 20$. (a) True $|\alpha_k|$ vs. f_k and $\bar{f}_k$. (b) $|\hat{\alpha}(f, \bar{f})|$ with added motion errors vs. f and $\bar{f}$. (c) $|\hat{\alpha}(f, \bar{f})|$ vs. f and $\bar{f}$ obtained by PGA. (d) $|\hat{\alpha}_k|$ vs. $\hat{f}_k$ and $\hat{\bar{f}}_k$ obtained by PGA-RELAX, (e) $|\hat{\alpha}_k|$ vs. $\hat{f}_k$ and $\hat{\bar{f}}_k$ obtained by MCRELAX.

7. Autofocus and Super Resolution SAR Image Formation

7.1 Introduction

Synthetic aperture radar (SAR) can produce high resolution images of targets or scenes of interest by transmitting signals with large bandwidths and utilizing the relative motion between the radar and the objects to be imaged. Since its invention in the early 1950's [1], SAR imaging technology has now been widely used in many military and civilian applications [2]. Conventional SAR has two common imaging modes, i.e., stripmap vs. spotlight modes. The stripmap mode is more efficient when used for coarse-resolution mapping of large regions, while the spotlight mode is a practical choice for fine-resolution imaging of localized areas. Major technical issues associated with SAR imaging include motion compensation and image formation. In this chapter, we will address the issue of autofocus and super resolution image formation from spotlight-mode SAR data in the presence of uncompensated phase errors across the synthetic aperture.

The ideal data collection geometry for the spotlight-mode SAR imaging is called "turntable imaging" where a stationary radar illuminating a uniformly rotating object [3] at a fixed distance. Fine cross-range resolution is obtained by utilizing the Doppler frequency gradient generated by the relative rotational motion between the object and the radar. In turntable imaging, the distance between the radar and the rotational axis of the object to be imaged is fixed. However, for airborne or spaceborne SAR, it is not easy to keep the distance between the radar and the scene center (or the reference point used for dechirping) a constant or to have an exact knowledge about their relative distances. Any uncertainty in the relative distance will produce a demodulation timing error which will result in a range shift and a phase error in the range-compressed data. With modern advanced onboard integrated navigation systems, including inertial measurement units (IMUs), the Global Positioning System (GPS), and the ring laser gyro technology, the amount of range shift is typically a small fraction of a range resolution cell [4] and hence can be ignored. However, the phase error across the synthetic aperture is usually large enough to blur SAR images. In addition to

the platform position uncertainty, propagation through atmospheric turbulence [5] or system phase instability will also produce phase errors across the synthetic aperture. Usually, the platform position uncertainty will produce low-frequency phase errors while the propagation and system instability will induce phase errors that generally have a high-frequency content.

Improving the accuracy of the navigation systems is cost expensive and can only deal with phase errors due to the platform position uncertainty. Instead, *autofocus* algorithms, which derive phase errors directly from collected phase history data, offer an attractive alternative to remove the phase errors independent of the error source. Many autofocus algorithms have been proposed in the SAR literature, (see, for example, [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]), for correcting phase errors across the synthetic aperture. A two subaperture-based approach, referred to as Map Drift, was proposed in [6] to estimate quadratic phase error coefficients by finding the peak of the cross-correlations between the two low-resolution intensity images formed using the two subapertures. Map Drift is very simple and computationally efficient but can only compensate quadratic phase errors. Multiple subaperture-based approaches were proposed in [7, 8, 9], which are the extensions of Map Drift to estimate high-order polynomial phase errors. However, the order of the phase error polynomials is still very limited because the higher the order, the more the subapertures needed, and hence the narrower the subapertures, and in turn, the cross-correlations of the resulting low-resolution images yield less accurate estimates of the polynomial coefficients. Another subaperture-based algorithm, referred to as the the phase difference (PD) algorithm, was proposed in [10] to compensate quadratic phase errors. PD and Map Drift differ in the manner in which they manipulate the subaperture data to yield the estimates of phase polynomial coefficients. PD obtains the error estimates from the non-coherent average (across range) of the Fourier transform of the conjugate product of the range-compressed data from the two subapertures. Unlike other subaperture-based algorithms [6, 7, 8, 9], PD is non-iterative. PD can also be extended to multiple subapertures in the way similar to that used in [7, 8, 9]. However, like [7, 8, 9], its capability to handle high-order phase errors is still very limited. A very robust algorithm, referred to as phase gradient autofocus (PGA), was first proposed in [11] and later refined in [12, 13]. PGA is based on inverse filtering

[4] but takes advantage of the redundancy of the phase error function by averaging across many range cells. Unlike the above subaperture-based algorithms [6, 7, 8, 9, 10], PGA is not model-based and hence can be used to estimate any high order phase errors. It has been shown in [14] that PGA is very robust over a variety of scene content and phase error distributions. However, fully automatic SAR image autofocus using PGA may be risky in some special cases [16] since the performance of PGA is sensitive to the choice of the threshold needed by the automatic windowing step or the size reduction rate needed by the progressive windowing scheme. Moreover, PGA suffers from error accumulation across the aperture. Prominent point processing (PPP) was also proposed for the autofocus of SAR images [15, 18]. Like PGA, PPP is not based on the modeling of phase errors. However, PPP relies heavily on the existence of isolated dominant scatterers, which is rarely met in practice. A good review of the above algorithms can be found in [18]. As pointed out in [18], comparison and evaluation of the above algorithms are made difficult by the fact that performance generally varies with scene content and phase error characteristics. For example, the single PPP algorithm is more robust than any other aforementioned algorithms against high-order phase errors but is more sensitive than any other algorithms to the scene content. It seems that PGA makes a good tradeoff between the robustness over scene content and phase error distributions.

In [17], a fully parametric algorithm, referred to as MCRELAX, was proposed for simultaneous autofocus and target feature extraction. MCRELAX assumes a two-dimensional (2-D) sinusoidal model for the target signal but assumes nothing (*arbitrary* unknowns) for the phase error distribution. MCRELAX is an alternating optimization approach to a nonlinear least-squares (NLS) fitting criterion that uses RELAX [19] for target feature extraction. To speed up the convergence, PGA is used in MCRELAX to provide the initial phase error estimates. When PGA fails to provide a reliable initial condition, MCRELAX may not converge or converge very slowly to a good solution. In addition, since MCRELAX uses all target scatterers to estimate both the phase errors and target features, when the size of the scene to be imaged is large, MCRELAX becomes computationally too expensive.

In this chapter, an autofocus algorithm, referred to as MCCLEAN, is proposed for cor-

recting synthetic aperture phase errors in SAR. It is a parametric algorithm based on the same data model as used in MCRELAX. The computational core of the algorithm is the CLEAN algorithm [20, 19], which involves only a sequence of 2-D FFT operations. Like the single PPP algorithm, MCCLEAN is more robust than any other algorithms against high-order phase errors. However, MCCLEAN does not rely on the existence of isolated dominant scatterers as the single PPP algorithm does. Compared to MCRELAX [17], MCCLEAN has better convergence property (no separate initialization step is required) and is computationally much more efficient when used as an independent autofocus approach for the SAR imaging of a large scene. For certain kinds of scene content and phase error distributions, MCCLEAN performs better than the well-known PGA algorithm. We also present a modified relaxation based algorithm, which has a similar structure as MCCLEAN, for simultaneous autofocus and super resolution target feature extraction of a small scene or small region of interest (ROI) in a large scene. Super resolution SAR images can then be formed from these estimated target features by using data extrapolation and FFT [21].

The remainder of the chapter is organized as follows. In Section 7.2, we derive the data model and formulate the problem of interest. Some preparations and the MCCLEAN algorithm are presented in Sections 7.3 and 7.4, respectively. Several experimental examples are provided in Section 7.5 to illustrate the performance of MCCLEAN. Finally, Section 7.6 contains our conclusions.

7.2 Data Model and Problem Formulation

A broadside data collection geometry in a spotlight-mode SAR is shown in Figure 7.1. The XYZ coordinate system is centered on a small patch of ground, where a target is located. The coordinate origin is referred to as the scene center or reference point. The angle θ and ϕ denote the azimuth and elevation angles, respectively, of the radar relative to the XYZ coordinate system. The distance between the radar and the scene center is denoted by R_0. The ground is illuminated by a narrow beam from the moving radar that moves along the θ direction but with ϕ and R_0 fixed. In Figure 7.1, R denotes the distance between the radar and a scatterer at the position (x, y, z).

The range R of the scatterer located at (x, y, z) can be written as

$$R = \left[(R_0 \cos\theta \cos\phi - x)^2 + (R_0 \sin\theta \cos\phi - y)^2 + (R_0 \sin\phi - z)^2\right]^{1/2}. \tag{7.1}$$

When the distance from the radar to the target is much larger than the size of the target (i.e., $\frac{x}{R_0} \ll 1$, $\frac{y}{R_0} \ll 1$, $\frac{z}{R_0} \ll 1$) and θ is very small, a good approximation of R can be found by using the Taylor series expansion, which has the form [17]

$$R \approx R_0 - \tilde{x} \cos\theta \cos\phi - \tilde{y} \sin\theta \cos\phi, \tag{7.2}$$

where

$$\tilde{x} = x + z \tan\phi - \frac{x^2 \sin^2\phi + y^2 + z^2 \cos^2\phi - xz \sin 2\phi}{2R_0 \cos\phi}, \tag{7.3}$$

and

$$\tilde{y} = y + \frac{xy \cos\phi + yz \sin\phi}{R_0}. \tag{7.4}$$

Note that the second term of the right side of (7.3) is due to the range layover of the scatterer with non-zero height z [4]. The third term of the right side of (7.3) and the second term of the right side of (7.4) are due to the range curvature effect [4].

The range resolution of a radar is determined by the radar bandwidth. To achieve high resolution in range, the radar must transmit wideband pulses, which are often linear frequency modulated (FM) chirp pulses. A normalized chirp pulse can be written as

$$s(t) = \exp\left[j(2\pi f_0 t + \pi\gamma t^2)\right], \quad |t| \leq T_0/2, \tag{7.5}$$

where f_0 denotes the carrier frequency, γ denotes the chirp rate, and T_0 denotes the pulse width. At look angle θ, the signal returned by a scatterer located at (x, y, z) has the form

$$r(t, \theta) = \delta_{x,y,z} \exp\left\{j\left[2\pi f_0 \left(t - \frac{2R}{c}\right) + \pi\gamma \left(t - \frac{2R}{c}\right)^2\right]\right\}, \tag{7.6}$$

where t denotes the *fast time*, $\delta_{x,y,z}$ is determined by the radar cross section (RCS) of the scatterer, and c denotes the speed of light.

In practice, a *dechirp-on-receive* approach is used to considerably reduce the sampling rate requirement of the A/D converters [4, 18]. The dechirped signal $d(t, \theta)$ is obtained by mixing

$r(t,\theta)$ with a reference signal $\exp\left\{-j[2\pi f_0(t-\frac{2R_0}{c})+\pi\gamma(t-\frac{2R_0}{c})^2]\right\}$, which is the conjugate of the normalized received signal of the scene center. Since the term $\exp\left[j\frac{4\pi\gamma}{c^2}(R-R_0)^2\right]$ is usually close to a constant, hence it can be ignored or partially removed [22, 18]. In this case, $d(t,\theta)$ has the form

$$d(t,\theta)=\delta_{x,y,z}\exp\left\{-j\frac{4\pi}{c}\left[f_0+\gamma\left(t-\frac{2R_0}{c}\right)\right](R-R_0)\right\}. \tag{7.7}$$

Inserting R in (7.2) into (7.7), we have

$$d(t,\theta)=\delta_{x,y,z}\exp\left[j(\tilde{x}t_x+\tilde{y}t_y)\right], \tag{7.8}$$

where

$$t_x=\frac{4\pi}{c}\left[f_0+\gamma(t-\frac{2R_0}{c})\right]\cos\phi\cos\theta, \tag{7.9}$$

and

$$t_y=\frac{4\pi}{c}\left[f_0+\gamma(t-\frac{2R_0}{c})\right]\cos\phi\sin\theta. \tag{7.10}$$

Note that $d(t,\theta)$ in (7.8) is a 2-D sinusoidal signal. The frequency pair of the 2-D sinusoid corresponds to the 2-D location $\{\tilde{x},\tilde{y}\}$ of the scatterer, while the amplitude is proportional to its RCS. Note that $(\tilde{x},\tilde{y})$ is the ambiguous location due to the layover and range curvature effects [4] and is not the true location (x,y,z) of the scatterer. The range curvature effect can be neglected for large R_0. However, this 2-D SAR imaging system cannot distinguish the scatterers located at (x,y,z) and $(\tilde{x},\tilde{y},0)$ and combines these scatterers into one scatterer. The scatterers we referred to below are those resulted from such combinations.

When a target has multiple scatterers with distinct $(\tilde{x},\tilde{y})$, $d(t,\theta)$ in (7.8) will be a sum of sinusoids. The 2-D locations and RCS's of the target scatterers are the 2-D target features. Since usually the samples on the t and θ axes are uniformly spaced, the samples of t_x and t_y occur at the points of a polar grid. Hence Polar-to-Cartesian interpolation (referred to as polar reformatting) is needed for the samples of t_x and t_y to occur at rectangular grid points [22]. After polar reformatting, the signal reflected by a radar target that consists of K 2-D

point scatterers in an ideal SAR system can be described as:

$$s(m,\bar{m}) = \sum_{k=1}^{K} \alpha_k \exp\left[j2\pi(mf_k + \bar{m}\bar{f}_k)\right], \quad m = 0,1,\cdots,M-1, \quad \bar{m} = 0,1,\cdots,\bar{M}-1, \tag{7.11}$$

where the complex amplitude α_k and the 2-D frequency pair $\{f_k, \bar{f}_k\}$, respectively, are proportional to the radar cross section (RCS) and the 2-D location (range and cross-range) of the kth scatterer of the target, and M and $\bar{M}$ denote the numbers of the available data samples in range and cross-range, respectively.

In the derivations of the above data model, we have assumed that the radar moves along the θ direction with R_0 known exactly. For a practical airborne or spaceborne SAR system, however, the distance between the moving radar and the reference point may not be known exactly due to platform position uncertainty [4]. Also, the round-trip time delay between the radar and the target may be somewhat random due to atmospheric turbulence [5], which has the same effect on SAR imaging as when R_0 is in error. For a given θ, let $\Delta R_0(\theta)$ denote the unknown distance error between the radar and the reference point. $\Delta R_0(\theta)$ will result in a shift and a constant phase error in the range compressed data corresponding to the look angle θ. The shift amount is typically a small fraction of the range resolution cell [4] and hence can be ignored. However, since f_0 tends to be very large in practical SAR systems, even a small $\Delta R_0(\theta)$ would cause a large phase error across the synthetic aperture. By neglecting the effect of polar reformatting, the signal obtained by a realistic SAR system can be described as

$$y(m,\bar{m}) = s(m,\bar{m}) \exp(j\psi_{\bar{m}}) + e(m,\bar{m}), \tag{7.12}$$

where $\{\psi_{\bar{m}}\}_{\bar{m}=0}^{\bar{M}-1}$ are the phase errors due to unknown $\Delta R_0(\theta)$ and system instability and $e(m,\bar{m})$ denotes the unknown noise. Let $\mathbf{Y}$ and $\mathbf{E}$ denote the $M \times \bar{M}$ matrices whose $m\bar{m}$th elements are $y(m,\bar{m})$ and $e(m,\bar{m})$, respectively. Let

$$\mathbf{P}(\mathbf{\Psi}) = \mathrm{diag}\left\{ \exp(j\psi_0), \ \exp(j\psi_1), \ \cdots, \ \exp(j\psi_{\bar{M}-1}) \right\}, \tag{7.13}$$

and

$$\boldsymbol{\omega}_M(f_k) = \left[1 \ \ \exp(j2\pi f_k) \ \ \cdots \ \ \exp[j2\pi(M-1)f_k] \right]^T, \tag{7.14}$$

where $(\cdot)^T$ denotes the transpose and

$$\mathbf{\Psi} = \begin{bmatrix} \psi_0 & \psi_1 & \cdots & \psi_{\bar{M}-1} \end{bmatrix}^T. \tag{7.15}$$

Then (7.12) can be written in the following matrix form:

$$\mathbf{Y} = \sum_{k=1}^{K} \alpha_k \boldsymbol{\omega}_M(f_k)\boldsymbol{\omega}_{\bar{M}}^T(\bar{f}_k)\mathbf{P}(\mathbf{\Psi}) + \mathbf{E}. \tag{7.16}$$

Our problem of interest herein is to estimate $\mathbf{\Psi}$ from $\mathbf{Y}$ in (7.16) for autofocus or both $\mathbf{\Psi}$ and $\{\alpha_k, f_k, \bar{f}_k\}_{k=1}^K$ for simultaneous autofocus and super resolution SAR image formation.

7.3 Preparations

The estimates $\left\{\hat{\alpha}_k, \hat{f}_k, \hat{\bar{f}}_k\right\}_{k=1}^K$ and $\hat{\mathbf{\Psi}}$, respectively, of $\left\{\alpha_k, f_k, \bar{f}_k\right\}_{k=1}^K$ and $\mathbf{\Psi}$ can be obtained by minimizing the following NLS criterion:

$$C_1\left(\left\{\alpha_k, f_k, \bar{f}_k\right\}_{k=1}^K, \mathbf{\Psi}\right) = \left\|\mathbf{Y} - \sum_{k=1}^{K} \alpha_k \boldsymbol{\omega}_M(f_k)\boldsymbol{\omega}_{\bar{M}}^T(\bar{f}_k)\mathbf{P}(\mathbf{\Psi})\right\|_F^2, \tag{7.17}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. It has been shown [19] that the NLS estimates so-obtained are statistically very accurate for both white and colored noise $e(m, \bar{m})$. However, the minimization of (7.17) is a very complicated optimization problem. The alternating minimization approach can be used for the above optimization problem, which repeats the two steps of phase error estimation as well as compensation and target feature extraction. Below we briefly describe the above two steps, which will lay down the basis for the presentation of our MCCLEAN algorithm.

Phase Error Estimation and Compensation

Assume that $\left\{\hat{\alpha}_k, \hat{f}_k, \hat{\bar{f}}_k\right\}_{k=1}^K$ are given. Let $\hat{\mathbf{S}}$ denote the matrix whose $m\bar{m}$th element is $\hat{s}(m, \bar{m})$, where $\hat{s}(m, \bar{m})$ is the same as $s(m, \bar{m})$ in (7.11) except that $\{\alpha_k, f_k, \bar{f}_k\}_{k=1}^K$ are replaced by $\{\hat{\alpha}_k, \hat{f}_k, \hat{\bar{f}}_k\}_{k=1}^K$. Let $\mathbf{y}_{\bar{m}}$ and $\hat{\mathbf{s}}_{\bar{m}}$, respectively, denote the $\bar{m}$th columns of $\mathbf{Y}$ and $\hat{\mathbf{S}}$. Then minimizing C_1 in (7.17) with respect to $\mathbf{\Psi}$ becomes minimizing

$$C_2(\mathbf{\Psi}) = \sum_{\bar{m}=0}^{\bar{M}-1} \left\|\mathbf{y}_{\bar{m}} - \hat{\mathbf{s}}_{\bar{m}} \exp\left(j\psi_{\bar{m}}\right)\right\|^2, \tag{7.18}$$

where $\|\cdot\|$ denotes the Euclidean norm, which gives the estimates of $\{\psi_{\bar{m}}\}_{\bar{m}=0}^{\bar{M}-1}$:

$$\hat{\psi}_{\bar{m}} = \text{angle}\left\{\mathbf{s}_{\bar{m}}^H \mathbf{y}_{\bar{m}}\right\}, \quad \bar{m} = 0, 1, \cdots, \bar{M}-1, \tag{7.19}$$

where $\hat{\psi}_{\bar{m}}$ is the $(\bar{m}+1)$th element of $\hat{\mathbf{\Psi}}$, angle(x) denotes the phase of x, and $(\cdot)^H$ denotes the conjugate transpose.

Once the phase error estimates $\hat{\mathbf{\Psi}}$ are obtained, phase error compensation can be done very easily. Let $\mathbf{Z}$ be the data matrix phase compensated by using $\mathbf{P}(\hat{\mathbf{\Psi}})$, where $\mathbf{P}(\hat{\mathbf{\Psi}})$ is the same as $\mathbf{P}(\mathbf{\Psi})$ in (7.13) except that $\mathbf{\Psi}$ is replaced by $\hat{\mathbf{\Psi}}$, i.e.,

$$\mathbf{Z} = \mathbf{Y}\mathbf{P}^{-1}(\hat{\mathbf{\Psi}}). \tag{7.20}$$

Since $\mathbf{P}(\hat{\mathbf{\Psi}})$ is a unitary matrix, minimizing C_1 in (7.17) is equivalent to minimizing the following cost function

$$C_3\left(\left\{\alpha_k, f_k, \bar{f}_k\right\}_{k=1}^{K}\right) = \left\|\mathbf{Z} - \sum_{k=1}^{K} \alpha_k \boldsymbol{\omega}_M(f_k)\boldsymbol{\omega}_{\bar{M}}^T(\bar{f}_k)\right\|_F^2, \tag{7.21}$$

which becomes a standard NLS target feature extraction problem.

Feature Extraction

The target feature estimates $\left\{\hat{\alpha}_k, \hat{f}_k, \hat{\bar{f}}_k\right\}_{k=1}^{K}$ of $\left\{\alpha_k, f_k, \bar{f}_k\right\}_{k=1}^{K}$ can be obtained by minimizing C_3 in (7.21) via the relaxation-based approaches, such as CLEAN [19, 20] or RELAX [19]. Before we summarize the two algorithms for later use, let us first present the following preparations.

Assume $\left\{\hat{\alpha}_i, \hat{f}_i, \hat{\bar{f}}_i\right\}_{i=1, i\neq k}^{\bar{K}}$ are given, where $\bar{K}$ denotes the intermediate number of scatterers. Let

$$\mathbf{Z}_k = \mathbf{Z} - \sum_{i=1, i\neq k}^{\bar{K}} \hat{\alpha}_i \boldsymbol{\omega}_M(\hat{f}_i)\boldsymbol{\omega}_{\bar{M}}^T(\hat{\bar{f}}_i). \tag{7.22}$$

Then minimizing $\left\|\mathbf{Z}_k - \alpha_k \boldsymbol{\omega}_M(f_k)\boldsymbol{\omega}_{\bar{M}}^T(\bar{f}_k)\right\|_F^2$ with respect to α_k, f_k, and $\bar{f}_k$ yields [19]

$$(\hat{f}_k, \hat{\bar{f}}_k) = \arg\max_{(f_k, \bar{f}_k)} \left|\boldsymbol{\omega}_M^H(f_k)\mathbf{Z}_k\boldsymbol{\omega}_{\bar{M}}^*(\bar{f}_k)\right|^2, \tag{7.23}$$

and

$$\hat{\alpha}_k = \left.\frac{\boldsymbol{\omega}_M^H(f_k)\mathbf{Z}_k\boldsymbol{\omega}_{\bar{M}}^*(\bar{f}_k)}{M\bar{M}}\right|_{f_k=\hat{f}_k, \bar{f}_k=\hat{\bar{f}}_k}. \tag{7.24}$$

Note that $\hat{f}_k$ and $\hat{\bar{f}}_k$ in (7.23) can be obtained as the location of the dominant peak of the 2-D periodogram $\left|\boldsymbol{\omega}_M^H(f_k)\mathbf{Z}_k\boldsymbol{\omega}_{\bar{M}}^*(\bar{f}_k)\right|^2/(M\bar{M})$, which can be efficiently computed by using 2-D FFT (fast Fourier transform) with the data matrix $\mathbf{Z}_k$ padded with zeros. Then $\hat{\alpha}_k$ is easily computed from the complex height of the peak of $\boldsymbol{\omega}_M^H(f_k)\mathbf{Z}_k\boldsymbol{\omega}_{\bar{M}}^*(\bar{f}_k)/(M\bar{M})$.

With the above preparations, now we summarize the CLEAN and RELAX algorithms.

Summary of the CLEAN Algorithm

Step (1): Assume $\bar{K}$=1. Obtain $\left\{\hat{f}_1, \hat{\bar{f}}_1\right\}$ and $\hat{\alpha}_1$ from $\mathbf{Z}$ by using (7.23) and (7.24), respectively.

Step (2): Assume $\bar{K} = 2$. Obtain $\mathbf{Z}_2$ with (7.22) by using $\hat{f}_1$, $\hat{\bar{f}}_1$, and $\hat{\alpha}_1$ obtained in Step (1). Obtain $\left\{\hat{f}_2, \hat{\bar{f}}_2\right\}$ and $\hat{\alpha}_2$ from $\mathbf{Z}_2$ by using (7.23) and (7.24), respectively.

Step (3): Assume $\bar{K} = 3$. Compute $\mathbf{Z}_3$ with (7.22) by using $\{\hat{f}_i, \hat{\bar{f}}_i, \hat{\alpha}_i\}_{i=1}^2$ obtained in Step (2). Obtain $\left\{\hat{f}_3, \hat{\bar{f}}_3\right\}$ and $\hat{\alpha}_3$ from $\mathbf{Z}_3$.

Remaining Steps: Continue similarly until $\bar{K} = K$.

Summary of the RELAX Algorithm

Step (1): Assume $\bar{K}$=1. Obtain $\left\{\hat{f}_1, \hat{\bar{f}}_1\right\}$ and $\hat{\alpha}_1$ from $\mathbf{Z}$ by using (7.23) and (7.24), respectively.

Step (2): Assume $\bar{K} = 2$. Obtain $\mathbf{Z}_2$ with (7.22) by using $\hat{f}_1$, $\hat{\bar{f}}_1$, and $\hat{\alpha}_1$ obtained in Step (1). Obtain $\left\{\hat{f}_2, \hat{\bar{f}}_2\right\}$ and $\hat{\alpha}_2$ from $\mathbf{Z}_2$ by using (7.23) and (7.24), respectively. Next, compute $\mathbf{Z}_1$ with (7.22) by using $\hat{f}_2$, $\hat{\bar{f}}_2$, and $\hat{\alpha}_2$ and redetermine $\left\{\hat{f}_1, \hat{\bar{f}}_1\right\}$ and $\hat{\alpha}_1$ from $\mathbf{Z}_1$.

Iterate the previous two substeps until "practical convergence" is achieved (to be discussed later on).

Step (3): Assume $\bar{K} = 3$. Compute $\mathbf{Z}_3$ with (7.22) by using $\{\hat{f}_i, \hat{\bar{f}}_i, \hat{\alpha}_i\}_{i=1}^2$ obtained in Step (2). Obtain $\left\{\hat{f}_3, \hat{\bar{f}}_3\right\}$ and $\hat{\alpha}_3$ from $\mathbf{Z}_3$. Next, compute $\mathbf{Z}_1$ with by using $\{\hat{f}_i, \hat{\bar{f}}_i, \hat{\alpha}_i\}_{i=2}^3$ and redetermine $\left\{\hat{f}_1, \hat{\bar{f}}_1\right\}$ and $\hat{\alpha}_1$ from $\mathbf{Z}_1$. Then compute $\mathbf{Z}_2$ by using $\{\hat{f}_i, \hat{\bar{f}}_i, \hat{\alpha}_i\}_{i=1,3}$ and redetermine $\left\{\hat{f}_2, \hat{\bar{f}}_2\right\}$ and $\hat{\alpha}_2$ from $\mathbf{Z}_2$.

Iterate the previous three substeps until "practical convergence".

Remaining Steps: Continue similarly until $\bar{K} = K$.

The "practical convergence" in the iterations of the above RELAX algorithm may be determined by checking whether the relative change (ε) of the cost function $C_3\left(\left\{\hat{f}_k, \hat{\bar{f}}_k, \hat{\alpha}_k\right\}_{k=1}^{K}\right)$ in (7.21) between two consecutive iterations is less than some threshold (say, 10^{-3}).

We remark that CLEAN is not a super resolution algorithm but RELAX is [19]. CLEAN is computationally more efficient than RELAX but its resolution and estimation accuracy are inferior to RELAX [19]. Yet we have found that for the purpose of autofocus of a large scene by using a small number of dominant scatterers only, using CLEAN and RELAX yields similarly focused SAR images. Hence we choose to use CLEAN in the autofocus algorithm to be presented in Section 7.4.

7.4 The MCCLEAN Algorithm

Usually, the number of scatterers needed to provide satisfactory phase error estimates is much smaller than the true number of scatterers in a SAR image, especially for a large scene. For a large scene, we can assume a small K to estimate the phase errors. The focused SAR image of the whole scene can then be formed by applying FFT to the phase compensated data. Since the FFT image of the so-obtained whole scene is usually well focused, we can cut out some small regions of interest (ROIs) for further autofocus and super resolution image formation. This two-layer processing scheme seems more reasonable in practice.

The flow chart of the proposed algorithm is shown in Figure 7.2, which is quite different from that of MCRELAX [17]. Note that in Figure 7.2, instead of two separate steps of phase error estimation and compensation and feature extraction of all K scatterers as used in MCRELAX, the new algorithm estimates both the phase errors and target features via an incremental refinement procedure. In other words, the estimates of both the phase errors and target features are steadily improved by increasing the number of intermediate scatterers $\tilde{K}$ from 1 to some desired number (automatically determined by the algorithm). Before we present the steps of MCCLEAN, let us copy and define some notations:

$\tilde{K}$: counter of the number of outer iterations (also the number of intermediate scatterers);

$\bar{I}$: counter of the number of inner iterations;

$\mathbf{Z}$: $M \times \bar{M}$ matrix denoting the compensated phase history data;

$\triangle\hat{\mathbf{\Psi}}_{\tilde{K}}(\bar{I})$: $\bar{M} \times 1$ vector denoting the *incremental* phase error estimates for the $\bar{I}$th inner iteration of the $\tilde{K}$th outer iteration;

$\triangle\hat{\mathbf{\Psi}}_{\tilde{K}}$: $\bar{M} \times 1$ vector representing the *incremental* phase error estimates accumulated within the $\tilde{K}$th outer iteration.

With the above preparations, the MCCLEAN algorithm can be summarized as follows:

Step 0: Let $\mathbf{Z} = \mathbf{Y}$.

Step 1: Assume $\tilde{K} = 1$.

Substep (a): Obtain $\left\{\hat{f}_k, \hat{\tilde{f}}_k, \hat{\alpha}_k\right\}_{k=1}$ by using Step (1) of CLEAN and then let $\bar{I} = 1$.

Substep (b): Estimate the *incremental* phase error $\triangle\hat{\mathbf{\Psi}}_1(\bar{I})$ and then let $\mathbf{Z} = \mathbf{Z}\mathbf{P}^{-1}(\triangle\hat{\mathbf{\Psi}}_1(\bar{I}))$.

Substep (c): Re-estimate $\left\{\hat{f}_k, \hat{\tilde{f}}_k, \hat{\alpha}_k\right\}_{k=1}$ via CLEAN by assuming $K = \tilde{K}$.

Substep (d): If the inner convergence (to be discussed later on) is achieved, then let $\triangle\hat{\mathbf{\Psi}}_1 = \sum_{i=1}^{\bar{I}} \triangle\hat{\mathbf{\Psi}}_1(i)$ and go to the next step; otherwise, let $\bar{I} = \bar{I} + 1$ and then go to Substep 1(b).

Substep (e): If the outer convergence (to be discussed later on) is achieved, then go to the Final Step; otherwise, go to the next step.

Step 2: Assume $\tilde{K} = 2$.

Substep (a): Obtain $\left\{\hat{f}_k, \hat{\tilde{f}}_k, \hat{\alpha}_k\right\}_{k=2}$ by using Step (2) of CLEAN and then let $\bar{I} = 1$.

Substep (b): Estimate the *incremental* phase error $\triangle\hat{\mathbf{\Psi}}_2(\bar{I})$ and then let $\mathbf{Z} = \mathbf{Z}\mathbf{P}^{-1}(\triangle\hat{\mathbf{\Psi}}_2(\bar{I}))$.

Substep (c): Re-estimate $\left\{\hat{f}_k, \hat{\tilde{f}}_k, \hat{\alpha}_k\right\}_{k=1}^{2}$ via CLEAN by assuming $K = \tilde{K}$.

Substep (d): If the inner convergence is achieved, then let $\triangle\hat{\mathbf{\Psi}}_2 = \sum_{i=1}^{\bar{I}} \triangle\hat{\mathbf{\Psi}}_2(i)$ and go to the next step; otherwise, let $\bar{I} = \bar{I} + 1$ and then go to Substep 2(b).

Substep (e): If the outer convergence is achieved, then go to the Final Step; otherwise, go to the next step.

Step 3: Assume $\tilde{K} = 3$.

Substep (a): Obtain $\left\{\hat{f}_k, \hat{\tilde{f}}_k, \hat{\alpha}_k\right\}_{k=3}$ by using Step (3) of CLEAN and then let $\bar{I} = 1$.

Substep (b): Estimate the *incremental* phase error $\triangle\hat{\mathbf{\Psi}}_3(\bar{I})$ and then let $\mathbf{Z} = \mathbf{Z}\mathbf{P}^{-1}(\triangle\hat{\mathbf{\Psi}}_3(\bar{I}))$.

Substep (c): Re-estimate $\left\{\hat{f}_k, \hat{\tilde{f}}_k, \hat{\alpha}_k\right\}_{k=1}^{3}$ via CLEAN by assuming $K = \tilde{K}$.

Substep (d): If the inner convergence is achieved, then let $\triangle\hat{\mathbf{\Psi}}_3 = \sum_{i=1}^{\bar{I}} \triangle\hat{\mathbf{\Psi}}_3(i)$ and go to the next step; otherwise, let $\bar{I} = \bar{I} + 1$ and then go to Substep 3(b).

Substep (e): If the outer convergence is achieved, then go to the Final Step; otherwise, go to the next step.

Continuing Steps: Continue similarly by increasing $\tilde{K}$ until the outer convergence is achieved.

Final Step: Form the focused SAR images of the whole scene by applying FFT to the phase compensated data.

The number of scatterers used by MCCLEAN to estimate the phase errors is determined by the algorithm automatically as follows. After the convergence of MCCLEAN at Step $\tilde{K}$, the phase error estimate $\hat{\mathbf{\Psi}}_{\tilde{K}}$ of $\mathbf{\Psi}$ is

$$\hat{\mathbf{\Psi}}_{\tilde{K}} = \sum_{k=1}^{\tilde{K}} \triangle\hat{\mathbf{\Psi}}_k. \tag{7.25}$$

The inner or outer convergence can be determined by checking the contribution of the current *incremental* phase error estimates to the current phase error estimates for each inner or outer iteration, respectively. Consider, for example, Step $\tilde{K}$ of the above MCCLEAN algorithm. After $\bar{I}$ inner iterations, we calculate

$$\epsilon_{\tilde{K}}(\bar{I}) = \frac{\| \triangle\hat{\mathbf{\Psi}}_{\tilde{K}}(\bar{I}) \|^2}{\| \hat{\mathbf{\Psi}}_{\tilde{K}-1} + \sum_{\bar{i}=1}^{\bar{I}} \triangle\hat{\mathbf{\Psi}}_{\tilde{K}}(\bar{i}) \|^2}, \tag{7.26}$$

which is the ratio of the norm of the vector denoting the current *incremental* phase error estimates to that of the current phase error estimates. If $\epsilon_{\tilde{K}}(\bar{I})$ is less than some threshold value, say $\epsilon_{\tilde{K}}$, then we declare that the inner convergence is achieved. (In our experimental examples, we use a variable threshold $\epsilon_{\tilde{K}} = 0.1/\tilde{K}$ to speed up the convergence of MCCLEAN.) After the inner convergence with $\bar{I}$ inner iterations, the accumulated *incremental* phase error estimates for Step $\tilde{K}$ is

$$\triangle\hat{\mathbf{\Psi}}_{\tilde{K}} = \sum_{\bar{i}=1}^{\bar{I}} \triangle\hat{\mathbf{\Psi}}_{\tilde{K}}(\bar{i}), \tag{7.27}$$

and the ratio of the norm of the vector denoting the accumulated *incremental* phase error

estimates within Step $\tilde{K}$ to that of the current phase error estimates is

$$\epsilon(\tilde{K}) = \frac{\| \triangle \hat{\mathbf{\Psi}}_{\tilde{K}} \|^2}{\| \hat{\mathbf{\Psi}}_{\tilde{K}} \|^2}. \tag{7.28}$$

If $\epsilon(\tilde{K})$ is less than some threshold value, say ϵ_o, then we declare that the outer convergence is achieved. (In our experimental examples, we use $\epsilon_o = 0.01$.)

We remark that MCCLEAN is a highly automatic approach and it is more robust than any other existing algorithms against high-order phase errors. When the target contains several dominant scatterers (not necessarily well isolated strong dominant scatterers), MCCLEAN is a very effective autofocus algorithm.

Other algorithms can also be constructed from Figure 7.2 by changing the feature extraction methods and converge control strategies for the inner and outer iteration loops. If we use the same converge control strategy but only replace the CLEAN algorithm in Substep (c) of each step of MCCLEAN with RELAX, we obtain a new algorithm. Yet we prefer using CLEAN since CLEAN is computationally more efficient than RELAX and when only a small number of scatterers are used for estimating the phase errors, CLEAN has similar performance as RELAX. This does not contradict the fact that RELAX outperforms CLEAN in estimation accuracy and resolution capability since when there are closely spaced scatterers, RELAX usually will revisit those unresolved scatterers when the assumed number of scatterers is quite large or close to the true number of scatterers. Hence, when the number of scatterers used for autofocus is much smaller than the true number of scatterers in a large scene, using CLEAN and RELAX yields similar results. The smaller the number of scatterers we use, the faster the autofocus algorithm.

For some applications, we may wish to achieve simultaneous autofocus and super resolution SAR image formation of a small scene or small ROI in a large scene. For this case, we will extract the target features with RELAX rather than CLEAN. Also, instead of using the *ad hoc* criterion of checking the phase error estimates to determine the inner and outer convergences, we check the NLS fitting criterion in (7.17) to test practical convergences [19]. This is because the goal of the relaxation based methods is to minimize the NLS fitting criterion in (7.17). The algorithm we present below is referred to as MCRELAX1, whose

structure resembles MCCLEAN rather than the original MCRELAX algorithm presented in [17]. The steps of MCRELAX1 are as follows:

Step 0: Let $\mathbf{Z} = \mathbf{Y}$.

Step 1: Assume $\tilde{K} = 1$.

Substep (a): Obtain $\left\{\hat{f}_k, \hat{\tilde{f}}_k, \hat{\alpha}_k\right\}_{k=1}$ via RELAX by assuming $K = 1$.

Substep (b): Estimate the phase error $\hat{\mathbf{\Psi}}$ and then let $\mathbf{Z} = \mathbf{Y}\mathbf{P}^{-1}(\hat{\mathbf{\Psi}})$.

Substep (c): Re-estimate $\left\{\hat{f}_k, \hat{\tilde{f}}_k, \hat{\alpha}_k\right\}_{k=1}$ via RELAX by assuming $K = \tilde{K}$.

Substep (d): If the practical convergence is achieved (to be discussed later on), then go to the next step; otherwise, go to Substep 1(b).

Step 2: Assume $\tilde{K} = 2$.

Substep (a): Obtain $\left\{\hat{f}_k, \hat{\tilde{f}}_k, \hat{\alpha}_k\right\}_{k=2}$ by using the first substep of Step (2) of RELAX.

Substep (b): Re-estimate the phase error $\hat{\mathbf{\Psi}}$ and then let $\mathbf{Z} = \mathbf{Y}\mathbf{P}^{-1}(\hat{\mathbf{\Psi}})$.

Substep (c): Re-estimate $\left\{\hat{f}_k, \hat{\tilde{f}}_k, \hat{\alpha}_k\right\}_{k=1}^{2}$ via RELAX by assuming $K = \tilde{K}$.

Substep (d): If the practical convergence is achieved (to be discussed later on), then go to the next step; otherwise, go to Substep 2(b).

Step 3: Assume $\tilde{K} = 3$.

Substep (a): Obtain $\left\{\hat{f}_k, \hat{\tilde{f}}_k, \hat{\alpha}_k\right\}_{k=3}$ by using the first substep of Step (3) of RELAX.

Substep (b): Re-estimate the phase error $\hat{\mathbf{\Psi}}$ and then let $\mathbf{Z} = \mathbf{Y}\mathbf{P}^{-1}(\hat{\mathbf{\Psi}})$.

Substep (c): Re-estimate $\left\{\hat{f}_k, \hat{\tilde{f}}_k, \hat{\alpha}_k\right\}_{k=1}^{3}$ via RELAX by assuming $K = \tilde{K}$.

Substep (d): If the practical convergence is achieved (to be discussed later on), then go to the next step; otherwise, go to Substep 3(b).

Continuing Steps: Continue similarly until $\tilde{K} = \hat{K}$, where $\hat{K}$ is the estimate of K and may be estimated with the following generalized Akaike information criterion (GAIC) (see [19]).

Final Step: Use the super resolution target feature estimates to extroplate the target signal [21] and then apply FFT to form super resolution SAR image.

The practical convergences of RELAX and MCRELAX1 may be determined by checking whether the relative changes of the cost function $C_3\left(\left\{\hat{f}_k, \hat{\tilde{f}}_k, \hat{\alpha}_k\right\}_{k=1}^{K}\right)$ in (7.21) and the

cost function $C_1\left(\left\{\hat{\alpha}_k, \hat{f}_k, \hat{\bar{f}}_k\right\}_{k=1}^{\tilde{K}}, \hat{\mathbf{\Psi}}\right)$ in (7.17) (where K is replaced by $\tilde{K}$), respectively, between two consecutive iterations are less than some threshold (say, 10^{-3}).

We consider using MCRELAX1 with the GAIC (see [19] and the references therein for details) to determine K, the number of target scatterers, by assuming the unknown noise and clutter being white. The estimate $\hat{K}$ of K is determined as an integer that minimizes the following GAIC cost function:

$$\mathrm{GAIC}_{\breve{K}} = M\bar{M}\ln\left(\sum_{m=0}^{M-1}\sum_{\bar{m}=0}^{\bar{M}-1}|\hat{e}(m,\bar{m})|^2\right) + \gamma\ln[\ln(M\bar{M})](4\breve{K}+\bar{M}+1), \tag{7.29}$$

where $\hat{e}(m,\bar{m})$ is determined by

$$\hat{e}(m,\bar{m}) = y(m,\bar{m}) - \sum_{k=1}^{\breve{K}}\hat{\alpha}_k \exp\left[j2\pi(m\hat{f}_k + \bar{m}\hat{\bar{f}}_k)\right]\exp(j\hat{\psi}_{\bar{m}}), \tag{7.30}$$

$4\breve{K}+\bar{M}+1$ denotes the total number of unknown real-valued parameters (of which $4\breve{K}$ are for the target features, $\bar{M}$ are for the phase errors, and 1 is for the white noise), and γ is a parameter of user choice.

We remark that, although MCRELAX1 looks more complicated than MCRELAX, the former has better convergence property than the latter since the former does not depend on other methods to generate initial conditions.

7.5 Experimental Examples

In this section, we present two experimental examples to illustrate the performance of the proposed MCCLEAN and MCRELAX1 algorithms. The imaged scene is the Michigan Stadium and the experimental data is a portion of the data collected by one of the two apertures of the ERIM's (Environmental Research Institute of Michigan's) DCS IFSAR (interferometric SAR). These data have already been motion compensated by some unknown means and phase errors are artificially added to them so as to test the autofocus performance of the proposed algorithms. All of the windowed FFT images given below are obtained by applying 256×256 point 2-D FFT with Kaiser window and shape parameter 6 to the original or compensated phase history data.

First, we show the autofocus capability of MCCLEAN for a large scene. The original phase history data $\mathbf{Y}$ is a 256×256 matrix (i.e., $M = \bar{M} = 256$). The added phase error distribution is $\psi_{\bar{m}} = \frac{10\pi \times 10^6}{\bar{M}} \left(\bar{m} - \frac{\bar{M}}{2}\right)^2 + \frac{\pi}{3} W(\bar{m})$, where $\bar{m} = 0, 1, \cdots, \bar{M} - 1$ and $W(\bar{m})$ is uniformly distributed on the interval $[-1 \quad 1]$. Note that the phase errors are composed of a low-frequency content (quadratic) and a high-frequency content (white). The original and corrupted windowed FFT images are shown in Figures 7.3(a) and (b), respectively. The windowed FFT image after autofocus via MCCLEAN is shown in Figure 7.3(c), which is very close to the original image shown in Figure 7.3(a). For the MCCLEAN algorithm, we have used $\epsilon_{\tilde{K}} = 0.1/\tilde{K}$ and $\epsilon_o = 0.01$ to test the inner and outer convergences, respectively, and the number of scatterers used for estimating the phase errors is 46.

Next, we show the simultaneous autofocus and super resolution imaging capability of MCRELAX1. A small region of interest was cut out from the top left corner of Figure 7.3(c), which has already been focused by using MCCLEAN. The ROI is 2-D inverse Fourier transformed and then dewindowed to generate the corresponding phase history data. The phase history data $\mathbf{Y}$ of the ROI is now a 40×40 matrix (i.e., $M = \bar{M} = 40$). For the purpose of comparison, a corresponding ROI was also cut out from Figure 7.3(a) to serve as the original image to be compared with. The original unwindowed and windowed FFT images and the corresponding super resolution image obtained via RELAX [21] are shown in Figures 7.4(a), (b), and (c), respectively. Figure 7.4(d) shows the focused windowed FFT image of the ROI from Figure 7.3(c). The super resolution images corresponding to Figure 7.4(d) obtained via RELAX and MCRELAX1 are shown in Figures 7.4(e) and (f), respectively. Note that both Figures 7.4(e) and (f) have higher resolution than the image shown in Figure 7.4(d) and Figure 7.4(f) is closer to the original super resolution image shown in Figure 7.4(c) than Figure 7.4(e). In this example, we have used the GAIC criterion [19] with $\gamma = 4$ to estimate the number of scatterers K of this ROI, which yields $\hat{K} = 59$, and for the super resolution image formation, we have used 2.0 for the data extroplation factor [21].

We have also tried the well-known PGA algorithm for the above two examples and found that its performance is very sensitive to the choice of the threshold needed by the automatic

windowing step or the size reduction rate needed by the progressive windowing scheme. We must also make a choice between automatic windowing and progressive windowing. When appropriate windowing scheme and proper parameters are selected, PGA can also be used to generate very good focused SAR images.

7.6 Conclusions

In this chapter, a parametric algorithm, referred to as MCCLEAN, is proposed for the autofocus of SAR image of a large scene, which can handle *arbitrary* phase errors. It is highly automatic and no separate initialization step is needed. We also present a similarly structured algorithm for the simultaneous autofocus and super resolution image formation of a small scene or small region of interest in a large scene. Experimental examples have been used to demonstrate the effectiveness of the proposed algorithms.

Acknowledgments

The authors thank Maj. T. Burns for providing us with the ERIM data.

Reference

[1] C. A. Wiley, "Synthetic aperture radars," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-21, pp. 440–443, May 1985.

[2] R. Klemm, "Current trends in SAR technology," *IEEE Aerospace and Electronic Systems Magazine*, vol. 12, no. 3, pp. 3–8, March 1997.

[3] D. A. Ausherman, A. Kozma, J. L. Walker, H. M. Jones, and E. C. Poggio, "Developments in radar imaging," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-20, no. 4, pp. 363–399, July 1984.

[4] C. V. Jakowatz, Jr., D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. A. Thompson, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach.* Norwell, MA: Kluwer Academic Publishers, 1996.

[5] L. A. Porcello, "Turbulence-induced phase errors in synthetic aperture radar," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 6, pp. 636–644, 1970.

[6] J. M. Swiger, "A digital autofocus method for synthetic array spotlight radar using map drift," *Hughes Aircraft Company Invention Disclosure, 1976.*

[7] C. E. Mancill and J. M. Swiger, "A map drift autofocus technique for correlating high order SAR phase errors (U)," *The 27th Annual Tri-Service Radar Symposium Record*, pp. 391–400, Monterey, CA, June 1981.

[8] W. D. Brown and D. C. Ghiglia, "Some methods for reducing propagation-induced phase errors in coherent imaging systems-I. Formalism," *Journal of the Optical Society of America*, vol. 5, pp. 924–941, June 1988.

[9] T. M. Calloway and G. W. Donohoe, "Subaperture autofocus for synthetic aperture radar," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, pp. 617–621, April 1994.

[10] G. N. Yoji, "Phase difference autofocusing for synthetic aperture radar imaging," *United States Patent No. 4,999,635*, March 12, 1991.

[11] P. H. Eichel, D. C. Ghiglia, and C. V. Jakowatz, Jr., "Speckle processing methods for synthetic-aperture -radar phase correction," *Optics Letters*, vol. 14, pp. 1–3, January 1989.

[12] P. H. Eichel and C. V. Jakowatz, Jr., "Phase-gradient algorithm as an optimal estimator of the phase derivative," *Optics Letters*, vol. 14, pp. 1101–1103, October 1989.

[13] C. V. Jakowatz, Jr. and D. E. Wahl, "Eigenvector method for maximum-likelihood estimation of phase errors in synthetic-aperture-radar imagery," *Journal of the Optical Society of America*, vol. 10, pp. 2539–2546, December 1993.

[14] D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and C. V. Jakowatz, Jr., "Phase gradient autofocus-a robust tool for high resolution SAR phase correction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, pp. 827–835, July 1994.

[15] S. Werness, W. Carrara, L. Joyce, and D. Franczak, "Moving target imaging algorithm for SAR data," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 1, pp. 57–67, January 1994.

[16] S. Barbarossa and A. Scaglione, "Autofocusing of SAR images based on the product high order ambiguity function," to appear in *IEE Proceedings-Radar, Sonar, and Navigation: Special issue on Spectral Analysis Applications in Radar.*

[17] Z.-S. Liu and J. Li, "SAR motion compensation and feature extraction via a relaxation based method," *Journal of the Optical Society of America A*, vol. 15, pp. 599–610, March 1998.

[18] W. G. Carrara, R. S. Goodman, and R. M. Majewski, *Spotlight Synthetic Aperture Radar: Signal Processing Algorithms.* Norwood, MA, USA: Artech House, Inc., 1995.

[19] J. Li and P. Stoica, "Efficient mixed-spectrum estimation with applications to target feature extraction," *IEEE Transactions on Signal Processing*, vol. 44, pp. 281–295, February 1996.

[20] J. A. Högbom, "Aperture synthesis with a non-regular distribution of interferometer baselines," *Astronomy and Astrophysics Supplements*, vol. 15, pp. 417–426, 1974.

[21] Z. Bi, J. Li, and Z.-S. Liu, "Super resolution SAR image formation via parametric spectral estimation methods," *to appear in IEEE Transactions on Aerospace and Electronic Systems*, January 1999.

[22] D. C. Munson, Jr., J. D. O'Brien, and W. K. Jenkins, "A tomographic formulation of spotlight-mode synthetic aperture radar," *Proceedings of the IEEE*, vol. 71, pp. 917 –925, August 1983.

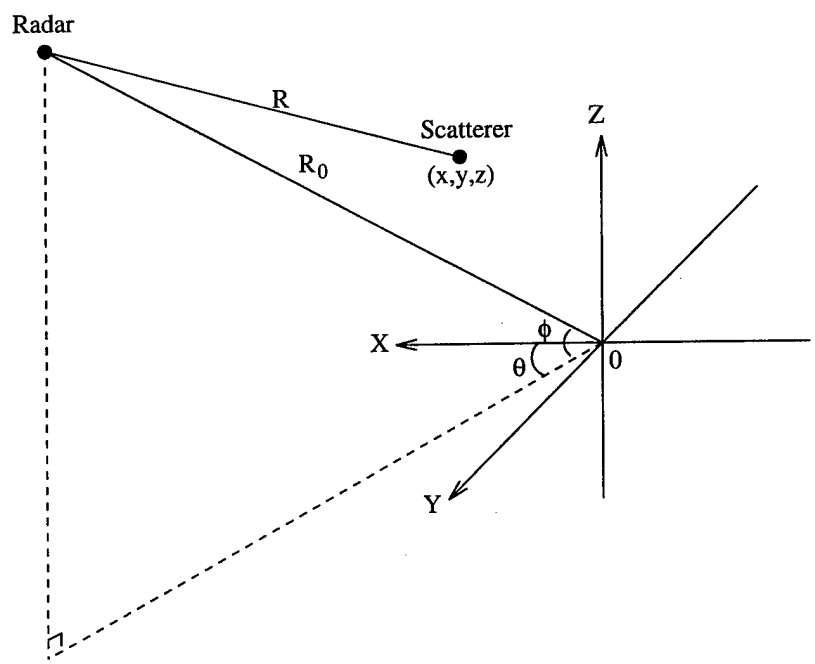


Figure 7.1: Data collection geometry in a spotlight-mode SAR.

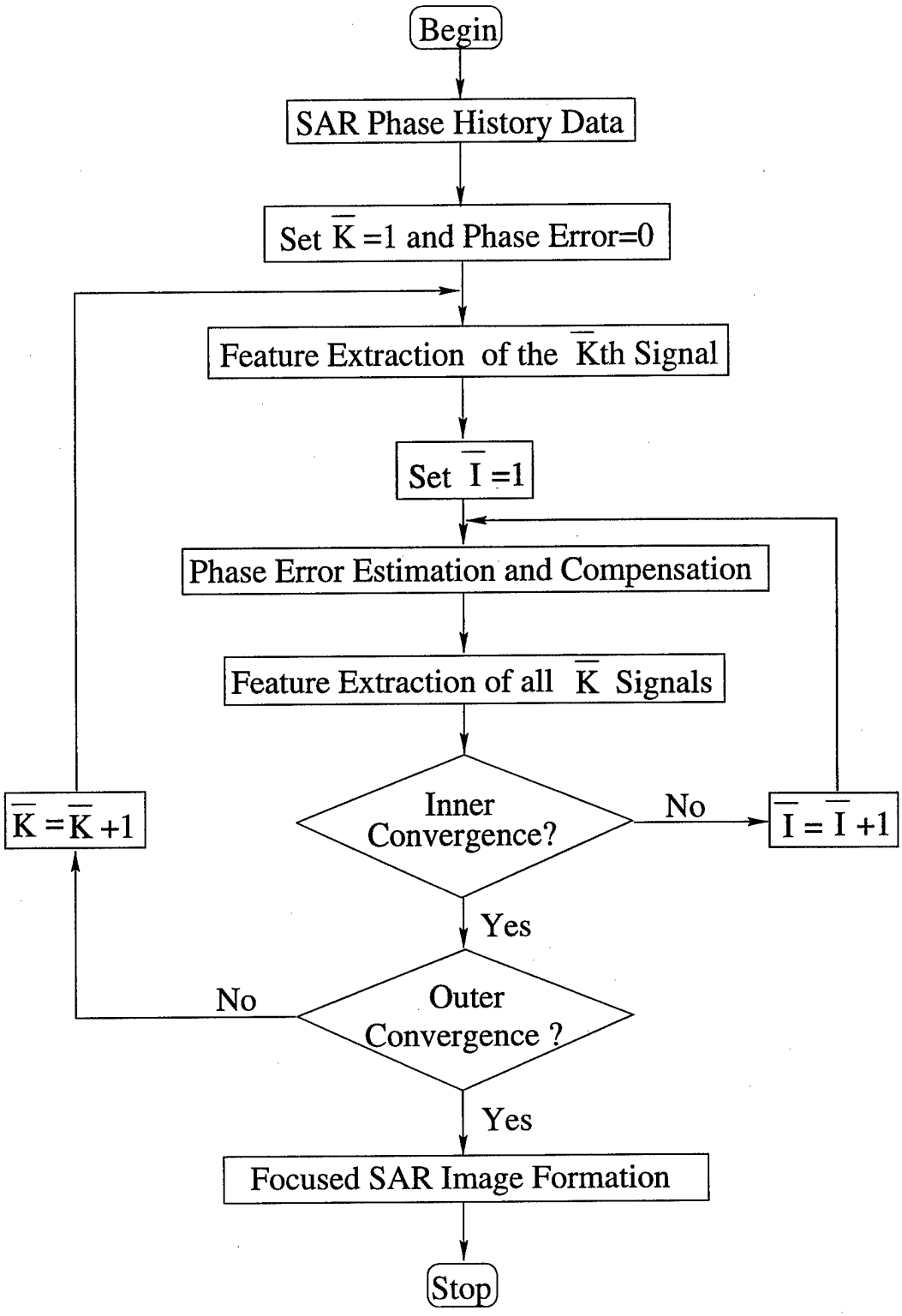
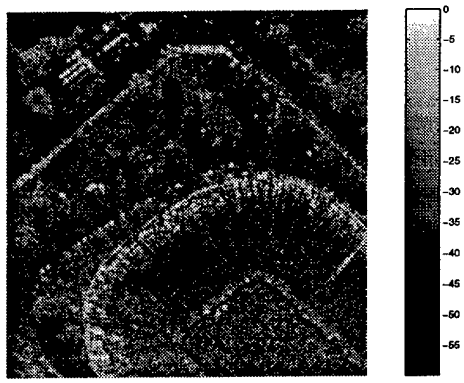
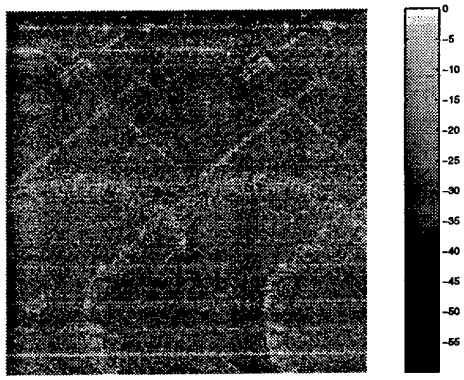


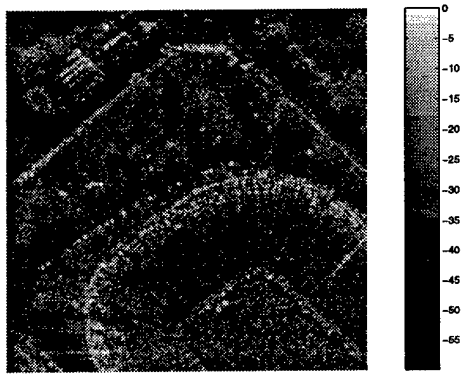
Figure 7.2: Flow chart of the MCCLEAN algorithm.



(a)



(b)



(c)

Figure 7.3: SAR image of the Michigan Stadium. (a) Original windowed FFT image. (b) Image corrupted by phase errors. (c) The windowed FFT image after autofocus via MC-CLEAN.

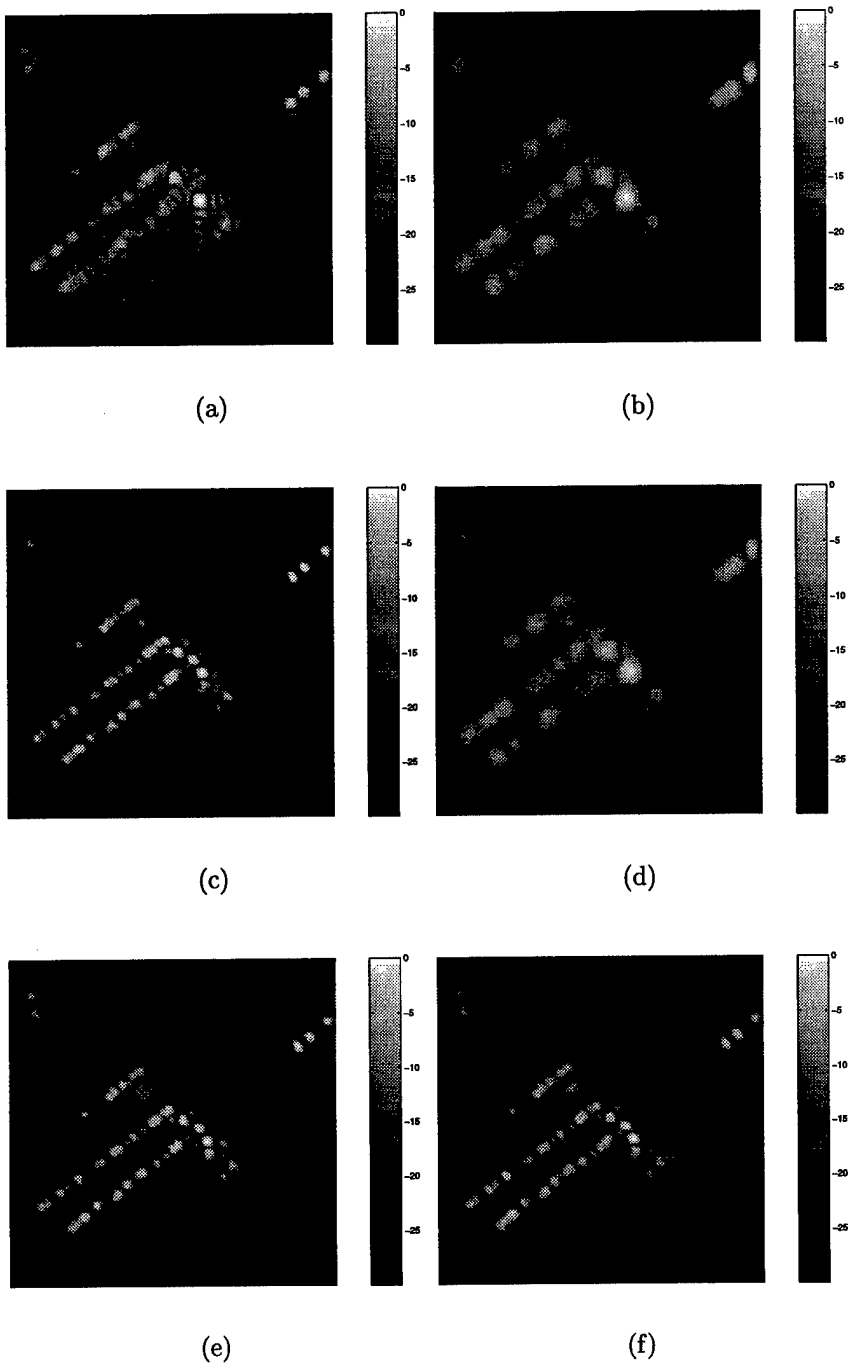


Figure 7.4: SAR image of a small region of interest out of the Michigan Stadium data. (a) Original unwindowed FFT image. (b) Original windowed FFT image. (c) Original super resolution image obtained via RELAX. (d) Windowed FFT image after autofocus via MCCLEAN (for large scene autofocus). (e) Super resolution image obtained via MCCLEAN (for large scene autofocus) plus RELAX. (f) Super resolution image obtained via MCCLEAN (for large scene autofocus) plus MCRELAX1.

8. Super Resolution SAR Imaging via Parametric Spectral Estimation Methods

8.1 Introduction

The conventional fast Fourier transform (FFT) method is a nonparametric spectral estimation approach and is robust and computationally very efficient for synthetic aperture radar (SAR) image formation. However, the FFT method generates SAR images with high sidelobes and poor resolutions due to the limited phase history data collected from a finite length synthetic aperture with a finite bandwidth radar. To reduce the sidelobes, different window functions can be applied to the SAR phase history data before FFT processing. Yet this is achieved at the cost of worsening the resolution. In [1, 2], many nonparametric and parametric spectral estimation methods are compared and discussed for their merits for SAR image formation. The nonparametric methods that have been used for SAR image formation and target feature extraction include, for example, reduced-rank variations of the Capon method [1, 2, 3, 4], the adaptive sidelobe reduction approaches [5], and the matched-filterbank based complex spectral estimation methods [6] including the Capon [7] and APES [8] methods. Yet the resolution of these nonparametric methods is not significantly better than that of the FFT based methods due to their nonparametric nature.

Parametric spectral estimation algorithms, which can be attractive alternatives to nonparametric methods, have been used extensively for SAR target feature extraction. The parametric methods that have been considered include, for example, autoregressive (AR) model based methods [1, 2, 9, 10], eigendecomposition based methods [1, 2, 3, 4, 11] including MUSIC [12] and ESPRIT [13], and nonlinear least squares fitting based methods [1, 2, 14, 15]. The parametric spectral estimation methods are devised based on certain parametric data models, mostly on sinusoidal data models to model ideal point scatterers. Robust parametric methods offer the promise of significantly improving the resolution and accuracy of the FFT methods. Since SAR images rather than target features are often used in SAR applications, we consider herein first extracting the target features from the SAR phase history data with the parametric methods and then forming SAR images by applying

the FFT methods to the simulated phase history data matrices of large dimensions. The simulated data matrices are based on the extracted features and the assumed data models. The dominant target features of the so-obtained SAR images can have a much better resolution than those obtained with the FFT based methods. Since the parametric data models are usually approximate, parametric methods robust against model errors should be used to extract target features.

In this chapter, we extend robust and high resolution relaxation-based parametric spectral estimation algorithms RELAX [14] and RELAX-NLS [15], which have been used effectively for SAR target feature extraction [14, 15], to SAR image formation and evaluate their performances with experimental data including the MSTAR and ERIM data. The RELAX algorithm assumes that SAR targets consist of only point scatterers, i.e., trihedral corner reflectors. The RELAX-NLS algorithm assumes that the SAR targets consist of both trihedral and dihedral corner reflectors. Although sinusoidal point scatterer data model tends to work well in range, it is more difficult to establish an approximate parametric data model in cross-range. The RELAX algorithm assumes a sinusoidal data model in cross-range as well while the RELAX-NLS algorithm assumes a data model consisting of both sinusoids and sinc $(\sin(x)/x)$ functions in cross-range. These models work well for many man-made targets. We will show with experimental examples that compared with the FFT methods, these more sophisticated parametric spectral estimation methods can provide SAR images with higher resolution even though the data models used by the parametric methods are only approximately correct.

The remainder of this chapter is organized as follows. Section 8.2 describes how to form SAR images via RELAX and RELAX-NLS. Section 8.3 presents experimental examples showing their SAR image formation performances. Finally, Section 8.4 contains our conclusions.

8.2 SAR Image Formation via Spectral Estimation Methods

We consider below forming super resolution SAR images using the robust relaxation based optimization algorithms including RELAX and RELAX-NLS.

8.2.1 RELAX

When the dominant features of a target, such as many man-made targets, can be approximated as point scatterers, the RELAX algorithm [14] can be used to form the SAR images and improve the resolutions of the dominant target features. Assume that there are K dominant scatterers in a target. Then the parametric data model used by RELAX has the form [16]:

$$y(n,\bar{n}) = \sum_{k=1}^{K} \alpha_k e^{j(\omega_k n + \bar{\omega}_k \bar{n})} + e(n,\bar{n}), \quad n = 0,1,\cdots,N-1, \quad \bar{n} = 0,1,\cdots,\bar{N}-1, \tag{8.1}$$

where N and $\bar{N}$, respectively, denote the numbers of the available data samples in range and cross-range; $\{\alpha_k\}_{k=1}^{K}$ and $\{\omega_k, \bar{\omega}_k\}_{k=1}^{K}$, respectively, denote the unknown complex amplitudes and 2-D unknown frequencies of the K sinusoids or point scatterers; finally, $e(n,\bar{n})$ denotes the unknown noise and clutter. The sinusoidal frequencies ω_k and $\bar{\omega}_k$ correspond to the 2-D location of the kth scatterer of a radar target; α_k is determined by its radar cross section (RCS). It has been shown in [14] that RELAX can be implemented with a sequence of Fourier transforms and is robust against errors in the data model and the assumed number of scatterers due to its simplicity. The SAR images obtained via RELAX are referred to as *RELAX SAR images* or simply *RELAX images.* The steps of using RELAX for SAR image formation are given below.

Step 1: Obtain the estimates of $\{\alpha_k, \omega_k, \bar{\omega}_k\}_{k=1}^{K}$ via RELAX by using the measured phase history data. See [14] for details.

Step 2: From the estimated parameters and based on the data model in (8.1), generate the simulated phase history data of large dimensions:

$$y_s(n_s,\bar{n}_s) = \sum_{k=1}^{K} \hat{\alpha}_k e^{j(\hat{\omega}_k n_s + \hat{\bar{\omega}}_k \bar{n}_s)}, \quad n_s = 0,1,\cdots,\beta N-1, \quad \bar{n}_s = 0,1,\cdots,\beta\bar{N}-1, \tag{8.2}$$

where β ($\beta \geq 1$) denotes an extrapolation factor and $\hat{\alpha}_k$, $\hat{\omega}_k$, and $\hat{\bar{\omega}}_k$, respectively, denote the estimates of α_k, ω_k, and $\bar{\omega}_k$, $k = 1, 2, \cdots, K$.

Step 3: Form RELAX SAR images containing only the dominant target features by applying the normalized FFT to the simulated phase history data $\{y_s(n_s, \bar{n}_s)\}$, i.e., by computing

$$\frac{1}{\beta^2 N\bar{N}} \sum_{n_s=0}^{\beta N-1} \sum_{\bar{n}_s=0}^{\beta \bar{N}-1} y_s(n_s, \bar{n}_s) e^{-j(\omega n_s + \bar{\omega} \bar{n}_s)}. \tag{8.3}$$

If we wish to suppress the sidelobes, we apply the normalized FFT to the windowed sequence $\{w_s(n_s, \bar{n}_s) y_s(n_s, \bar{n}_s)\}$, where the window sequence $w_s(n_s, \bar{n}_s)$ satisfies

$$\sum_{n_s=0}^{\beta N-1} \sum_{\bar{n}_s=0}^{\beta \bar{N}-1} = \beta^2 N\bar{N}. \tag{8.4}$$

We may also wish to form RELAX SAR images containing both the dominant target features and background clutter since, for example, the shadow information may be desired for automatic target recognition. If so, we apply the normalized FFT to the sum of the simulated phase history data $\{y_s(n_s, \bar{n}_s)\}$ and $\{\beta^2 \hat{e}(n, \bar{n})\}$ with zero padding to have dimensions βN and $\beta \bar{N}$, where $\hat{e}(n, \bar{n})$ denotes the estimated background clutter and is determined by

$$\hat{e}(n, \bar{n}) = y(n, \bar{n}) - \sum_{k=1}^{K} \hat{\alpha}_k e^{j(\hat{\omega}_k n + \hat{\bar{\omega}}_k \bar{n})}, \quad n = 0, 1, \cdots, N-1, \quad \bar{n} = 0, 1, \cdots, \bar{N}-1. \tag{8.5}$$

Note that scaling the $\hat{e}(n, \bar{n})$ by a factor of β^2 is needed when the background clutter is included in the RELAX SAR images since both of its dimensions are $1/\beta$ times of those of the simulated phase history data. If we wish to suppress the sidelobes, we apply the normalized FFT to the sum of $\{w_s(n_s, \bar{n}_s) y_s(n_s, \bar{n}_s)\}$ and $\{\beta^2 w_e(n, \bar{n}) \hat{e}(n, \bar{n})\}$ with zero padding to have dimensions βN and $\beta \bar{N}$, where the window sequence $w_e(n, \bar{n})$ satisfies

$$\sum_{n=0}^{N-1} \sum_{\bar{n}=0}^{\bar{N}-1} w_e(n, \bar{n}) = N\bar{N}. \tag{8.6}$$

Note that since we cannot model the clutter effectively, its resolution cannot be improved. We remark that β does not determine the resolution of the RELAX SAR images since the resolution is determined by RELAX. We need to choose $\beta > 1$ to demonstrate the super resolution property of the RELAX algorithm for target feature extraction and β is a parameter of user choice.

We consider using the generalized Akaike information criterion (GAIC) (see [14] and the references therein for details) to determine K, the number of sinusoids, by assuming the unknown noise and clutter being white. The estimate $\hat{K}$ of K is determined as an integer that minimizes the following GAIC cost function:

$$\mathrm{GAIC}_{\breve{K}} = N\bar{N}\ln\left(\sum_{n=0}^{N-1}\sum_{\bar{n}=0}^{\bar{N}-1}|\hat{e}(n,\bar{n})|^2\right) + \gamma\ln[\ln(N\bar{N})](4\breve{K}+1), \tag{8.7}$$

where $\hat{e}(n,\bar{n})$ is determined by (8.5) with K replaced by $\breve{K}$, $4\breve{K}+1$ denotes the total number of unknown real-valued parameters (of which $4\breve{K}$ are for the sinusoids and 1 is for the white noise), and γ is a parameter of user choice.

8.2.2 RELAX-NLS

The RELAX-NLS algorithm [15] can be used to form the SAR images of the dominant target features when the targets consist of both trihedral (point scatterers) and dihedral corner reflectors. In [15], a data model consisting of sinc functions in cross-range has been used to model dihedrals and has the form:

$$y_d(n,\bar{n}) = \sum_{k=1}^{K_d}\alpha_{d_k}\mathrm{sinc}[\pi b_k(\bar{n}-\tau_k)]e^{j(\omega_{d_k}n+\bar{\omega}_{d_k}\bar{n})}, \quad n=0,1,\cdots,N-1, \quad \bar{n}=0,1,\cdots,\bar{N}-1, \tag{8.8}$$

where N and $\bar{N}$, as before, denote the numbers of the available data samples in range and cross-range, respectively; α_{d_k}, $\{\omega_{d_k},\bar{\omega}_{d_k}\}$, and b_k, $k=1,2,\cdots,K_d$, are, respectively, proportional to the maximal RCS, the central location, and the length of the kth dihedral corner reflector; τ_k, $k=1,2,\cdots,K_d$, denotes the peak location of the data sequence in cross-range and is determined by the orientation of the kth dihedral corner reflector; finally, K_d is the number of the dihedral corners. Assume that there are K_d dihedral corner reflectors and K_t trihedral corner reflectors in a target and $K = K_d + K_t$. Then the parametric data model used to describe both trihedrals and dihedrals in the presence of noise and clutter has the form [15]:

$$y(n,\bar{n}) = y_d(n,\bar{n}) + y_t(n,\bar{n}) + e(n,\bar{n}), \quad n=0,1,\cdots,N-1, \quad \bar{n}=0,1,\cdots,\bar{N}-1, \tag{8.9}$$

where

$$y_t(n,\bar{n}) = \sum_{k=1}^{K_t} \alpha_{t_k} e^{j(\omega_{t_k} n + \bar{\omega}_{t_k} \bar{n})}, \quad n = 0, 1, \cdots, N-1, \quad \bar{n} = 0, 1, \cdots, \bar{N}-1, \tag{8.10}$$

with $\{\alpha_{t_k}\}_{k=1}^{K_t}$, $\{\omega_{t_k}, \bar{\omega}_{t_k}\}_{k=1}^{K_t}$, respectively, denoting the unknown complex amplitudes and 2-D unknown frequencies of the K_t trihedral corner reflectors. The RELAX-NLS algorithm proposed in [15] can be used to identify the types of the corner reflectors in a target and effectively estimate the target features

$\{\alpha_{t_k}, \omega_{t_k}, \bar{\omega}_{t_k}\}_{k=1}^{K_t}$ and $\{\alpha_{d_k}, b_k, \omega_{d_k}, \bar{\omega}_{d_k}, \tau_k\}_{k=1}^{K_d}$ by utilizing an alternating optimization method [17] to minimize a nonlinear least squares cost function. Like other alternating optimization algorithms [18], the RELAX-NLS is not guaranteed to converge to the global minimum, while it is guaranteed to converge to at least a local minimum under mild conditions. The SAR images obtained via RELAX-NLS are referred to as *RELAX-NLS SAR images* or simply *RELAX-NLS images.* To form the RELAX-NLS SAR images, the dominant target features are first extracted by using the RELAX-NLS algorithm (see [15] for details) and then Steps 2 and 3 discussed in Section 2.1 are used except that the simulated phase history data of large dimensions used in Step 2 is now determined by

$$y_s(n_s,\bar{n}_s) = \beta \sum_{k=1}^{K_d} \hat{\alpha}_{d_k} \mathrm{sinc}[\pi \hat{b}_k(\bar{n}_s - \hat{\tau}_k)] e^{j(\hat{\omega}_{d_k} n_s + \hat{\bar{\omega}}_{d_k} \bar{n}_s)} + \sum_{k=1}^{K_t} \hat{\alpha}_{t_k} e^{j(\hat{\omega}_{t_k} n_s + \hat{\bar{\omega}}_{t_k} \bar{n}_s)}, \tag{8.11}$$

where $n_s = 0, 1, \cdots, \beta N - 1$ and $\bar{n}_s = 0, 1, \cdots, \beta \bar{N} - 1$ with β being an extrapolation factor ($\beta \geq 1$); $\hat{\alpha}_{d_k}$, $\hat{\omega}_{d_k}$, $\hat{\bar{\omega}}_{d_k}$, $\hat{b}_k$, and $\hat{\tau}_k$, respectively, denote the estimates of α_{d_k}, ω_{d_k}, $\bar{\omega}_{d_k}$, b_k, and τ_k, $k = 1, 2, \cdots, K_d$; finally, $\hat{\alpha}_{t_k}$, $\hat{\omega}_{t_k}$, and $\hat{\bar{\omega}}_{t_k}$, respectively, denote the estimates of α_{t_k}, ω_{t_k}, and $\bar{\omega}_{t_k}$, $k = 1, 2, \cdots, K_t$. The clutter estimate $\hat{e}(n,\bar{n})$ used in Step 3 is now determined by

$$\hat{e}(n,\bar{n}) = y(n,\bar{n}) - \sum_{k=1}^{K_d} \hat{\alpha}_{d_k} \mathrm{sinc}[\pi \hat{b}_k(\bar{n} - \hat{\tau}_k)] e^{j(\hat{\omega}_{d_k} n + \hat{\bar{\omega}}_{d_k} \bar{n})} - \sum_{k=1}^{K_t} \hat{\alpha}_{t_k} e^{j(\hat{\omega}_{t_k} n + \hat{\bar{\omega}}_{t_k} \bar{n})}, \tag{8.12}$$

with $n = 0, 1, \cdots, N-1$ and $\bar{n} = 0, 1, \cdots, \bar{N}-1$. Note that scaling the simulated data for the dihedral corner reflectors by a factor of β in (8.11) is needed since the sinc function goes to zero as n_s increases or decreases away from τ_k. Note also that $\beta > 1$ is used to demonstrate the super resolution property of the RELAX-NLS algorithm for target feature extraction and β is a parameter of user choice. Finally, since the Fourier transforms of the sinc functions of

sufficient lengths do not result in sidelobes, when $y_s(n_s, \bar{n}_s)$ in (8.11) is windowed, the second term of $y_s(n_s, \bar{n}_s)$ is multiplied by $w_s(n_s, \bar{n}_s)$ and the first term is multiplied by $w_1(n_s)$, where $w_1(n_s)$ is a 1-D window sequence satisfying

$$\sum_{n_s=0}^{\beta N-1} = \beta N. \tag{8.13}$$

We can also determine K, the total number of trihedral and dihedral corners, by extending the GAIC discussed in Section 2.1 and assuming white noise and clutter. The estimate $\hat{K}$ of K is determined as an integer that minimizes the following extended GAIC cost function:

$$\text{GAIC}_{\check{K}} = N\bar{N}\ln\left(\sum_{n=0}^{N-1}\sum_{\bar{n}=0}^{\bar{N}-1}|\hat{e}(n,\bar{n})|^2\right) + \gamma\ln[\ln(N\bar{N})](4\check{K}_t + 6\check{K}_d + 1), \tag{8.14}$$

where γ is a parameter of user choice; $\check{K} = \check{K}_t + \check{K}_d$ with $\check{K}_t$ and $\check{K}_d$ denoting the numbers of trihedral and dihedral corners, respectively, determined by RELAX-NLS given $\check{K}$; $\hat{e}(n, \bar{n})$ is determined by (8.12) with K_t and K_d replaced by $\check{K}_t$ and $\check{K}_d$, respectively; finally, $4\check{K}_t + 6\check{K}_d + 1$ is the total number of unknown real-valued parameters (of which $4\check{K}_t$ and $6\check{K}_d$, respectively, are for the trihedral and dihedral corners and 1 is for the white noise).

8.3 Experimental Results

In this section, we demonstrate the SAR image formation performances of the RELAX and RELAX-NLS methods by using the experimental data. In the following examples, the extrapolation factor $\beta = 2$ is used and GAIC with $\gamma = 4$ is used to determine $\hat{K}$ for both RELAX and RELAX-NLS. Kaiser windows with shape parameter 6 are used to obtain windowed SAR images.

First consider SAR image formation via 2-D FFT and RELAX by using a portion of the 2-D data corresponding to some roof rims collected by one of the two apertures of the ERIM's (Environment Research Institute of Michigan's) DCS interferometric synthetic aperture radar (IFSAR). Figures 8.1(a) and (b), respectively, show the unwindowed and windowed 2-D FFT images obtained by zero-padding the 40×40 phase history data. GAIC gives $\hat{K} = 59$. Figure 8.2 shows the unwindowed and windowed RELAX images with and

without the background clutter and the corresponding clutter images with $\hat{K} = 59$. (The unwindowed and windowed clutter images are obtained by applying the normalized FFT to $\{\hat{e}(n,\bar{n})\}$ and $\{w_e(n,\bar{n})\hat{e}(n,\bar{n})\}$, respectively.) Note that the RELAX images have a higher resolution than the FFT images for the dominant scatterers. Utilizing only 25% of the 40×40 phase history data, i.e., using a 20×20 phase history data, we form the 2-D FFT SAR images shown in Figure 8.3. Figure 8.4 shows the RELAX images as well as the the clutter images with $\hat{K} = 41$ determined by the GAIC. Comparing Figures 8.4(f) and 8.1(b), we note that the two images are quite similar although the former uses only 25% of the data used by the latter.

Consider next two examples of MSTAR target chip image formation, where the field data was collected by the Sandia National Laboratory (SNL) using the STARLOS sensor. The data was collected by a spotlight-mode SAR with center frequency 9.6 GHz, bandwidth 0.591 GHz, elevation angle 15°, and range about 4.5 kilometers. Figures 8.5 and 8.6 show the photos of a tank taken from the azimuth angles 0° and 90°, respectively. Figures 8.7(a) and (b), respectively, show the unwindowed and windowed 2-D FFT SAR images of the target at 0° azimuth angle. We next form SAR images with and without background clutter by using the RELAX and RELAX-NLS methods. Figure 8.8 shows the SAR images, including the corresponding clutter images, of the target at 0° azimuth angle obtained via RELAX with $\hat{K} = 27$ determined via GAIC. For this example, RELAX-NLS and RELAX yield identical images since all reflectors are identified by RELAX-NLS as trihedrals. Note again that the SAR images obtained via RELAX and RELAX-NLS have a higher resolution than those obtained via FFT methods for the dominant target scatterers. Figures 8.9(a) and (b), respectively, show the unwindowed and windowed 2-D FFT SAR images of the target at 90° azimuth angle. Figure 8.10 shows the RELAX SAR images with and without background clutter and the corresponding clutter images with $\hat{K} = 37$ determined via GAIC. Figure 8.11 shows the RELAX-NLS SAR images with and without background clutter and the corresponding clutter images obtained with $\hat{K} = 36$ determined via GAIC. We note that although RELAX considers all reflectors as trihedrals, the RELAX images still resemble the FFT images, which shows the robustness of the RELAX algorithm.

Finally, we consider an example of the MSTAR Slicy data consisting of both trihedral and dihedral corner reflectors collected by the SNL using the STARLOS sensor. The field data was collected by a spotlight-mode SAR with a carrier frequency 9.559 GHz and bandwidth 0.591 GHz. The radar was about 5 kilometers away from the ground target shown in Figure 8.12. The SAR images are obtained when the target is illuminated by the radar from the azimuth angle 0° and elevation angle 30°. The unwindowed and windowed 2-D FFT images are shown in Figures 8.13(a) and (b), respectively. We have also applied RELAX and RELAX-NLS to this 32×32 phase history data matrix and determine $\hat{K}$ via GAIC. Figure 8.14 shows the unwindowed and windowed RELAX SAR images with and without background clutter and the corresponding clutter images with $\hat{K} = 36$ determined via GAIC. We note that RELAX images have a better resolution than the FFT images for the dominant trihedrals even though the data model in the cross-range dimension used by RELAX is not correct for this example. Figure 8.15 shows the unwindowed and windowed RELAX-NLS SAR images with and without background clutter and the corresponding clutter images with $\hat{K} = 24$ determined via GAIC. We note again that the RELAX-NLS images have a better resolution than the 2-D FFT images and the corner reflector types are mostly identified correctly.

8.4 Conclusions

In this chapter, we have demonstrated how to form super resolution SAR images via sophisticated parametric spectral estimation algorithms including RELAX and RELAX-NLS. Experimental examples have shown that the robust RELAX and RELAX-NLS algorithms offer significant advantages over the FFT methods to better resolve the dominant target scatterers.

Reference

[1] S. R. DeGraaf, "SAR imaging via modern 2-D spectral estimation methods," *SPIE Proceedings on Optical Engineering in Aerospace Sensing*, vol. 2230, pp. 36–47, Orlando, FL, April 1994.

[2] S. R. DeGraaf, "SAR imaging via modern 2-D spectral estimation methods," *IEEE Transactions on Image Processing*, vol. 7, pp. 729–761, May 1998.

[3] G. R. Benitz, "Adaptive high-definition imaging," *SPIE Proceedings on Optical Engineering in Aerospace Sensing,* Orlando, FL, pp. 106–119, April 1994.

[4] G. R. Benitz, "High definition vector imaging for synthetic aperture radar," *Proceedings of the 31st Asilomar Conference on Signals, Systems and Computers,* Pacific Grove, CA, November 1997.

[5] S. R. DeGraaf, "Sidelobe reduction via adaptive FIR filtering in SAR imagery," *IEEE Transactions on Image Processing*, vol. 3, pp. 292–301, May 1994.

[6] H. Li, J. Li, and P. Stoica, "Performance analysis of forward-backward matched-filterbank spectral estimators," *IEEE Transactions on Signal Processing*, vol. 46, pp. 1954–1966, July 1998.

[7] J. Capon, "High resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, pp. 1408–1418, August 1969.

[8] J. Li and P. Stoica, "An adaptive filtering approach to spectral estimation and SAR imaging," *IEEE Transactions on Signal Processing*, vol. 44, pp. 1469–1484, June 1996.

[9] A. Farina, F. Prodi, and F. Vinelli, "Application of superresolution techinques to radar imaging," *Chinese Journal of Systems Engineering and Electronics*, vol. 5, no. 1, pp. 1–14, January 1994.

[10] A. Farina, A. Forte, F. Prodi, and F. Vinelli, "Superresolution capabilities in 2D direct and inverse SAR processing," *International Symposium on Noise and Clutter Rejection in Radars and Imaging Sensors*, pp. 151–156, Kanagawa Science Park, Japan, November 1994.

[11] S. Barbarossa, L. Marsili, and G. Mungari, "SAR super-resolution imaging by signal subspace projection techniques," *AEU International Journal of Electronics and Communications*, vol. 50, no. 2, pp. 133–138, March 1996.

[12] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. AP-34, pp. 276–280, March 1986.

[13] R. Roy and T. Kailath, "ESPRIT – Estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-37, pp. 984–995, July 1989.

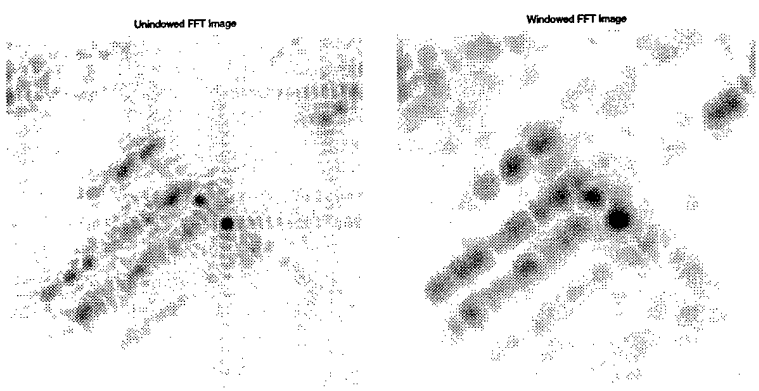
[14] J. Li and P. Stoica, "Efficient mixed-spectrum estimation with applications to target feature extraction," *IEEE Transactions on Signal Processing*, vol. 44, pp. 281–295, February 1996.

[15] Z.-S. Liu and J. Li, "Feature extraction of SAR targets consisting of trihedral and dihedral corner reflectors," *IEE Proceedings on Radar, Sonar and Navigation*, vol. 145, pp. 161–172, March 1998.

[16] C. V. Jakowatz, Jr., D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. A. Thompson, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach.* Norwell, MA: Kluwer Academic Publishers, 1996.

[17] B. D. Bunday, *Basic Optimization Methods.* Edward Arnold Ltd, London, 1984.

[18] W. I. Zangwill, *Nonlinear Programming: A Unified Approach.* Prentice-Hall, Inc., Englewood Cliffs, NJ 07632, 1969.



(a) (b)

Figure 8.1: SAR images obtained via 2-D FFT by using the 40×40 ERIM data. (a) Unwindowed 2-D FFT image. (b) Windowed 2-D FFT image.

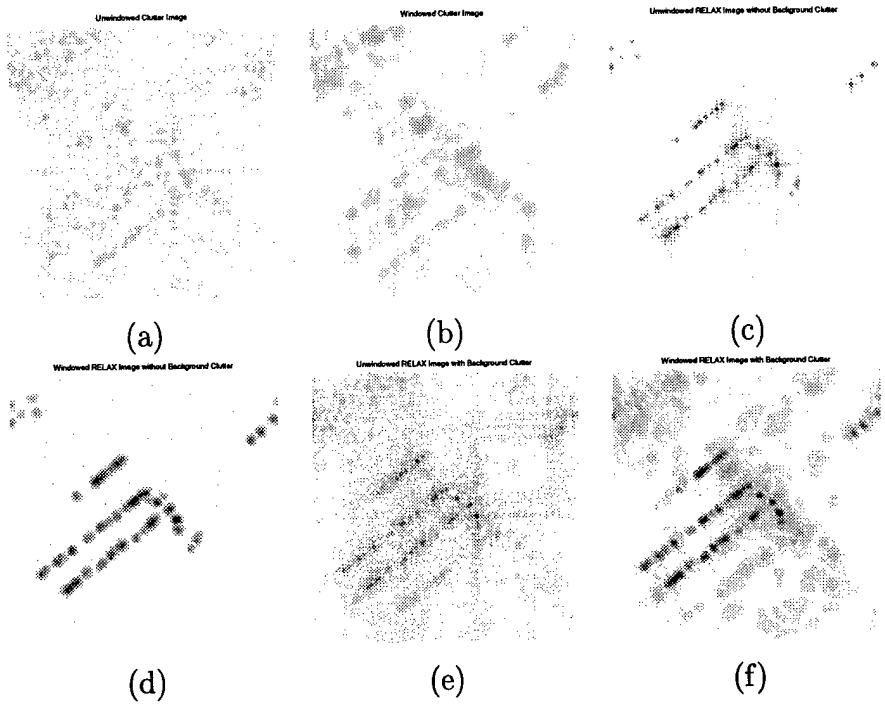


Figure 8.2: SAR images obtained via RELAX by using the 40×40 ERIM data with $\hat{K} = 59$. (a) Unwindowed clutter image. (b) Windowed clutter image. (c) Unwindowed RELAX image without background clutter. (d) Windowed RELAX image without background clutter. (e) Unwindowed RELAX image with background clutter. (f) Windowed RELAX image with background clutter.

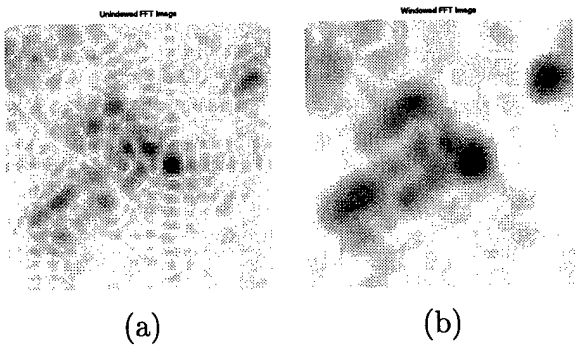


Figure 8.3: SAR images obtained via 2-D FFT by using the 20×20 ERIM data. (a) Unwindowed 2-D FFT image. (b) Windowed 2-D FFT image.

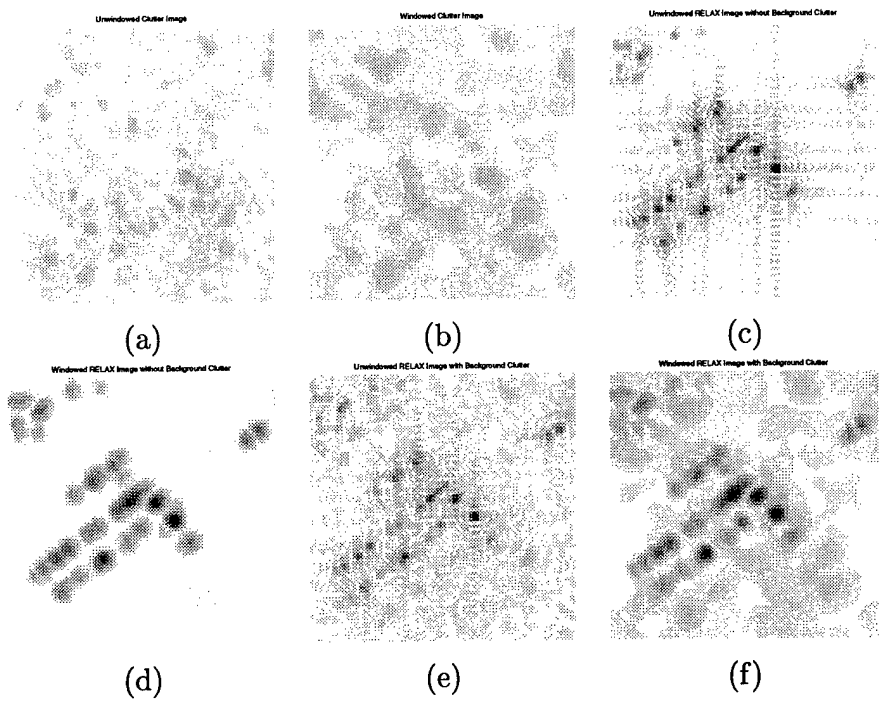


Figure 8.4: SAR images obtained via RELAX by using the 20×20 ERIM data with $\hat{K} = 41$. (a) Unwindowed clutter image. (b) Windowed clutter image. (c) Unwindowed RELAX image without background clutter. (d) Windowed RELAX image without background clutter. (e) Unwindowed RELAX image with background clutter. (f) Windowed RELAX image with background clutter.

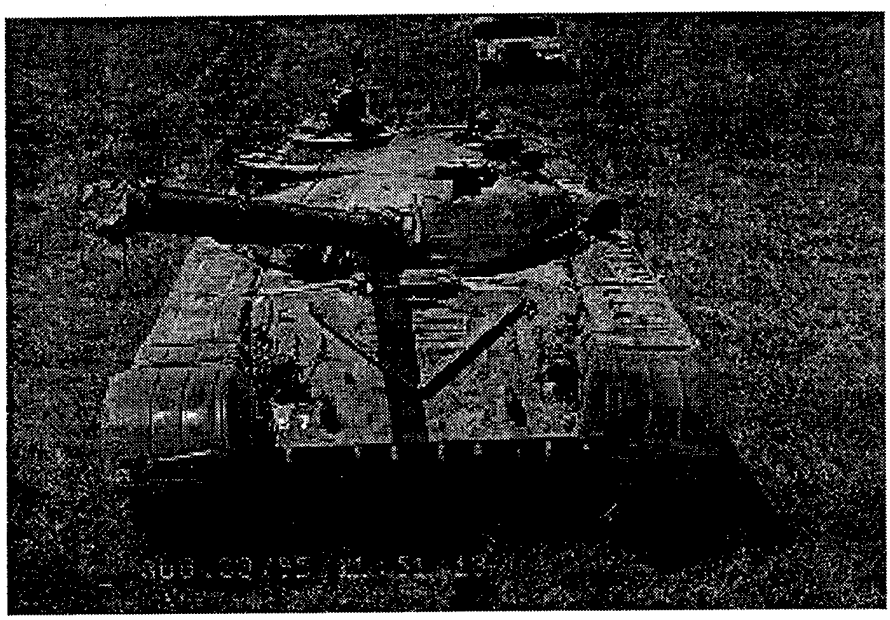
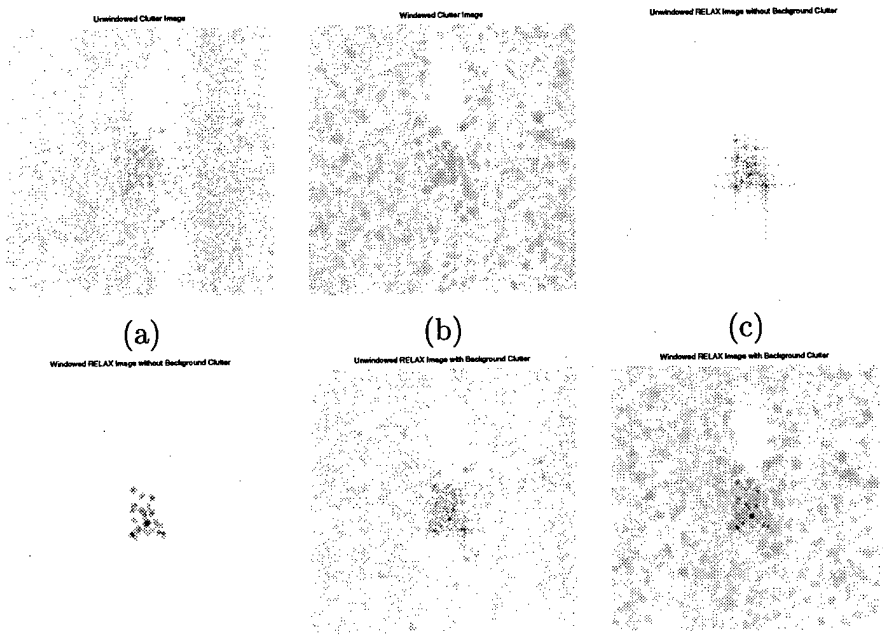


Figure 8.5: Tank photo taken at 0° azimuth angle.

Figure 8.6: Tank photo taken at 90° azimuth angle.

(a) (b)

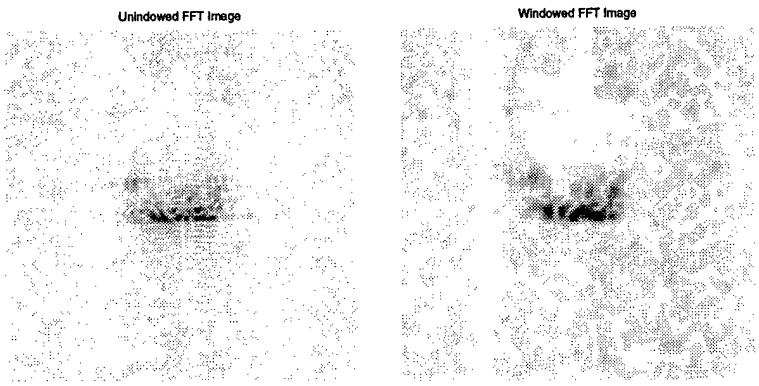
Figure 8.7: SAR images obtained via 2-D FFT from the MSTAR data hb03353.015 (0° azimuth and 15° elevation angles). (a) Unwindowed 2-D FFT image. (b) Windowed 2-D FFT image.



(a) (b) (c)

(d) (e) (f)

Figure 8.8: SAR images obtained via RELAX by using the MSTAR data hb03353.015 (0° azimuth and 15° elevation angles) with $\hat{K} = 27$. (a) Unwindowed clutter image. (b) Windowed clutter image. (c) Unwindowed RELAX image without background clutter. (d) Windowed RELAX image without background clutter. (e) Unwindowed RELAX image with background clutter. (f) Windowed RELAX image with background clutter.



(a) (b)

Figure 8.9: SAR images obtained via 2-D FFT from the MSTAR data hb03365.015 (90° azimuth and 15° elevation angles). (a) Unwindowed 2-D FFT image. (b) Windowed 2-D FFT image.

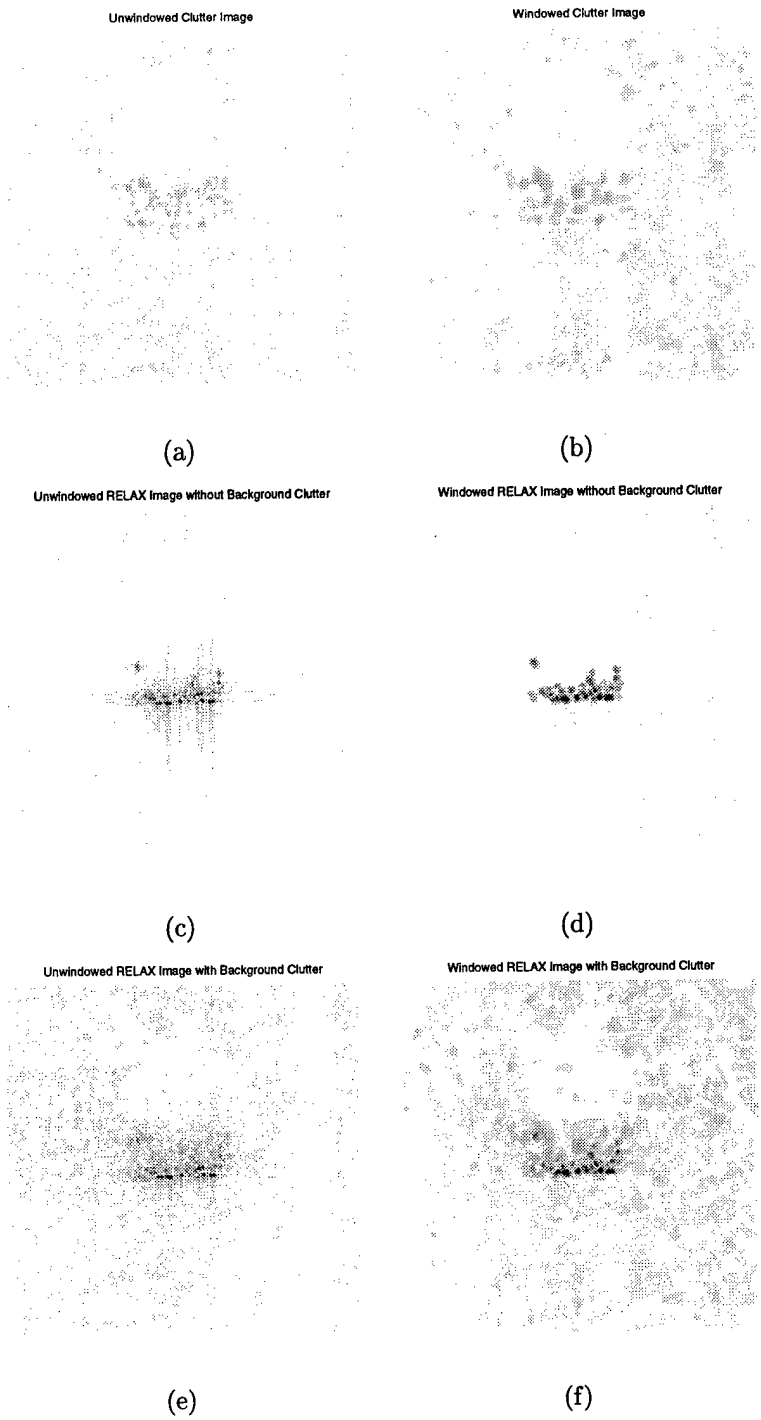


Figure 8.10: SAR images obtained via RELAX by using the MSTAR data hb03365.015 (90° azimuth and 15° elevation angles) with $\hat{K} = 37$. (a) Unwindowed clutter image. (b) Windowed clutter image. (c) Unwindowed RELAX image without background clutter. (d) Windowed RELAX image without background clutter. (e) Unwindowed RELAX image with background clutter. (f) Windowed RELAX image with background clutter.

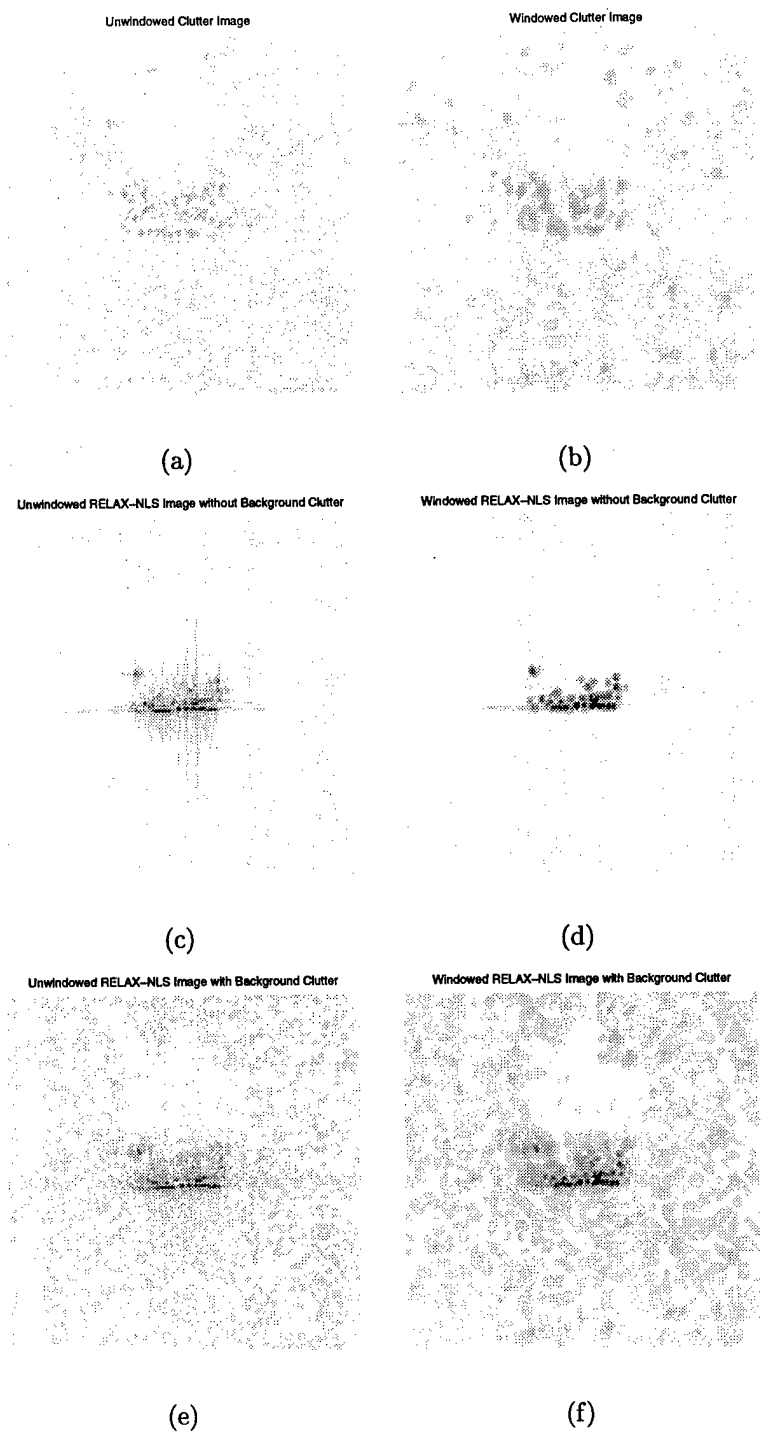


Figure 8.11: SAR images obtained via RELAX-NLS by using the MSTAR data hb03365.015 (90° azimuth and 15° elevation angles) with $\hat{K} = 36$. (a) Unwindowed clutter image. (b) Windowed clutter image. (c) Unwindowed RELAX-NLS image without background clutter. (d) Windowed RELAX-NLS image without background clutter. (e) Unwindowed RELAX-NLS image with background clutter. (f) Windowed RELAX-NLS image with background clutter.

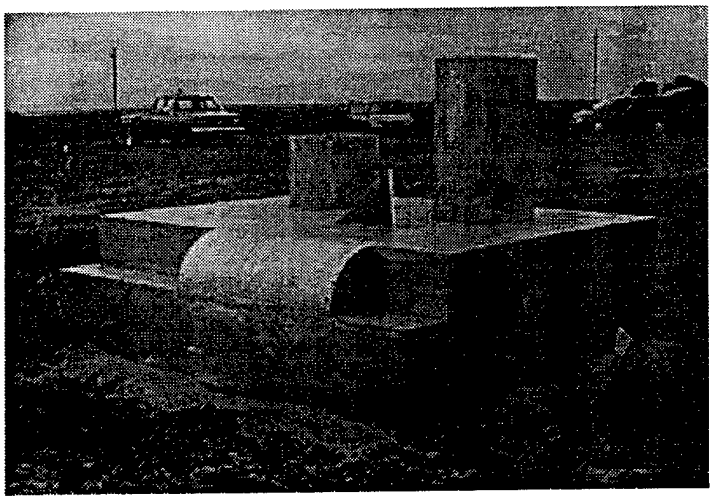
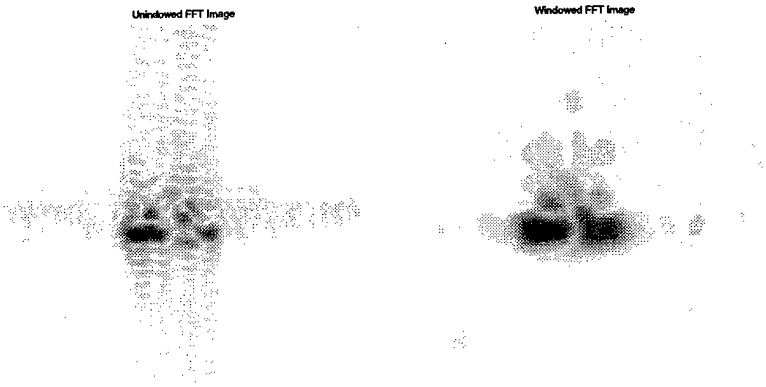


Figure 8.12: Target photo taken at 45° azimuth angle.



(a) (b)

Figure 8.13: SAR images obtained via 2-D FFT from the Slicy data hb15533.015 (0° azimuth and 30° elevation angles). (a) Unwindowed 2-D FFT image. (b) Windowed 2-D FFT image.

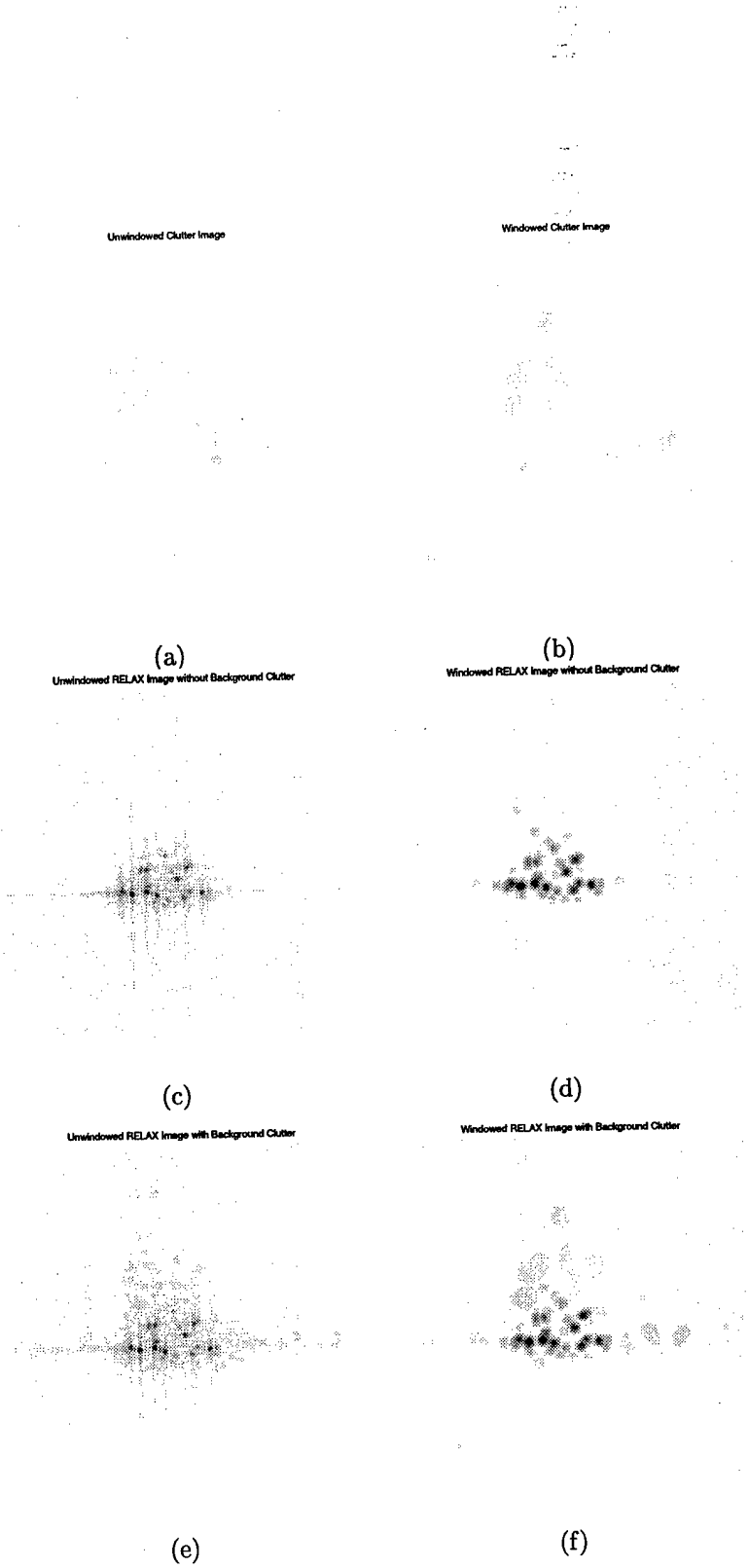


Figure 8.14: SAR images obtained via RELAX by using the Slicy data (0° azimuth and 30° elevation angles) with $\hat{K} = 36$. (a) Unwindowed clutter image. (b) Windowed clutter image. (c) Unwindowed RELAX image without background clutter. (d) Windowed RELAX image without background clutter. (e) Unwindowed RELAX image with background clutter. (f) Windowed RELAX image with background clutter.

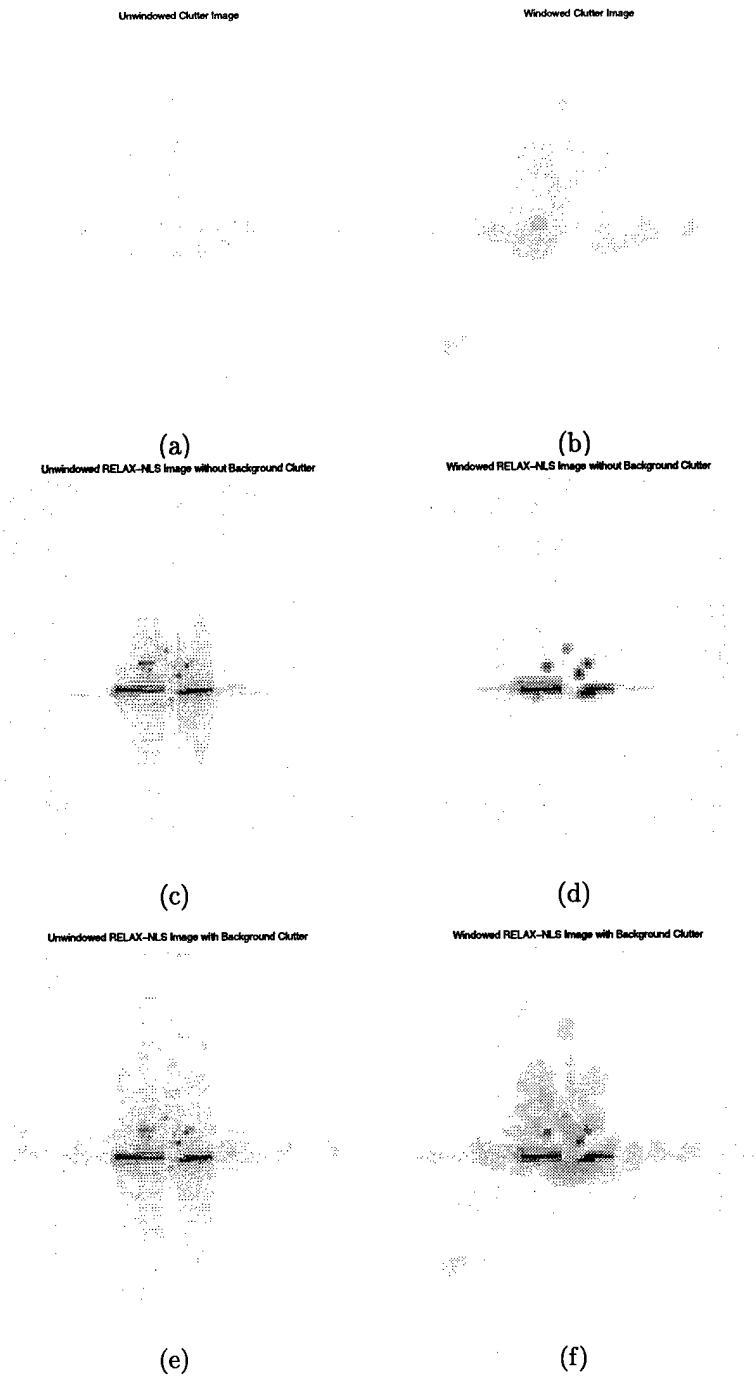


Figure 8.15: SAR images obtained via RELAX-NLS by using the Slicy data (0° azimuth and 30° elevation angles) with $\hat{K} = 24$. (a) Unwindowed clutter image. (b) Windowed clutter image. (c) Unwindowed RELAX-NLS image without background clutter. (d) Windowed RELAX-NLS image without background clutter. (e) Unwindowed RELAX-NLS image with background clutter. (f) Windowed RELAX-NLS image with background clutter.

9. Using Curvilinear SAR for Three-Dimensional Target Feature Extraction

9.1 Introduction

Three-dimensional (3-D) features of a target scatterer include the radar cross section (RCS), the two-dimensional (2-D) location (range and cross-range), and the height (the third dimensional parameter) of the scatterer. In [1], we described how to extract the 3-D target features via an interferometric synthetic aperture radar (IFSAR) system [2, 3], which uses a pair of vertically displaced antennas to obtain two coherent and parallel measurement apertures. The IFSAR system can be used for both 2-D and 3-D SAR imaging and 3-D target feature extraction. However, the IFSAR system suffers from ambiguity problems since it provides only two vertical parallel apertures. For example, the system cannot resolve more than one target scatterer at the same projected range and cross-range but at different heights [1].

In this chapter, we describe how to extract the 3-D target features via a curvilinear synthetic aperture radar (CLSAR) system [4, 5, 6]. The CLSAR system uses a single antenna to obtain a curved measurement aperture. The 3-D SAR images obtained via using FFT (fast Fourier transform) with CLSAR suffer from severe high sidelobes and hence are of little practical use. Hence CLSAR may not be suitable for the imaging of distributed targets. Yet CLSAR can be used with spectral estimation methods [4, 5, 6] to extract 3-D features of small targets consisting of a small number of point scatterers and can avoid the ambiguity problems suffered by IFSAR.

In [4, 5, 6], relaxation-based methods have been used with 3-D backprojection [7] for 3-D target feature extraction via CLSAR. The relaxation-based methods have been proved to be quite useful also in several other applications such as in radio astronomy [8], microwave imaging [9], spectral estimation [10], and in both 1-D (one-dimensional) and 2-D line spectral estimation and target feature extraction [11]. As explained in [11], these methods may have different implementation forms and structures and the implementation structure used in

[11] makes the method converge to the global minimizer of a nonlinear least squares (NLS) criterion with a high probability and in a fast manner.

In this chapter, a self-contained detailed derivation of the data model in a Cartesian coordinate is presented. The Cramér-Rao bounds (CRBs) of the parameter estimates are also derived. We describe how the RELAX algorithm [11] can be used for 3-D target feature extraction with CLSAR for different curvilinear apertures.

The remainder of the chapter is organized as follows. In Section 9.2, the data model is derived and the problem of interest is formulated. Section 9.3 discusses how the RELAX algorithm can be extended for 3-D target feature extraction via CLSAR. In Section 9.4, we derive the CRBs for the parameter estimates. Section 9.5 shows the results of several examples illustrating the performances of different curvilinear apertures and the RELAX method. Finally, Section 9.6 contains our conclusions.

9.2 Data Models and Problem Formulations

We start with establishing 1-D data models for point scatterers and then extend the discussions to 3-D data models and problem formulations.

9.2.1 High Range Resolution Radar

We first describe how one can obtain 1-D target features via a high range resolution radar as a preparation for the analysis in the following subsections. The range resolution of a radar is determined by the radar bandwidth. To achieve high resolution in range, the radar must transmit wide band pulses, which are often linear frequency modulated (FM) chirp pulses [12, 13] . A normalized chirp pulse can be written as

$$s(t) = e^{-j(2\pi f_0 t + \gamma t^2)}, \quad |t| \leq T_0/2, \tag{9.1}$$

where f_0 denotes the carrier frequency, 2γ denotes the FM rate, and T_0 denotes the width of the pulse. We assume that f_0, γ, and T_0 are known. The signal returned by a scatterer

of a target has the form

$$r(t) = \delta_\tau e^{-j[2\pi f_0(t-\tau)+\gamma(t-\tau)^2]}, \tag{9.2}$$

where δ_τ is determined by the RCS of the scatterer and τ denotes the round-trip time delay. The demodulated signal $\tilde{d}(t)$ is obtained by mixing $r(t)$ with $s^*(t-\tau_0)$ for some given τ_0 (see Section 9.2.2, where $(\cdot)^*$ denotes the complex conjugate,

$$\tilde{d}(t) = \delta_\tau e^{j2(\pi f_0-\gamma\tau_0)(\tau-\tau_0)} e^{-j\gamma(\tau-\tau_0)^2} e^{j2\gamma(\tau-\tau_0)t}. \tag{9.3}$$

The term $e^{-j\gamma(\tau-\tau_0)^2}$ in (9.3) is usually close to a constant for all $\tau_{\min} \leq \tau \leq \tau_{\max}$, where $\tau_{\max}$ and $\tau_{\min}$ correspond to the maximum and minimum values, respectively, of the round-trip time delays between the scatterers of a target and the radar and $\tau_{\min} \leq \tau_0 \leq \tau_{\max}$. This term can also be partially removed [12]. Let $\tilde{D}(\omega)$ denote the Fourier transform of $\tilde{d}(t)$. Then the inverse Fourier transform of $\tilde{D}(\omega)e^{j\frac{\omega^2}{4\gamma}}$ will have the term $e^{-j\gamma(\tau-\tau_0)^2}$ removed. Yet this removal can only be approximate since $\tilde{d}(t)$ is not known for all t and hence $\tilde{D}(\omega)$ is not known exactly. The closer $e^{-j\gamma(\tau-\tau_0)^2}$ is to a constant for $\tau_{\min} \leq \tau \leq \tau_{\max}$, the better its removal. With this removal, we have

$$d(t) = \delta_\tau e^{j2(\pi f_0-\gamma\tau_0)(\tau-\tau_0)} e^{j2\gamma(\tau-\tau_0)t}, \tag{9.4}$$

which is a complex sinusoid with frequency $2\gamma(\tau-\tau_0)$ and amplitude $\delta_\tau e^{j2(\pi f_0-\gamma\tau_0)(\tau-\tau_0)}$. We know $\tau_{\max}$ and $\tau_{\min}$ approximately since we assume that the altitude, antenna beamwidth, and grazing angle of the radar are known. We also assume that $(\tau_{\max}-\tau_{\min}) \ll T_0$. Then for $-T_0/2+\tau_{\max} \leq t \leq T_0/2+\tau_{\min}$, the scatterers of the target at different ranges correspond to different frequencies of the signal $d(t)$, while the RCS's of the scatterers are proportional to the amplitudes of the corresponding sinusoids. The ranges and RCS's of the target scatterers are the 1-D target features.

9.2.2 Full Synthetic Aperture Radar

We now describe how one can obtain 3-D target features via the full synthetic aperture shown in Figure 9.1(a), which prepares the ground for the discussions on CLSAR in the next subsection. The cross-range and height resolutions of an ordinary ranging radar is

limited by its antenna beamwidth. For an airborne or spaceborne system, a narrow antenna beamwidth requires an antenna that may be too large to be carried on board of the airplane or the spacecraft. Spotlight-mode SAR avoids this requirement by collecting coherent radar returns while viewing a target from many different azimuth and elevation angles, as shown in Figure 9.1(a). By properly processing the return signals, we can also achieve high resolution in both cross-range and elevation.

A broadside data collection geometry in a spotlight-mode synthetic aperture radar (SAR) is shown in Figure 9.2 [13]. The XYZ coordinate system is centered on a small patch of ground, where a target is located. The ground is illuminated by a narrow radio frequency (RF) beam from the moving radar that rotates (with radius R_0) around the coordinate origin. In Figure 9.2, R denotes the distance between the radar and a scatterer at the position (x, y, z), and θ and ϕ are the azimuth and elevation angles of the radar relative to the XYZ coordinate system. We assume that θ, ϕ, and R_0 are known.

The range R of the scatterer located at (x, y, z) can be written as

$$R = \left[(R_0 \cos\theta \cos\phi - x)^2 + (R_0 \sin\theta \cos\phi - y)^2 + (R_0 \sin\phi - z)^2\right]^{1/2}. \tag{9.5}$$

It has been shown in Appendix A that under the conditions given in Appendix A, R can be simplified as

$$R \approx \tilde{R}_0 - \tilde{x}\cos\theta\cos\phi - \tilde{y}\sin\theta\cos\phi - \tilde{z}\sin\phi, \tag{9.6}$$

where

$$\tilde{R}_0 = R_0 + \frac{[(x^2 - z^2)\sin\phi_0\cos\phi_0 - 2xz\cos^2\phi_0](\phi - \phi_0)}{2R_0} \tag{9.7}$$

$$\tilde{x} = x + \frac{-(y^2 + z^2)\cos\phi_0 + 2xz\sin\phi_0}{2R_0}, \tag{9.8}$$

$$\tilde{y} = y + \frac{xy\cos\phi_0 + yz\sin\phi_0}{R_0}, \tag{9.9}$$

and

$$\tilde{z} = z - \frac{(x^2 + y^2)\sin\phi_0}{2R_0}. \tag{9.10}$$

Note that the second terms of the right sides of (9.8), (9.9), and (9.10) are due to the range and elevation curvature effects and can be neglected for large R_0. Let $\tau_0 = \frac{2R_0}{c}$. Since $\tau =$

$\frac{2R}{c}$, then from (9.4), we have

$$d(t,\theta,\phi) = \delta_{x,y,z} \exp\left[\frac{j4(\pi f_0 - \gamma\tau_0 + \gamma t)}{c}(\tilde{R}_0 - R_0 - \tilde{x}\cos\theta\cos\phi - \tilde{y}\sin\theta\cos\phi - \tilde{z}\sin\phi)\right], \tag{9.11}$$

where $\delta_{x,y,z}$ is proportional to the RCS of the scatterer located at (x,y,z). For $4(\pi f_0 - \gamma\tau_0 + \gamma t)(\tilde{R}_0 - R_0)/c \ll 2\pi$, where $|t - \tau_0| < T_0$, we can write (9.11) as

$$d(t,\theta,\phi) \approx \delta_{x,y,z} e^{j(\tilde{x}t_x + \tilde{y}t_y + \tilde{z}t_z)}, \tag{9.12}$$

where

$$t_x = -\frac{4(\pi f_0 - \gamma\tau_0 + \gamma t)\cos\theta\cos\phi}{c}, \tag{9.13}$$

$$t_y = -\frac{4(\pi f_0 - \gamma\tau_0 + \gamma t)\sin\theta\cos\phi}{c}, \tag{9.14}$$

and

$$t_z = -\frac{4(\pi f_0 - \gamma\tau_0 + \gamma t)\sin\phi}{c}. \tag{9.15}$$

Note that $d(t,\theta,\phi)$ is a 3-D complex sinusoid. The frequencies of the 3-D sinusoid correspond to the 3-D location $(\tilde{x},\tilde{y},\tilde{z})$ of the scatterer, while the amplitude is proportional to its RCS. Note that $(\tilde{x},\tilde{y},\tilde{z})$ is not the true location (x,y,z) of the scatterer, but is close to (x,y,z) for large R_0. (See the Appendix B on how to calculate (x,y,z) from $(\tilde{x},\tilde{y},\tilde{z})$.) When a target has multiple scatterers, $d(t,\theta,\phi)$ in (9.12) will be a sum of sinusoids. The 3-D locations and RCS's of the target scatterers are the 3-D target features. Since usually the samples on the t, θ, and ϕ axes are uniformly spaced, the samples of t_x, t_y, and t_z occur at the points of a polar grid. Hence Polar-to-Cartesian interpolation may be needed for the data samples to occur at rectangular grid points. (See Section 9.5 for an alternative approach.)

After Polar-to-Cartesian interpolation and sampling, the signal obtained by the 3-D full aperture SAR can be written as:

$$y(n,\tilde{n},\bar{n}) = \sum_{k=1}^{K} \alpha_k e^{j(\omega_k n + \tilde{\omega}_k \tilde{n} + \bar{\omega}_k \bar{n})} + e(n,\tilde{n},\bar{n}), \tag{9.16}$$

where $n = 0,1,\cdots,N-1$, $\tilde{n} = 0,1,\cdots,\tilde{N}-1$, and $\bar{n} = 0,1,\cdots,\bar{N}-1$ with N, $\tilde{N}$, and $\bar{N}$ denoting the numbers of available data samples in the three dimensions; K denotes the

number of sinusoids; α_k, $k = 1, 2, \cdots, K$, denotes the unknown complex amplitude of the kth sinusoid; ω_k, $\tilde{\omega}_k$, and $\bar{\omega}_k$, $k = 1, 2, \cdots, K$, denote the 3-D unknown frequencies of the kth sinusoid; finally, $e(n, \tilde{n}, \bar{n})$, $n = 0, 1, \cdots, N-1$, $\tilde{n} = 0, 1, \cdots, \tilde{N}-1$, and $\bar{n} = 0, 1, \cdots, \bar{N}-1$, denotes the unknown noise. The sinusoidal frequencies $\omega_k = 2\pi f_k$, $\tilde{\omega}_k = 2\pi \tilde{f}_k$, and $\bar{\omega}_k = 2\pi \bar{f}_k$, correspond to the 3-D location of the kth scatterer of a radar target; α_k is determined by its radar cross section.

Let

$$\begin{aligned}
\mathbf{y} \;=\; \Big[\; & y(0,0,0),\;\; y(1,0,0),\;\; \cdots,\;\; y(N-1,0,0), \\
& y(0,1,0),\;\; y(1,1,0),\;\; \cdots,\;\; y(N-1,1,0), \\
& \cdots\cdots\;\; y(0,\tilde{N}-1,0),\;\; y(1,\tilde{N}-1,0),\;\; \cdots,\;\; y(N-1,\tilde{N}-1,0), \\
& y(0,0,1),\;\; y(1,0,1),\;\; \cdots,\;\; y(N-1,0,1), \\
& \cdots\cdots\;\; y(0,0,\bar{N}-1),\;\; y(1,0,\bar{N}-1),\;\; \cdots,\;\; y(N-1,0,\bar{N}-1), \\
& \cdots\cdots\;\; y(0,\tilde{N}-1,\bar{N}-1),\;\; y(1,\tilde{N}-1,\bar{N}-1),\;\; \cdots,\;\; y(N-1,\tilde{N}-1,\bar{N}-1)\;\Big]^T.
\end{aligned} \tag{9.17}$$

Let $\mathbf{e}$ be defined similarly from $e(n, \tilde{n}, \bar{n})$ as $\mathbf{y}$ from $y(n, \tilde{n}, \bar{n})$. Let

$$\mathbf{a}_k = \begin{bmatrix} 1 \\ e^{j\omega_k} \\ \vdots \\ e^{j(N-1)\omega_k} \end{bmatrix}, \quad \tilde{\mathbf{a}}_k = \begin{bmatrix} 1 \\ e^{j\tilde{\omega}_k} \\ \vdots \\ e^{j(\tilde{N}-1)\tilde{\omega}_k} \end{bmatrix}, \quad \bar{\mathbf{a}}_k = \begin{bmatrix} 1 \\ e^{j\bar{\omega}_k} \\ \vdots \\ e^{j(\bar{N}-1)\bar{\omega}_k} \end{bmatrix}. \tag{9.18}$$

Then

$$\mathbf{y} = \mathbf{A}\boldsymbol{\alpha} + \mathbf{e}, \tag{9.19}$$

where

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_K \end{bmatrix}^T, \tag{9.20}$$

and

$$\mathbf{A} = \begin{bmatrix} \bar{\mathbf{a}}_1 \otimes \tilde{\mathbf{a}}_1 \otimes \mathbf{a}_1 & \cdots & \bar{\mathbf{a}}_K \otimes \tilde{\mathbf{a}}_K \otimes \mathbf{a}_K \end{bmatrix}, \tag{9.21}$$

where $\otimes$ denotes the Kronecker product [14].

9.2.3 Curvilinear SAR

Since it is not practical to use the full aperture shown in Figure 9.1(a) to extract 3-D target features, a curvilinear aperture consisting of M different radar viewing angles, such as the one shown in Figure 9.1(b), may be an alternative. In the curvilinear SAR, the received data vector $\mathbf{y}_c$ is an $(MN) \times 1$ subvector of $\mathbf{y}$. Let $\mathbf{I}_c$ denote an $M \times (\tilde{N}\bar{N})$ matrix with each column and row containing only one non-zero unit element corresponding to the locations of the available data samples. Then

$$\mathbf{y}_c = \mathbf{A}_c \boldsymbol{\alpha} + \mathbf{e}, \tag{9.22}$$

where

$$\mathbf{A}_c = \left[\{\mathbf{I}_c(\bar{\mathbf{a}}_1 \otimes \tilde{\mathbf{a}}_1)\} \otimes \mathbf{a}_1 \quad \cdots \quad \{\mathbf{I}_c(\bar{\mathbf{a}}_K \otimes \tilde{\mathbf{a}}_K)\} \otimes \mathbf{a}_K \right]. \tag{9.23}$$

The unknown sinusoidal parameters $\{\omega_k, \tilde{\omega}_k, \bar{\omega}_k, \alpha_k, \}_{k=1}^{K}$ are our features of interest and are to be estimated from the $\mathbf{y}_c$ collected by CLSAR.

CLSAR, however, suffers from severe high sidelobes when used with FFT to form SAR images. We show this observation by using a simple simulated example as shown in Figure 9.4. Figures 9.4(a) and (b) show the mesh plots of the modulus of the RCS of a single scatterer obtained by using 2-D FFT (range information suppressed for the illustration purpose only) when the full aperture in Figure 9.1(a) and the curved aperture in Figure 9.3(a) are used, respectively. Note that the SAR images formed by using the nonparametric FFT method with CLSAR can be of little practical use due to the severe high sidelobes. In the following sections, we consider parametric 3-D target feature extraction methods for CLSAR.

9.3 The RELAX Algorithm

The RELAX algorithm [11] can be extended to extract the 3-D target features. We first consider using RELAX with the full aperture for 3-D target feature extraction and then extend the approach to the case of curvilinear apertures.

9.3.1 Full Aperture

The RELAX algorithm [11] minimizes the following nonlinear least squares (NLS) cost function:

$$C = \|\mathbf{y} - \mathbf{A}\boldsymbol{\alpha}\|^2, \tag{9.24}$$

where $||\cdot||$ denotes the Euclidean norm. When the noise $e(n, \tilde{n}, \bar{n})$ is the zero-mean white Gaussian random process, the NLS estimates obtained with RELAX coincide with the maximum likelihood (ML) estimates of the target features. When the noise is colored, the NLS estimates are no longer the ML estimates, but they still possess excellent statistical performance [11].

The minimization of the cost function C in (9.24) is a complicated optimization problem. We present below the relaxation-based (RELAX) minimization approach that leads to a conceptually and computationally simple method. For each fixed K, we perform a complete relaxation-based search by letting only the parameters of one scatterer vary and freezing all others at their most recently determined values. In this way, we will also take advantage of the fact that the parameter estimates for the first $K-1$ scatterers can be used to initialize the search for the parameters of the Kth one.

To make the paper self-contained, let us now briefly prepare for the RELAX approach. Let

$$\mathbf{y}_k = \mathbf{y} - \sum_{i=1, i\neq k}^{K} \hat{\alpha}_i (\hat{\bar{\mathbf{a}}}_i \otimes \hat{\tilde{\mathbf{a}}}_i \otimes \hat{\mathbf{a}}_i), \tag{9.25}$$

where $\hat{\mathbf{a}}_i$, $\hat{\tilde{\mathbf{a}}}_i$, and $\hat{\bar{\mathbf{a}}}_i$ are formed, similarly to those in Equation (9.18), from $\hat{\omega}_i$, $\hat{\tilde{\omega}}_i$, and $\hat{\bar{\omega}}_i$, respectively, and $\{\hat{\omega}_i, \hat{\tilde{\omega}}_i, \hat{\bar{\omega}}_i, \hat{\alpha}_i\}_{i=1, i\neq k}^{K}$ are assumed to have been estimated. Then minimizing C in (9.24) with respect to α_k yields the estimate $\hat{\alpha}_k$ of α_k:

$$\hat{\alpha}_k = \left. \frac{[\bar{\mathbf{a}}_k \otimes \tilde{\mathbf{a}}_k \otimes \mathbf{a}_k]^H \mathbf{y}_k}{N\tilde{N}\bar{N}} \right|_{\omega_k = \hat{\omega}_k, \tilde{\omega}_k = \hat{\tilde{\omega}}_k, \bar{\omega}_k = \hat{\bar{\omega}}_k}, \tag{9.26}$$

and

$$\{\hat{\omega}_k, \hat{\tilde{\omega}}_k, \hat{\bar{\omega}}_k\} = \arg \max_{\omega_k, \tilde{\omega}_k, \bar{\omega}_k} \left| [\bar{\mathbf{a}}_k \otimes \tilde{\mathbf{a}}_k \otimes \mathbf{a}_k]^H \mathbf{y}_k \right|^2. \tag{9.27}$$

Hence $\{\hat{\omega}_k, \hat{\tilde{\omega}}_k, \hat{\bar{\omega}}_k\}$ can be obtained as the location of the dominant peak of the 3-D periodogram,

$$\frac{\left|[\bar{\mathbf{a}}_k \otimes \tilde{\mathbf{a}}_k \otimes \mathbf{a}_k]^H \mathbf{y}_k\right|^2}{N\tilde{N}\bar{N}},$$

which can be computed efficiently with 3-D FFT. Note that padding with zeros for the 3-D FFT is necessary to achieve high accuracy. An alternative approach is to find an approximate location corresponding to the global maximum with the 3-D FFT without much zero-padding and then use the approximate location as the initial condition to find a more accurate position via a multidimensional search method, such as the FMINV function in PV-WAVE. We used the latter approach in our examples presented in Section 9.5. Note that $\hat{\alpha}_k$ is easily computed from the complex height of the peak of $[\bar{\mathbf{a}}_k \otimes \tilde{\mathbf{a}}_k \otimes \mathbf{a}_k]^H \mathbf{y}_k/(N\tilde{N}\bar{N})$.

With the above preparations, we now proceed to present the steps of the RELAX algorithm for 3-D target feature extraction with the full aperture SAR.

Step (1): Assume $K = 1$. Obtain $\{\hat{\omega}_k, \hat{\tilde{\omega}}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\}_{k=1}$ from $\mathbf{y}$ by using (9.27) and (9.26).

Step (2): Assume $K = 2$. Compute $\mathbf{y}_2$ with (9.25) by using $\{\hat{\omega}_k, \hat{\tilde{\omega}}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\}_{k=1}$ obtained in Step (1). Obtain $\{\hat{\omega}_k, \hat{\tilde{\omega}}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\}_{k=2}$ from $\mathbf{y}_2$. Next, compute $\mathbf{y}_1$ by using $\{\hat{\omega}_k, \hat{\tilde{\omega}}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\}_{k=2}$ and then redetermine $\{\hat{\omega}_k, \hat{\tilde{\omega}}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\}_{k=1}$ from $\mathbf{y}_1$.

Iterate the previous two substeps until "practical convergence" is achieved (to be discussed later on).

Step (3): Assume $K = 3$. Compute $\mathbf{y}_3$ by using $\{\hat{\omega}_k, \hat{\tilde{\omega}}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\}_{k=1}^2$ obtained in Step (2). Obtain $\{\hat{\omega}_k, \hat{\tilde{\omega}}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\}_{k=3}$ from $\mathbf{y}_3$. Next, compute $\mathbf{y}_1$ by using $\{\hat{\omega}_k, \hat{\tilde{\omega}}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\}_{k=2}^3$ and redetermine $\{\hat{\omega}_k, \hat{\tilde{\omega}}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\}_{k=1}$ from $\mathbf{y}_1$. Then compute $\mathbf{y}_2$ by using $\{\hat{\omega}_k, \hat{\tilde{\omega}}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\}_{k=1,3}$ and redetermine $\{\hat{\omega}_k, \hat{\tilde{\omega}}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\}_{k=2}$ from $\mathbf{y}_2$.

Iterate the previous three substeps until "practical convergence".

Remaining Steps: Continue similarly until K is equal to the desired or estimated number of sinusoids. (Whenever K is unknown, it can be estimated from the available data, for instance, by using generalized AIC rules which are particularly tailored to the RELAX method of parameter estimation. See, e.g., [11].)

The "practical convergence" in the iterations of the above RELAX method may be de-

termined by checking the relative change ε of the cost function $C\left(\{\hat{\omega}_k, \hat{\tilde{\omega}}_k, \hat{\bar{\omega}}_k, \hat{\alpha}_k\}_{k=1}^{K}\right)$ in (9.24) between two consecutive iterations. Our numerical examples show that the iterations usually converge in a few steps.

9.3.2 Curvilinear Aperture

The RELAX algorithm for this case of curvilinear aperture is similar to that of the full aperture except that Equations (9.26) and (9.27) above are replaced by

$$\hat{\boldsymbol{\alpha}}_k = \left.\frac{\left[\{\mathbf{I}_c(\bar{\mathbf{a}}_k \otimes \tilde{\mathbf{a}}_k)\} \otimes \mathbf{a}_k\right]^H \mathbf{y}_{c_k}}{MN}\right|_{\omega_k=\hat{\omega}_k, \tilde{\omega}_k=\hat{\tilde{\omega}}_k, \bar{\omega}_k=\hat{\bar{\omega}}_k}, \tag{9.28}$$

and

$$\{\hat{\omega}_k, \hat{\tilde{\omega}}_k, \hat{\bar{\omega}}_k\} = \arg \max_{\omega_k, \tilde{\omega}_k, \bar{\omega}_k} \left|\left[\{\mathbf{I}_c(\bar{\mathbf{a}}_k \otimes \tilde{\mathbf{a}}_k)\} \otimes \mathbf{a}_k\right]^H \mathbf{y}_{c_k}\right|^2, \tag{9.29}$$

respectively, where

$$\mathbf{y}_{c_k} = \mathbf{y}_c - \sum_{i=1, i\neq k}^{K} \hat{\alpha}_i \left[\{\mathbf{I}_c(\hat{\bar{\mathbf{a}}}_i \otimes \hat{\tilde{\mathbf{a}}}_i)\} \otimes \hat{\mathbf{a}}_i\right]. \tag{9.30}$$

Let $\check{\mathbf{y}}_{c_k}$ be similar to $\mathbf{y}_k$ except that the elements in $\mathbf{y}_k$ that are missing in $\mathbf{y}_{c_k}$ are replaced with zeros in $\check{\mathbf{y}}_{c_k}$. Then the right hand side of (9.29) can also be computed by applying 3-D FFT to $\check{\mathbf{y}}_{c_k}$.

For the special case where the curvilinear aperture consists of the orthogonal subapertures such as those shown in Figure 9.3, the aforementioned zero-filling and 3-D FFT can be avoided. Note that the analysis based on Figure 9.3(a) also holds for Figures 9.3(b) and (c) since they are the rotations of the coordinate system in Figure 9.3(a). Hence we only consider Figure 9.3(a) below. For this case, $\mathbf{A}_c$ becomes

$$\mathbf{A}_c = \left[\begin{bmatrix} \tilde{\mathbf{a}}_1 \\ \bar{\mathbf{b}}_1 \end{bmatrix} \otimes \mathbf{a}_1 \quad \cdots \quad \begin{bmatrix} \tilde{\mathbf{a}}_k \\ \bar{\mathbf{b}}_k \end{bmatrix} \otimes \mathbf{a}_k\right], \tag{9.31}$$

where $\bar{\mathbf{b}}_k$ is a subvector of $\bar{\mathbf{a}}_k$ without its first element. Then

$$\hat{\alpha}_k = \left.\frac{\left\{\begin{bmatrix} \tilde{\mathbf{a}}_k^T & \bar{\mathbf{b}}_k^T \end{bmatrix}^T \otimes \mathbf{a}_k\right\}^H \mathbf{y}_{c_k}}{(\tilde{N} + \bar{N} - 1)N}\right|_{\omega_k=\hat{\omega}_k, \tilde{\omega}_k=\hat{\tilde{\omega}}_k, \bar{\omega}_k=\hat{\bar{\omega}}_k}, \tag{9.32}$$

and

$$\begin{aligned}\{\hat{\omega}_k, \hat{\tilde{\omega}}_k, \hat{\bar{\omega}}_k\} &= \arg\max_{\omega_k, \tilde{\omega}_k, \bar{\omega}_k} \left| \left\{ \begin{bmatrix} \tilde{\mathbf{a}}_k^T & \bar{\mathbf{b}}_k^T \end{bmatrix}^T \otimes \mathbf{a}_k \right\}^H \mathbf{y}_{c_k} \right|^2 \\ &= \arg\max_{\omega_k, \tilde{\omega}_k, \bar{\omega}_k} \left| (\tilde{\mathbf{a}}_k \otimes \mathbf{a}_k)^H \mathbf{y}_{c_{k1}} + (\bar{\mathbf{b}}_k \otimes \mathbf{a}_k)^H \mathbf{y}_{c_{k2}} \right|^2, \end{aligned} \tag{9.33}$$

where

$$\mathbf{y}_{c_k} \triangleq \begin{bmatrix} \mathbf{y}_{c_{k1}} \\ \mathbf{y}_{c_{k2}} \end{bmatrix} \quad \begin{matrix} (\tilde{N}N) \times 1 \\ (\bar{N}-1)N \times 1. \end{matrix} \tag{9.34}$$

The right hand side of (9.33) can be calculated efficiently with the 2-D FFT's, which consists of applying 2-D FFT to $\mathbf{y}_{c_{k1}}$ and $\begin{bmatrix} \mathbf{0}^T & \mathbf{y}_{c_{k2}}^T \end{bmatrix}^T$, where $\mathbf{0}$ here denotes the $\bar{N} \times 1$ zero vector. Then $\hat{\alpha}_k$ is easily computed from the complex height of the peak of $[(\tilde{\mathbf{a}}_k \otimes \mathbf{a}_k)^H \mathbf{y}_{c_{k1}} + (\bar{\mathbf{b}}_k \otimes \mathbf{a}_k)^H \mathbf{y}_{c_{k2}}]/[(\tilde{N} + \bar{N} - 1)N]$. Hence using the apertures in Figure 9.3 can significantly reduce the amount of computations needed for target feature extraction.

9.4 Cramér-Rao Bound of the Paramaeter Estimates

We derive the CRB matrix for the parameter estimates when the noise covariance matrix is arbitrary and unknown.

Consider first the case of the full aperture. Let $\mathbf{Q} = \mathrm{E}\{\mathbf{e}\mathbf{e}^H\}$ be the noise covariance matrix. The extended Slepian-Bangs formula for the ijth element of the Fisher information matrix has the form [15, 16]:

$$\{\mathrm{FIM}\}_{ij} = \mathrm{tr}\left(\mathbf{Q}^{-1}\mathbf{Q}_i'\mathbf{Q}^{-1}\mathbf{Q}_j'\right) + 2\mathrm{Re}\left[\left(\boldsymbol{\alpha}^H\mathbf{A}^H\right)_i' \mathbf{Q}^{-1} \left(\mathbf{A}\boldsymbol{\alpha}\right)_j'\right], \tag{9.35}$$

where $\mathbf{X}_i'$ denotes the derivative of $\mathbf{X}$ with respect to the ith unknown parameter, $\mathrm{tr}(\mathbf{X})$ denotes the trace of $\mathbf{X}$, and $\mathrm{Re}(\mathbf{X})$ denotes the real part of $\mathbf{X}$. Note that FIM is block diagonal since $\mathbf{Q}$ does not depend on the parameters in $(\mathbf{A}\boldsymbol{\alpha})$, and $(\mathbf{A}\boldsymbol{\alpha})$ does not depend on the elements of $\mathbf{Q}$. Hence the CRB matrix for the target features of interest can be calculated from the second term on the right side of (9.35). Let

$$\boldsymbol{\eta} = \begin{bmatrix} \mathrm{Re}^T(\boldsymbol{\alpha}) & \mathrm{Im}^T(\boldsymbol{\alpha}) & \boldsymbol{\omega}^T & \tilde{\boldsymbol{\omega}}^T & \bar{\boldsymbol{\omega}}^T \end{bmatrix}, \tag{9.36}$$

where $\boldsymbol{\omega} = \begin{bmatrix} \omega_1 & \omega_2 & \cdots & \omega_K \end{bmatrix}^T$, $\tilde{\boldsymbol{\omega}} = \begin{bmatrix} \tilde{\omega}_1 & \tilde{\omega}_2 & \cdots & \tilde{\omega}_K \end{bmatrix}^T$, and $\bar{\boldsymbol{\omega}} = \begin{bmatrix} \bar{\omega}_1 & \bar{\omega}_2 & \cdots & \bar{\omega}_K \end{bmatrix}^T$. Let

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_1 & \mathbf{F}_2 & \mathbf{F}_3 & \mathbf{F}_4 & \mathbf{F}_5 \end{bmatrix}, \tag{9.37}$$

where

$$\mathbf{F}_1 = \mathbf{A}, \tag{9.38}$$

$$\mathbf{F}_2 = j\mathbf{A}, \tag{9.39}$$

$$\mathbf{F}_3 = \mathbf{D}_\omega \boldsymbol{\Phi}, \tag{9.40}$$

$$\mathbf{F}_4 = \mathbf{D}_{\tilde{\omega}} \boldsymbol{\Phi}, \tag{9.41}$$

and

$$\mathbf{F}_5 = \mathbf{D}_{\bar{\omega}} \boldsymbol{\Phi}, \tag{9.42}$$

with $\mathbf{A}$ being defined in (9.21), the kth columns of $\mathbf{D}_\omega$, $\mathbf{D}_{\tilde{\omega}}$, and $\mathbf{D}_{\bar{\omega}}$ being $\partial[\bar{\mathbf{a}}_k \otimes \tilde{\mathbf{a}}_k \otimes \mathbf{a}_k]/\partial\omega_k, \partial[\bar{\mathbf{a}}_k \otimes \tilde{\mathbf{a}}_k \otimes \mathbf{a}_k]/\partial\tilde{\omega}_k$, and $\partial[\bar{\mathbf{a}}_k \otimes \tilde{\mathbf{a}}_k \otimes \mathbf{a}_k]/\partial\bar{\omega}_k$, respectively, and

$$\boldsymbol{\Phi} = \mathrm{diag}\left\{ \alpha_1, \quad \alpha_2, \quad \cdots \quad \alpha_K \right\}. \tag{9.43}$$

Then the CRB matrix for the parameter vector $\boldsymbol{\eta}$ is given by:

$$\mathrm{CRB}(\boldsymbol{\eta}) = [2\mathrm{Re}(\mathbf{F}^H \mathbf{Q}^{-1} \mathbf{F})]^{-1}. \tag{9.44}$$

For the case of curvilinear aperture, the CRB matrix for the target parameters is similar to the one in (9.44) except that the $\mathbf{A}$ in (9.44) is now replaced by $\mathbf{A}_c$.

9.5 Numerical and Experimental Results

We use CRBs to study the performances of different curved apertures for target feature extraction. We also present several examples showing the performances of the RELAX algorithm presented in this paper for 3-D target feature extraction with a CLSAR.

9.5.1 Performance Analysis of Different Curvilinear Apertures via CRBs

Since the CRBs are the best unbiased performance any asymptotic estimator can achieve, we compare the CRBs of the target features when different curved apertures are used for 3-D target feature extraction and compare them with the CRB of the full aperture. Without loss of generality, let us consider the case of a single scatter with $\alpha = 1$ and $\omega = \tilde{\omega} = \bar{\omega} = 0$. The additive noise is assumed to be zero-mean white Gaussian with variance $\sigma^2 = 40$. The full aperture data is generated according to (9.16) with $N = \tilde{N} = \bar{N} = 32$.

Consider first the example of $M = 63$. The curved apertures we consider include the parabolic one in Figure 9.1(b), the **L**-shaped one in Figure 9.3(a), and the ones in Figure 9.5. For the arc apertures, the Arc-1 aperture is a quarter of a circle whose center is at the upper right corner of the full aperture in Figure 9.1(a) and whose radius is 31, as shown in Figure 9.5(b), and the Arc-2 aperture is one half of a parabolic aperture whose vertex is at the lower right corner of the full aperture and who starts at the upper left corner of the full aperture in Figure 9.1(a), as shown in Figure 9.5(c). These curvilinear apertures are subsets of the full aperture in Figure 9.1(a) and are made to be as large as possible. For example, the radius of the circular aperture is 15.5 in this example. Table 9.1 shows the CRBs of the target parameters. As expected, the CRB for ω is the same for all of the curvilinear apertures since ω is in the range direction, which is normal to the plane of the curvilinear apertures. Comparing the CRBs for the **L**-shaped, Arc-1, and Arc-2 apertures, we note that as expected, the more curved the aperture is, the lower the CRBs for the target features. Since the Arc-2 aperture is the least curved, it has the largest CRBs. When the aperture becomes one straight line and hence no longer curved, the CRBs go to infinity since we can no longer extract 3-D target features. Note also that from the left to right of Table 9.1, the CRBs for $\tilde{\omega}$ and $\bar{\omega}$ decrease since the aperture length decreases. The CRB for α, however, does not always decrease.

Consider next the circular aperture in Figure 9.5(a) when $\omega = \tilde{\omega} = \bar{\omega} = 0$. Figure 9.6 shows the CRBs of the target parameters as a function of the radius of the circular aperture for different M. Note that as expected, the larger the radius and/or M, the lower CRBs of

the target parameters. Note also that, as expected, the CRB for ω does not change with the radius for the fixed M since again ω is in the range direction, which is orthogonal to the plane of the curvilinear aperture.

9.5.2 Experimental Examples

We now present two experimental examples to demonstrate the performances of the RELAX algorithm for 3-D target feature extraction with a CLSAR. In the following examples, we use $\varepsilon = 0.01$ to test the convergence of the RELAX algorithm.

We first consider an experimental example, where the indoor data was obtained by the Radar Signature Branch, Naval Air Warfare Center, Mugu, California. The radar carrier frequency is 9.968 GHz and the bandwidth 1.524 GHz. The $32 \times 32 \times 32$ data set was obtained with the full aperture shown in Figure 9.1(a) with 32 samples in each dimension and the angular increments were 0.28°. The target consists of $K = 8$ corner reflectors with a cubic configuration and was about 15 meters away from the radar. The scatterers were about 0.5 meters apart.

Instead of Polar-to-Cartesian interpolation, we created an $N \times \tilde{N} \times \bar{N}$ rectangular grid and mapped the data sample at each (t_x, t_y, t_z) to the nearest grid point, where t_x, t_y, and t_z are functions of t, θ, and ϕ (see Equations (9.13) to (9.15)). Let $\check{N}$ denote the number of available data samples. Then for both full and curvilinear apertures, the data model can be written as

$$\mathbf{y}_r = \mathbf{I}_r \mathbf{A} \boldsymbol{\alpha} + \mathbf{e}, \tag{9.45}$$

where $\mathbf{I}_r$ denotes an $\check{N} \times (N\tilde{N}\bar{N})$ matrix with each column and row containing only one non-zero unit element corresponding to the locations of the available data samples in the rectangular grid. Note that the larger the dimensions of the rectangular grid, the more accurate the mapping approximation. For large enough dimensions, however, the noise will become the dominant source of error. For our examples, we chose $N = \tilde{N} = \bar{N} = 128$. As compared to Polar-to-Cartesian interpolation, the grid mapping approach avoids the interpolation step needed by the former, which could be complicated for an arbitrary

curvilinear aperture. The former approach, however, is computationally more efficient due to its smaller data dimensions.

The steps of applying RELAX to $\mathbf{y}_r$ are similar to those for the case of full aperture except that Equations (9.26) and (9.27) above are replaced by

$$\hat{\boldsymbol{\alpha}}_k = \left. \frac{\left[\mathbf{I}_r(\bar{\mathbf{a}}_k \otimes \tilde{\mathbf{a}}_k \otimes \mathbf{a}_k)\right]^H \mathbf{y}_{r_k}}{\check{N}} \right|_{\omega_k=\hat{\omega}_k, \tilde{\omega}_k=\hat{\tilde{\omega}}_k, \bar{\omega}_k=\hat{\bar{\omega}}_k}, \tag{9.46}$$

and

$$\{\hat{\omega}_k, \hat{\tilde{\omega}}_k, \hat{\bar{\omega}}_k\} = \arg \max_{\omega_k, \tilde{\omega}_k, \bar{\omega}_k} \left| \left[\mathbf{I}_r(\bar{\mathbf{a}}_k \otimes \tilde{\mathbf{a}}_k \otimes \mathbf{a}_k)\right]^H \mathbf{y}_{r_k} \right|^2, \tag{9.47}$$

respectively, where

$$\mathbf{y}_{r_k} = \mathbf{y}_r - \sum_{i=1, i\neq k}^{K} \hat{\alpha}_i \mathbf{I}_r \left(\hat{\bar{\mathbf{a}}}_i \otimes \hat{\tilde{\mathbf{a}}}_i \otimes \hat{\mathbf{a}}_i\right). \tag{9.48}$$

Let $\check{\mathbf{y}}_{r_k}$ be similar to $\mathbf{y}_k$ except that the elements in $\mathbf{y}_k$ that are missing in $\mathbf{y}_{r_k}$ are replaced with zeros in $\check{\mathbf{y}}_{r_k}$. Then the right hand side of (9.47) can also be computed by applying 3-D FFT to $\check{\mathbf{y}}_{r_k}$. Since the dimensions of the rectangular grid are larger than the dimensions of the original data due to zero-filling, the 3-D FFT of $\check{\mathbf{y}}_{r_k}$ is approximately periodic and we should limit our attentions to only one period. For our examples, the former is four times as large as the latter and hence the peak searching in (9.47) is limited to the frequency intervals $[\ 2\pi \times (-0.125) \quad 2\pi \times 0.125\ ]$. Note that the computational advantages of FFT over DFT (discrete Fourier transform) and backprojection diminish as the dimensions of the rectangular grid increases, especially for curvilinear apertures.

We remark that for the special case of the curvilinear apertures consisting of orthogonal subapertures shown in Figure 9.3, we can still use 2-D FFT with the rectangular grid mapping approach and we omit the details here for the interest of brevity. For this special case, though, the Polar-to-Cartesian interpolations needed are 2-D and hence are much easier to implement than for an arbitrary curvilinear aperture.

For the data model in (9.45), the CRB matrix for the target parameters of interest is similar to the one in (9.44) except that the $\mathbf{A}$ in (9.44) is now replaced by $\mathbf{I}_r\mathbf{A}$. The CRB analysis results for this data model should also be similar to those observed in the

previous subsection. Hence the CRB analysis is omitted for this data model for the interest of conciseness.

Figure 9.7 shows the extracted scatterers when $K = 8$ is used with RELAX. Figure 9.7(a) is obtained when the full aperture shown in Figure 9.1(a) is used to extract the 3-D target features and Figure 9.7(b) is obtained when the L-shaped aperture shown in Figure 9.3(a) is used. The centers of the circles denote the locations of the extracted scatterers in 3-D space and the radius of each circle is proportional to the modulus of the RCS of the corresponding scatterer. The triangles show the projections of the scatterer locations onto the horizontal plane and their sizes are also scaled according to the RCS's of the scatterers. Note that Figure 9.7(b) is similar to Figure 9.7(a) even though the former is obtained by using only 0.19% amount of data used by the latter.

Consider next an experimental example, where the field data was obtained by the Deployable Signature Measurement System (DSMS), Carderock Division, Naval Surface Warfare Center, Bethesda, Maryland. The radar was carried on board of a helicoptor. The radar carrier frequency is 9.449 GHz and the bandwidth 0.498 GHz. The data set was obtained with the curved aperture shown in Figure 9.8 where there are 64 look angles and 64 samples per look angle. The radar was about 300 meters away from the ground target. The ground target consists of 13 corner reflectors on the ground plane and 7 corner reflectors mounted on a wooden tripod that is about 2.65 meters tall. The true distribution of the scatterers is shown in Figure 9.9(a), where the centers of the squares denote the locations of the scatterers in 3-D space and the length of each square is proportional to the modulus of the RCS of the corresponding scatterer. The triangles show the projections of the scatterer locations onto the ground plane and their sizes are also scaled to be proportional to the RCS's of the scatterers.

For this example, we also use the rectangular grid mapping with $N = \tilde{N} = \bar{N} = 128$. Figure 9.9(b) shows the scatterers obtained by using RELAX with $K = 20$. We note that among the 20 scatterers, we got 18 of them approximately correct. The two that are missing could be because they are in the shadows of the other scatterers.

9.6 Conclusions

The main subject of this chapter has been to extract the 3-D target features via a curvilinear synthetic aperture radar (CLSAR). Since CLSAR itself is a relatively new technology, a self-contained derivation of the data model has been presented. The Cramér-Rao bounds (CRBs) on the parameter estimates have also been derived. We have used the CRBs to study the performances of different curvilinear apertures for target feature extraction. Finally, the RELAX parameter estimation method has been extended to extract the 3-D target features with a CLSAR. We have presented several experimental examples showing the feature extraction performances of the RELAX algorithm.

Acknowledgments

We acknowledge Mr. K. V. Viswanath for his help in plotting some of the figures in the paper.

Appendix A - The Approximation of the Range R

The range R in (9.5) can be approximated as follows. Under the conditions $\frac{x}{R_0} \ll 1$, $\frac{y}{R_0} \ll 1$, and $\frac{z}{R_0} \ll 1$, we have

$$R = R_0 \left[1 - 2\frac{x}{R_0}\cos\theta\cos\phi - 2\ fracyR_0 \sin\theta\cos\phi - 2\frac{z}{R_0}\sin\phi + \frac{x^2+y^2+z^2}{R_0^2} \right]^{1/2} \tag{9.49}$$

$$\approx R_0 \left[1 - \frac{x}{R_0}\cos\theta\cos\phi - \frac{y}{R_0}\sin\theta\cos\phi - \frac{z}{R_0}\sin\phi + \frac{x^2+y^2+z^2}{2R_0^2} - \frac{(x\cos\theta\cos\phi + y\sin\theta\cos\phi + z\sin\phi)^2}{2R_0^2} \right]. \tag{9.50}$$

Let $\psi = \phi - \phi_0$, where ϕ_0 is the average of all ϕ used to form the synthetic aperture. For very small ψ, we have

$$\cos\phi \approx \cos\phi_0 - (\sin\phi_0)\psi, \tag{9.51}$$

$$\sin\phi \approx \sin\phi_0 + (\cos\phi_0)\psi. \tag{9.52}$$

For very small θ, we have $\cos\theta \approx 1$, $\sin\theta \approx \theta$. Then keeping first-order terms yields

$$\begin{aligned}
& x^2\sin^2\phi + x^2\sin^2\theta\cos^2\phi + y^2\sin^2\phi + y^2\cos^2\theta\cos^2\phi + z^2\cos^2\phi \\
\approx\ & x^2\sin\phi[\sin\phi_0 + (\cos\phi_0)\psi] + y^2\sin\phi[\sin\phi_0 + (\cos\phi_0)\psi] + y^2\cos\phi[\cos\phi_0 - (\sin\phi_0)\psi] \\
& + z^2\cos\phi[\cos\phi_0 - (\sin\phi_0)\psi] \\
\approx\ & [(x^2+y^2)\sin\phi_0]\sin\phi + [(y^2+z^2)\cos\phi_0]\cos\theta\cos\phi + [(x^2-z^2)\sin\phi_0\cos\phi_0]\psi. \quad (9.53)
\end{aligned}$$

We also have

$$\begin{aligned}
& 2xy\cos\theta\sin\theta\cos^2\phi + 2yz\sin\theta\cos\phi\sin\phi + 2xz\cos\theta\cos\phi\sin\phi \\
\approx\ & 2xy\sin\theta\cos\phi[\cos\phi_0 - (\sin\phi_0)\psi] + 2yz\sin\theta\cos\phi[\sin\phi_0 + (\cos\phi_0)\psi] \\
& + 2xz\cos\theta\cos\phi[\sin\phi_0 + (\cos\phi_0)\psi] \\
\approx\ & 2(xy\cos\phi_0 + yz\sin\phi_0)\sin\theta\cos\phi + (2xz\sin\phi_0)\cos\theta\cos\phi + (2xz\cos^2\phi_0)\psi. (9.54)
\end{aligned}$$

Hence R is simplified to (9.6).

Appendix B - Calculating (x, y, z) from $(\tilde{x}, \tilde{y}, \tilde{z})$

Given R_0, ϕ_0, and $(\tilde{x}, \tilde{y}, \tilde{z})$, (x, y, z) can be determined from (9.8), (9.9), and (9.10) as follows:

$$\begin{aligned}
& \eta_1^2\eta_2^2 y^8 + 2\eta_1 R_0\left[\eta_1\left(1+\eta_2^2\right)(R_0 + 2\eta_2\tilde{z}) + \eta_1 R_0 - 2\eta_2^4\tilde{x}\right]y^6 + R_0^2\left[2\eta_2^2\left(1+\eta_2^2\right)\tilde{y}^2\right. \\
& +4\eta_1^2\left(\eta_2^2+1\right)^2\tilde{z}^2 + 4\eta_2^6\tilde{x}^2 + 4R_0\eta_1^2\eta_2\left(5+\eta_2^2\right)\tilde{z} - 8\eta_2^3\eta_1\left(1+\eta_2^2\right)\tilde{x}\tilde{z} \\
& \left.+4R_0\left(2\eta_1^4 - 3\eta_2^4\right)\tilde{x} + R_0^2\eta_1^2\left(-8+\eta_2^2\right)\right]y^4 - 8\eta_1 R_0^3\left(\eta_1 R_0 + 2\eta_1\eta_2\tilde{z} - 2\eta_2^2\tilde{x}\right)\tilde{y}y^3 \\
& +2\eta_2 R_0^3\left[2\tilde{z}\left(1+\eta_2^4\right) + R_0\eta_2\left(3+\eta_2^2\right) + 2\eta_2^3\eta_1\tilde{x}\right]\tilde{y}^2y^2 - 8\eta_2^2 R_0^4\tilde{y}^3 y + \eta_1^2\eta_2^2 R_0^4\tilde{y}^4 \\
& = 0, \qquad (9.55)
\end{aligned}$$

$$\begin{aligned}
x \ =\ & -\frac{1}{2}\left\{\eta_1\eta_2^2 y^4 + R_0\left[R_0\eta_1(2+\eta_2^2) - 2\eta_2^4\tilde{x} + 2\eta_1\eta_2\tilde{z}(1+\eta_2^2)\right]y^2\right. \\
& \left. -2\tilde{y}\eta_1 R_0^2 y - R_0^2\eta_1\eta_2^2\tilde{y}^2\right\} \Big/ \left[R_0(\eta_1^2 y + \tilde{y}\eta_2^2)y\right], \qquad (9.56)
\end{aligned}$$

and

$$
\begin{aligned}
z &= \frac{1}{2}\left\{\eta_1^2\eta_2 y^4 + R_0\eta_1\left[2\eta_1\tilde{z}(1+\eta_2^2) + \eta_2(\eta_1 R_0 - 2\tilde{x}\eta_2^2)\right]y^2\right. \\
&\quad \left. -2\eta_2 R_0^2\tilde{y}y + R_0^2\eta_2\tilde{y}^2(1+\eta_2^2)\right\} / \left[R_0(\eta_1^2 y + \tilde{y}\eta_2^2)y\right], \qquad (9.57)
\end{aligned}
$$

where $\eta_1 = \cos\phi_0$ and $\eta_2 = \sin\phi_0$. Note that the left side of (9.55) is an eighth-order polynomial, and hence has eight zeros. We pick the root that is closest to $\tilde{y}$ as the solution for y.

We now present two examples. First, let $R_0 = 15$, $\phi_0 = \pi/5$, and $(\tilde{x}, \tilde{y}, \tilde{z}) = (0.5, 0.4, 0.3)$. We obtain $(x, y, z) = (0.5005, 0.3849, 0.3078)$. (We remark that the other seven possible solutions for y are either complex or far away from $\tilde{y}$.) Next, let $R_0 = 300$, $\phi_0 = \pi/4$, and $(\tilde{x}, \tilde{y}, \tilde{z}) = (5, 4, 2)$. We obtain $(x, y, z) = (4.9991, 3.9346, 2.0477)$. Note that in both examples, $(\tilde{x}, \tilde{y}, \tilde{z})$ are very close to (x, y, z) and hence can be used to approximate (x, y, z).

Reference

[1] J. Li, Z.-S. Liu, and P. Stoica, "3-D target feature extraction via interferometric SAR," *IEE Proceedings on Radar, Sonar and Navigation*, vol. 144, April 1997.

[2] L. C. Graham, "Synthetic interferometer radar for topographic mapping," *Proceedings of the IEEE*, vol. 62, pp. 763–768, June 1974.

[3] R. Gens and J. L. V. Gendren, "SAR interferometry - issues, techniques, applications," *International Journal of Remote Sensing*, vol. 17, pp. 1803–1835, July 1996.

[4] K. Knaell, "Three-dimensional SAR from curvilinear apertures," *Proceedings of the 1996 IEEE National Radar Conference*, pp. 220–225, Ann Arbor, MI, May 1996.

[5] K. Knaell, "Three-dimensional SAR from practical apertures," *SPIE Proceedings on Optical Engineering in Aerospace Sensing, San Diego*, vol. 2562, pp. 31–41, 1995.

[6] K. Knaell, "Three-dimensional SAR from curvilinear apertures," *SPIE Proceedings on Optical Engineering in Aerospace Sensing*, vol. 2230, pp. 120–134, Orlando, FL, April 1994.

[7] K. K. Knaell and G. P. Cardillo, "Radar tomography for the generation of three-dimensional images," *IEE Proceedings on Radar, Sonar and Navigation*, vol. 142, pp. 54–60, April 1995.

[8] J. A. Högbom, "Aperture synthesis with a non-regular distribution of interferometer baselines," *Astronomy and Astrophysics Supplements*, vol. 15, pp. 417–426, 1974.

[9] J. Tsao and B. D. Steinberg, "Reduction of sidelobe and speckle artifacts in microwave imaging: The CLEAN technique," *IEEE Transactions on Antennas and Propagation*, vol. 36, pp. 543–556, April 1988.

[10] P. T. Gough, "A fast spectral estimation algorithm based on the FFT," *IEEE Transactions on Signal Processing*, vol. 42, pp. 1317–1322, June 1994.

[11] J. Li and P. Stoica, "Efficient mixed-spectrum estimation with applications to target feature extraction," *IEEE Transactions on Signal Processing*, vol. 44, pp. 281–295, February 1996.

[12] D. C. Munson, Jr., J. D. O'Brien, and W. K. Jenkins, "A tomographic formulation of spotlight-mode synthetic aperture radar," *Proceedings of the IEEE*, vol. 71, pp. 917 –925, August 1983.

[13] C. V. Jakowatz, Jr., D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. A. Thompson, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach.* Norwell, MA: Kluwer Academic Publishers, 1996.

[14] G. H. Golub and C. F. Van Loan, *Matrix Computations.* Baltimore, MD: Johns Hopkins University Press, 1984.

[15] W. Bangs, *Array processing with generalized beamformers.* Ph.D. dissertation, Yale University, New Haven, CT, 1971.

[16] P. Stoica and A. Nehorai, "Performance study of conditional and unconditional direction-of-arrival estimation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-38, pp. 1783–1795, October 1990.

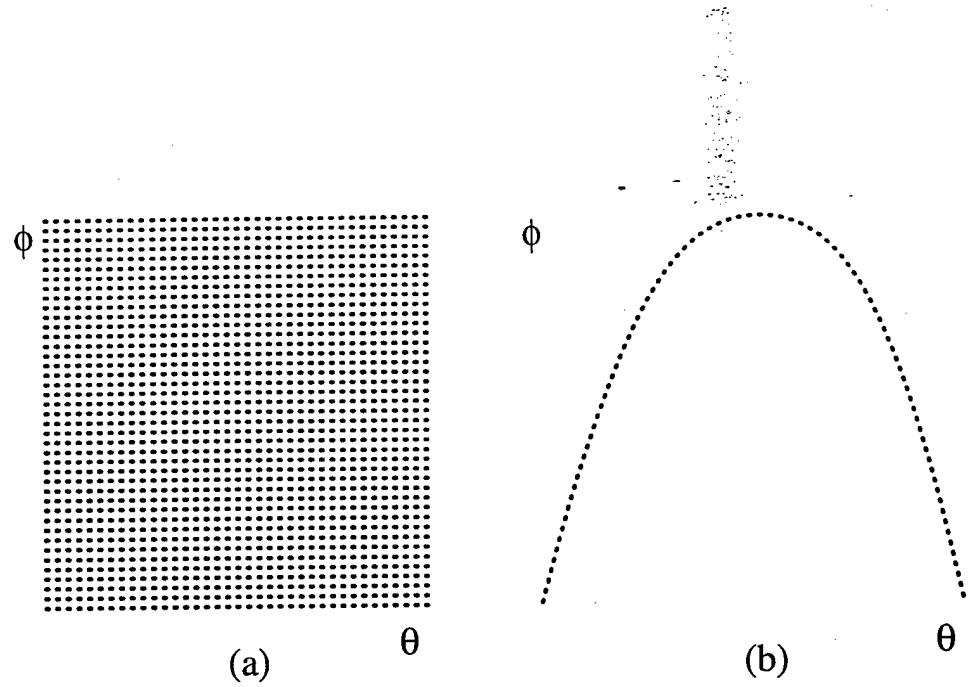


Figure 9.1: Possible apertures for a 3-D SAR system. a) Full aperture. b) Parabolic aperture.

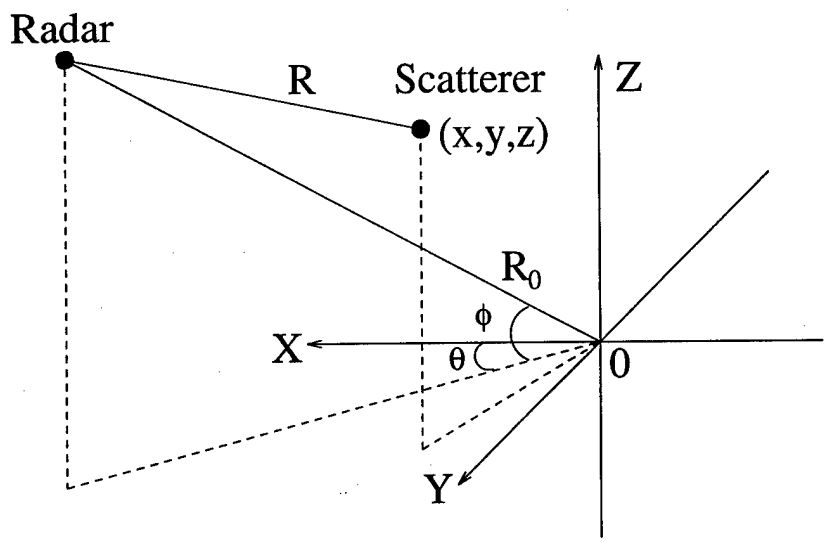


Figure 9.2: Data collection geometry.

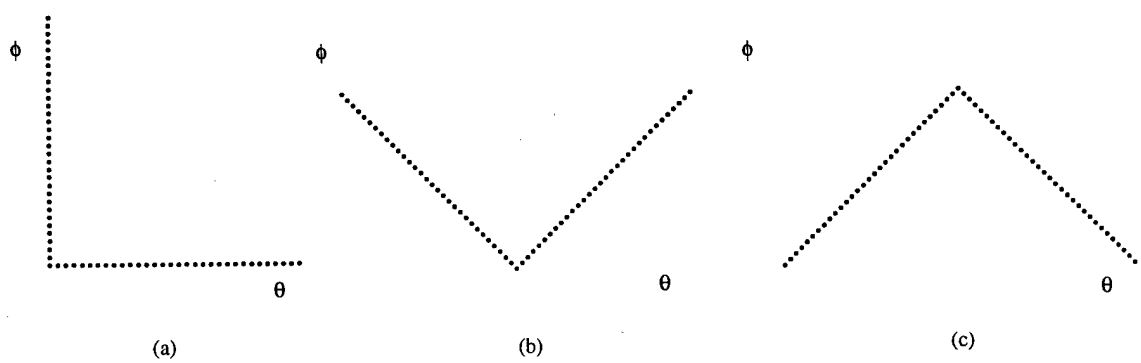
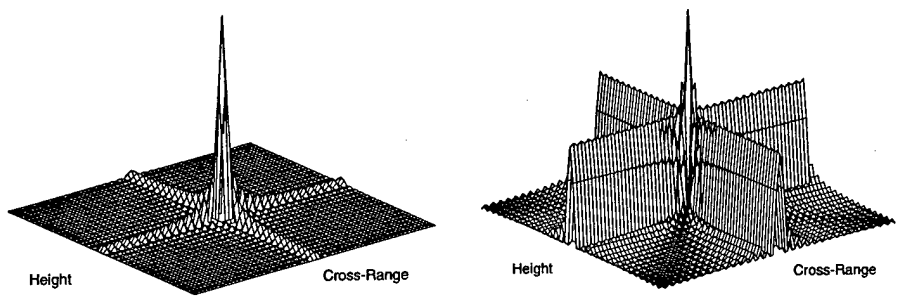


Figure 9.3: Apertures consisting of orthogonal subapertures.

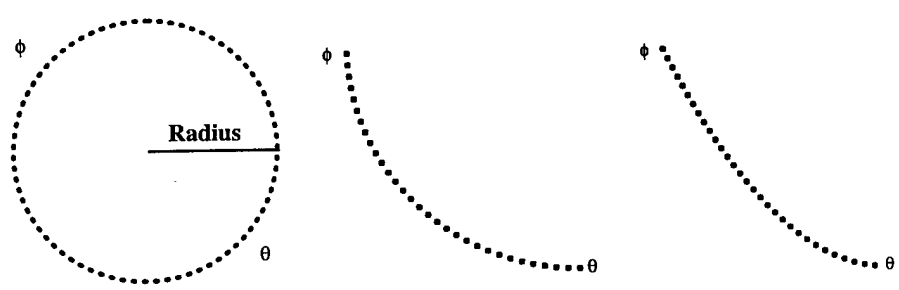
Aperture	CRB(α)	CRB(ω)	CRB($\tilde{\omega}$)	CRB($\bar{\omega}$)
Full	-5.9875	-51.4511	-51.4511	-51.4511
Circular	5.3663	-39.3415	-40.8469	-40.8788
Parabolic	5.8693	-39.3415	-40.4915	-39.6521
L-shaped	4.8584	-39.3415	-38.2414	-38.2414
Arc-1	11.7482	-39.3415	-31.7272	-31.7272
Acr-2	17.2473	-39.3415	-26.8015	-27.4636

Table 9.1: Comparison of the CRBs (in dB) of the target features for the cases of the full, circular, parabolic, L-shaped, Arc-1, and Arc-2 apertures when $M = 63$, $K = 1$, $\omega = \tilde{\omega} = \bar{\omega} = 0$, $\alpha = 1$, and $\sigma^2 = 40$.



(a) (b)

Figure 9.4: Mesh plots of the modulus of the RCS obtained by using 2-D FFT with different apertures (range information suppressed). (a) Full aperture as shown in Figure 9.1(a). (b) Curved aperture as shown in Figure 9.3(a).



(a) (b) (c)

Figure 9.5: Curvilinear apertures. a) The circular aperture. b) The Arc-1 aperture. c) The Arc-2 aperture.

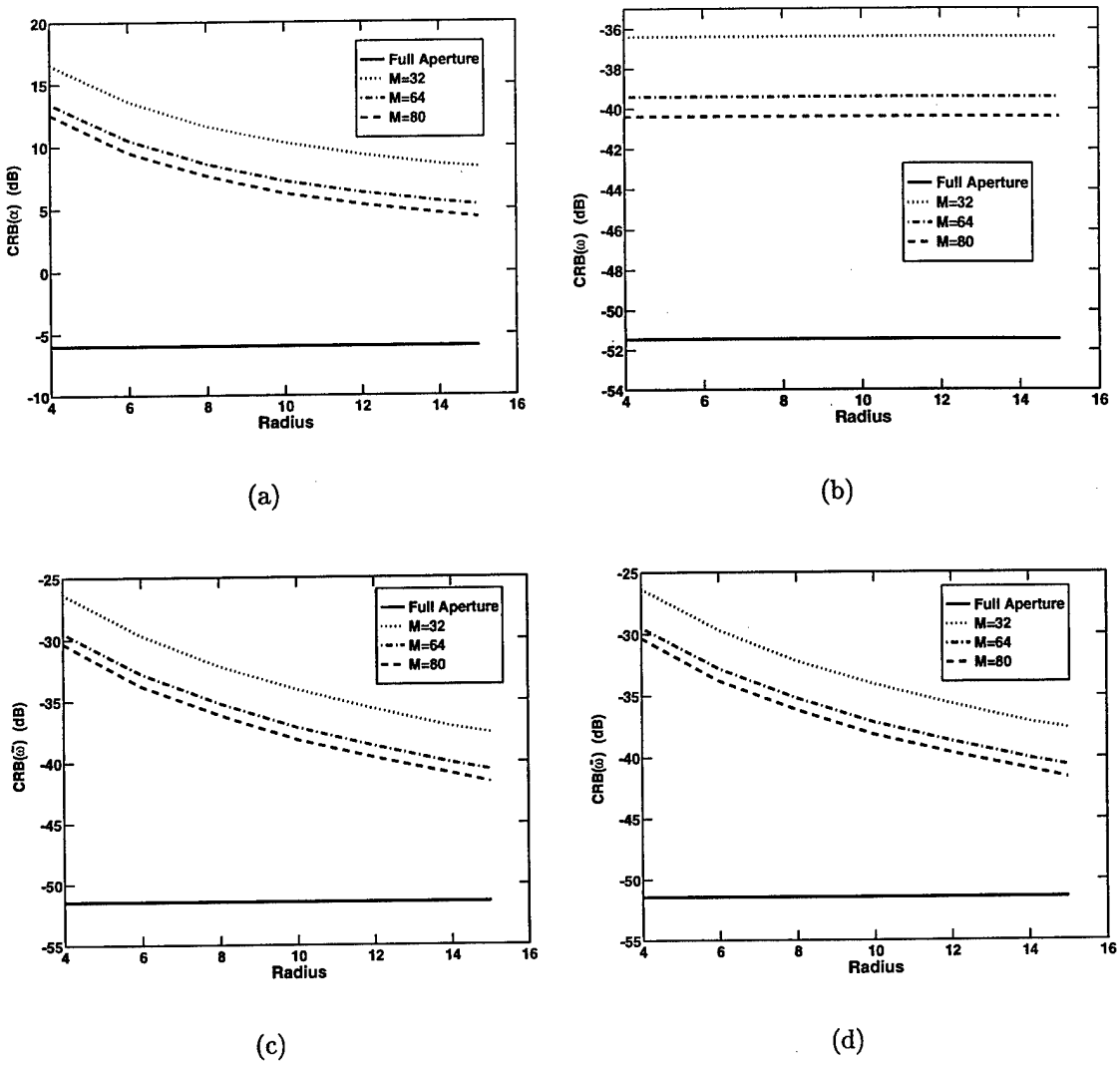
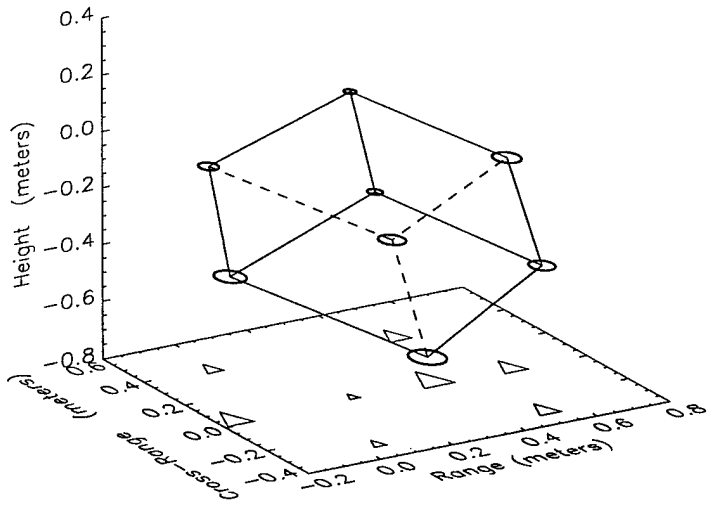
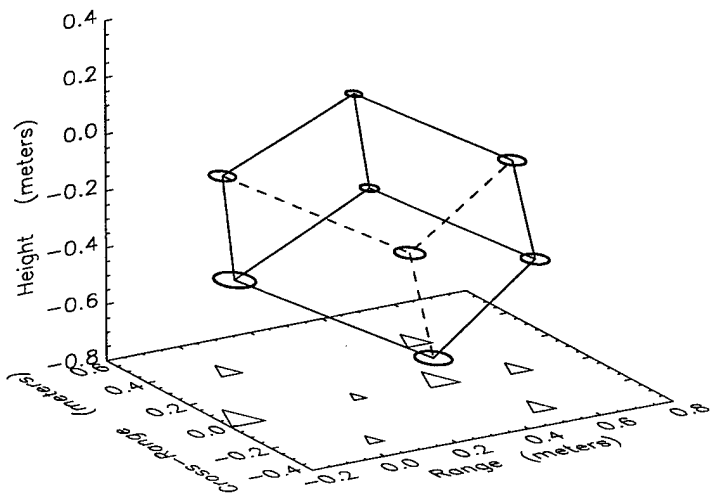


Figure 9.6: The CRBs of the target parameters as a function of the circular aperture radius when $K = 1$, $\alpha = 1$, $\omega = \tilde{\omega} = \bar{\omega} = 0$, and $\sigma^2 = 40$. a) The CRB of the complex amplitude α. b) The CRB of ω. c) The CRB of $\tilde{\omega}$. d) The CRB of $\bar{\omega}$.



(a)



(b)

Figure 9.7: 3-D plots of $K = 8$ scatterers extracted by using RELAX with the indoor experimental data. (a) Obtained with full aperture as shown in Figure 9.1(a). (b) Obtained with curved aperture as shown in Figure 9.3(a).

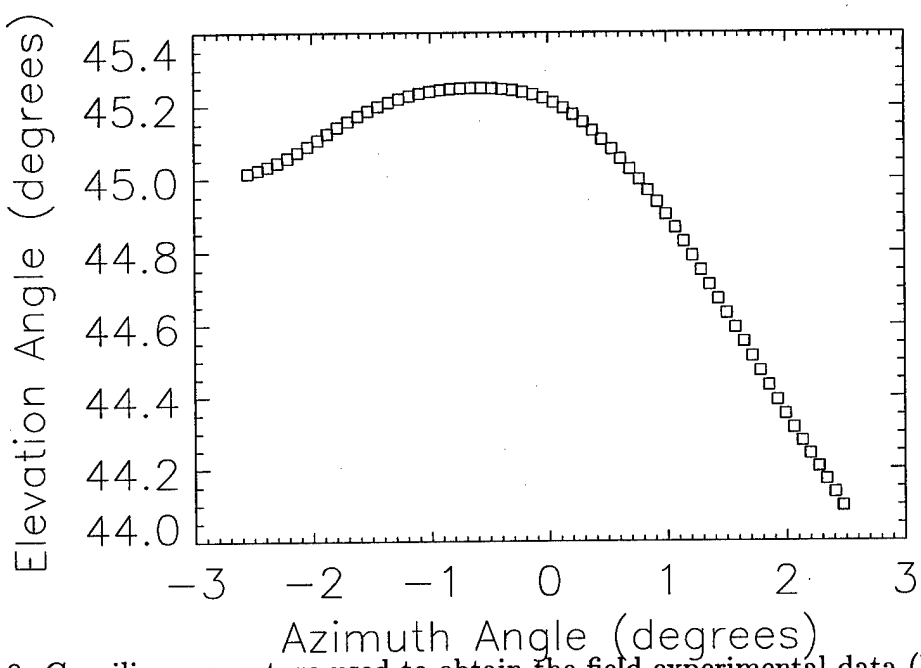
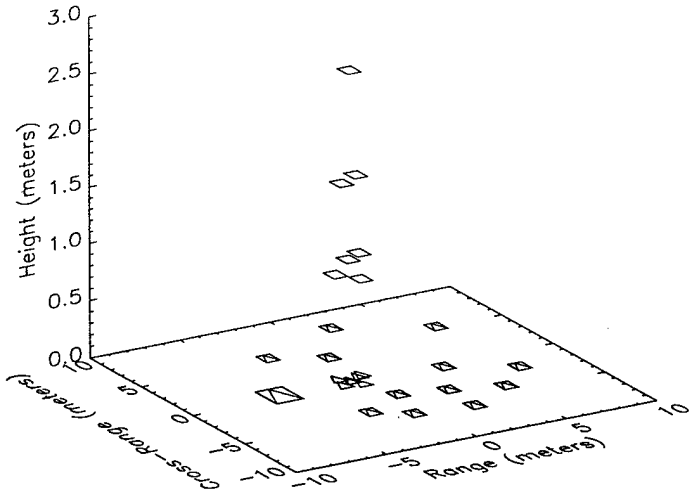
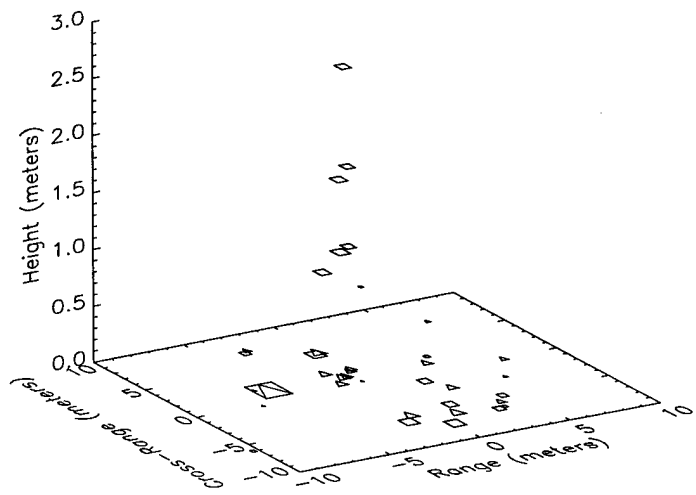


Figure 9.8: Curvilinear aperture used to obtain the field experimental data (DSMS data).



(a)



(b)

Figure 9.9: 3-D plots of $K = 20$ scatterers extracted by using RELAX with the field experimental data (DSMS data). (a) True scatterer distribution. (b) Obtained with RELAX.

10. Autofocus and Feature Extraction in Curvilinear SAR

10.1 Introduction

Synthetic aperture radar (SAR) can be used to form images and extract target features with high resolution in range, cross-range, and elevation directions. Strip-map and spotlight-mode SAR systems are two commonly used systems. The former is often used for large area terrain imaging and feature extraction. The latter is used to image a relatively small patch of the earth with high resolution. Both the strip-map and spotlight-mode SAR provide only two-dimensional (2-D) target features including the radar cross section (RCS) and 2-D location (range and cross-range) for each scatterer. Several other SAR systems can be used to obtain three-dimensional (3-D) target features, which include the radar cross section (RCS), the 2-D location, and the height for each scatterer. An interferometric synthetic aperture radar (IFSAR) system [1, 2, 3], which uses a pair of antennas displaced in the cross-track plane to obtain two coherent and parallel measurement apertures, can be used for both 2-D and 3-D SAR imaging and 3-D target feature extraction. However, the IFSAR system suffers from ambiguity problems since it provides only two vertical parallel apertures. For example, the system cannot resolve more than one target scatterer at the same projected range and cross-range but at different heights [3]. A curvilinear synthetic aperture radar (CLSAR) system [4, 5, 6, 7] can be used with spectral estimation methods to extract the 3-D target features and avoid the ambiguity problems suffered by the IFSAR system.

For a practical SAR system, however, aperture errors exist due to atmospheric turbulence and platform position uncertainty. These errors can significantly degrade the SAR image quality and the estimation accuracy of target parameters. Hence SAR imaging and target feature extraction algorithms must be combined with effective motion compensation methods to obtain the best results in practice.

Many algorithms, such as the phase-gradient autofocus (PGA) algorithm [8, 9, 10] and the motion compensation RELAX (MCRELAX) algorithm [11], have been proposed to compensate for the unknown motion errors in 2-D SAR imaging by assuming that the motion errors

mainly result in phase errors in the cross-range direction. The purpose of this chapter is to address the autofocus problem in CLSAR. The problem herein is much more complicated since the aperture errors can no longer be approximated as phase errors in the cross-range direction.

In this chapter, we present a relaxation-based autofocus method to compensate for the curvilinear aperture errors in CLSAR. This method is referred to as the autofocus RELAX or AUTORELAX algorithm. The AUTORELAX algorithm can be used to compensate for the curvilinear aperture errors and extract 3-D target features. This chapter is an extension of [7], in which we described how to extract the 3-D target features via CLSAR with a relaxation-based algorithm without considering the existence of the curvilinear aperture errors.

The remainder of this chapter is organized as follows. In Section 10.2, the data model is given and the problem of interest is formulated. In Section 10.3, we present the relaxation-based autofocus algorithm. Section 10.4 shows the experimental and numerical examples illustrating the performance of AUTORELAX. Finally, Section 10.5 contains our conclusions.

10.2 Data Model and Problem Formulation

10.2.1 High Range Resolution Radar

We first describe how one can obtain 1-D target features via a high range resolution radar as a preparation for the analysis in the following subsections. The range resolution of a radar is determined by the radar bandwidth. To achieve high resolution in range, the radar must transmit wideband pulses, which are often linear frequency modulated (FM) chirp pulses [12, 13] . A normalized chirp pulse can be written as

$$s(t) = e^{-j(2\pi f_0 t + \gamma t^2)}, \quad |t| \leq T_0/2, \tag{10.1}$$

where f_0 denotes the carrier frequency, 2γ denotes the FM rate, and T_0 denotes the width of the pulse. We assume that f_0, γ, and T_0 are known. The signal returned by a scatterer

of a target has the form

$$r(t) = \delta_\tau e^{-j[2\pi f_0(t-\tau)+\gamma(t-\tau)^2]}, \tag{10.2}$$

where δ_τ is determined by the RCS of the scatterer and τ denotes the round-trip time delay. The dechirped signal $\tilde{d}(t)$ is obtained by mixing $r(t)$ with $s^*(t-\tau_0)$ for a reference τ_0 (see Section 10.2.2), where $(\cdot)^*$ denotes the complex conjugate and

$$\tilde{d}(t) = \delta_\tau e^{j2(\pi f_0-\gamma\tau_0)(\tau-\tau_0)} e^{-j\gamma(\tau-\tau_0)^2} e^{j2\gamma(\tau-\tau_0)t}. \tag{10.3}$$

The term $e^{-j\gamma(\tau-\tau_0)^2}$ in (10.3) is usually close to a constant for all $\tau_{\min} \leq \tau \leq \tau_{\max}$, where $\tau_{\max}$ and $\tau_{\min}$ correspond to the maximum and minimum values, respectively, of the round-trip time delays between the scatterers of a target and the radar and $\tau_{\min} \leq \tau_0 \leq \tau_{\max}$. This term can also be partially removed [12]. Let $\tilde{D}(\omega)$ denote the Fourier transform of $\tilde{d}(t)$. Then the inverse Fourier transform of $\tilde{D}(\omega)e^{j\frac{\omega^2}{4\gamma}}$ will have the term $e^{-j\gamma(\tau-\tau_0)^2}$ removed. Yet this removal can only be approximate since $\tilde{d}(t)$ is not known for all t and hence $\tilde{D}(\omega)$ is not known exactly. The closer $e^{-j\gamma(\tau-\tau_0)^2}$ is to a constant for $\tau_{\min} \leq \tau \leq \tau_{\max}$, the better its removal. With this removal, we have

$$d(t) = \delta_\tau e^{j2(\pi f_0-\gamma\tau_0)(\tau-\tau_0)} e^{j2\gamma(\tau-\tau_0)t}, \tag{10.4}$$

which is a complex sinusoid with frequency $2\gamma(\tau-\tau_0)$ and amplitude $\delta_\tau e^{j2(\pi f_0-\gamma\tau_0)(\tau-\tau_0)}$. We know $\tau_{\max}$ and $\tau_{\min}$ approximately since we assume that the altitude, antenna beamwidth, and grazing angle of the radar are known. We also assume that $(\tau_{\max}-\tau_{\min}) \ll T_0$. Then for $-T_0/2+\tau_{\max} \leq t \leq T_0/2+\tau_{\min}$, the scatterers of the target at different ranges correspond to different frequencies of the signal $d(t)$, while the RCS's of the scatterers are proportional to the complex amplitudes of the corresponding sinusoids. The ranges and RCS's of the target scatterers are the 1-D target features.

10.2.2 Curvilinear Synthetic Aperture Radar

We now describe the 3-D data model for the curvilinear synthetic aperture radar. The cross-range and height resolution of an ordinary ranging radar is limited by its antenna beamwidth. For an airborne or spaceborne system, a narrow antenna beamwidth requires

an antenna that may be too large to be carried on board of the airplane or the spacecraft. Curvilinear spotlight-mode SAR avoids this requirement by collecting coherent radar returns while viewing a target from many different azimuth and elevation angles. (A possible curvilinear aperture is shown in Figure 10.1.) By properly processing the return signals, we can also achieve high resolution in both cross-range and elevation.

A broadside data collection geometry in a spotlight-mode synthetic aperture radar (SAR) is shown in Figure 10.2 [13]. The XYZ coordinate system is centered on a small patch of ground, where a target is located. The ground is illuminated by a narrow radio frequency (RF) beam from the moving radar that rotates (with radius R_0) around the coordinate origin. In Figure 10.2, R denotes the distance between the radar and a scatterer at the position (x, y, z), and θ and ϕ are the azimuth and elevation angles of the radar relative to the XYZ coordinate system. We assume that θ, ϕ, and R_0 are known.

The range R of the scatterer located at (x, y, z) can be written as

$$R = \left[(R_0 \cos\theta \cos\phi - x)^2 + (R_0 \sin\theta \cos\phi - y)^2 + (R_0 \sin\phi - z)^2\right]^{1/2}. \tag{10.5}$$

Under the conditions $\frac{x}{R_0} \ll 1$, $\frac{y}{R_0} \ll 1$, and $\frac{z}{R_0} \ll 1$, we have

$$\begin{aligned} R \approx R_0 \Bigg[& 1 - \frac{x}{R_0}\cos\theta\cos\phi - \frac{y}{R_0}\sin\theta\cos\phi - \frac{z}{R_0}\sin\phi + \frac{x^2+y^2+z^2}{2R_0^2} \\ & - \frac{(x\cos\theta\cos\phi + y\sin\theta\cos\phi + z\ sin\phi)^2}{2R_0^2}\Bigg]. \end{aligned} \tag{10.6}$$

Let $\psi = \phi - \phi_0$, where ϕ_0 is the average of all ϕ used to form the synthetic aperture. For very small ψ, we have

$$\cos\phi \approx \cos\phi_0 - (\sin\phi_0)\psi, \tag{10.7}$$

$$\sin\phi \approx \sin\phi_0 + (\cos\phi_0)\psi. \tag{10.8}$$

For very small θ, we have $\cos\theta \approx 1$ and $\sin\theta \approx \theta$. Then keeping first-order terms yields

$$\begin{aligned} & x^2\sin^2\phi + x^2\sin^2\theta\cos^2\phi + y^2\sin^2\phi + y^2\cos^2\theta\cos^2\phi + z^2\cos^2\phi \\ \approx\ & [(x^2+y^2)\sin\phi_0]\sin\phi + [(y^2+z^2)\ cos\phi_0]\cos\theta\cos\phi + [(x^2-z^2)\sin\phi_0\cos\phi_0]\psi \end{aligned} \tag{10.9}$$

We also have

$$\begin{aligned} & 2xy\cos\theta\sin\theta\cos^2\phi + 2yz\sin\theta\ cos\phi\sin\phi + 2xz\cos\theta\cos\phi\sin\phi \\ \approx\ & 2(xy\cos\phi_0 + yz\sin\phi_0)\sin\theta\cos\phi + (2xz\sin\phi_0)\cos\theta\cos\phi + (2xz\cos^2\phi_0)\psi \end{aligned} \tag{10.10}$$

Hence R can be simplified as

$$R \approx \tilde{R}_0 - \tilde{x}\cos\theta\cos\phi - \tilde{y}\sin\theta\cos\phi - \tilde{z}\sin\phi, \tag{10.11}$$

where

$$\tilde{R}_0 = R_0 + \frac{[(x^2 - z^2)\sin\phi_0\cos\phi_0 - 2xz\cos^2\phi_0](\phi - \phi_0)}{2R_0} \tag{10.12}$$

$$\tilde{x} = x + \frac{-(y^2 + z^2)\cos\phi_0 + 2xz\sin\phi_0}{2R_0}, \tag{10.13}$$

$$\tilde{y} = y + \frac{xy\cos\phi_0 + yz\sin\phi_0}{R_0}, \tag{10.14}$$

and

$$\tilde{z} = z - \frac{(x^2 + y^2)\sin\phi_0}{2R_0}. \tag{10.15}$$

Note that the second terms of the right sides of (10.12), (10.13), (10.14), and (10.15) are due to the range and elevation curvature effects and can be neglected for large R_0.

Let $\tau_0 = \frac{2R_0}{c}$. Since $\tau = \frac{2R}{c}$, then from (10.4), we have

$$d(t,\theta,\phi) = \delta_{x,y,z}\exp\left[\frac{j4(\pi f_0 - \gamma\tau_0 + \gamma t)}{c}(\tilde{R}_0 - R_0 - \tilde{x}\cos\theta\cos\phi - \tilde{y}\sin\theta\cos\phi - \tilde{z}\sin\phi)\right], \tag{10.16}$$

where $\delta_{x,y,z}$ is proportional to the RCS of the scatterer located at (x, y, z). For $4(\pi f_0 - \gamma\tau_0 + \gamma t)(\tilde{R}_0 - R_0)/c \ll 2\pi$, where $|t - \tau_0| < T_0/2$, we can write (10.16) as

$$d(t,\theta,\phi) \approx \delta_{x,y,z}e^{j(\tilde{x}t_x + \tilde{y}t_y + \tilde{z}t_z)}, \tag{10.17}$$

where

$$t_x = -\frac{4(\pi f_0 - \gamma\tau_0 + \gamma t)\cos\theta\cos\phi}{c}, \tag{10.18}$$

$$t_y = -\frac{4(\pi f_0 - \gamma\tau_0 + \gamma t)\sin\theta\cos\phi}{c}, \tag{10.19}$$

and

$$t_z = -\frac{4(\pi f_0 - \gamma\tau_0 + \gamma t)\sin\phi}{c}. \tag{10.20}$$

Note that $d(t, \theta, \phi)$ is a 3-D complex sinusoid. The frequencies of the 3-D sinusoid correspond to the 3-D location $(\tilde{x}, \tilde{y}, \tilde{z})$ of the scatterer, while the amplitude is proportional to its RCS. Note also that $(\tilde{x}, \tilde{y}, \tilde{z})$ is not the true location (x, y, z) of the scatterer, but is close to (x, y, z) for large R_0 [7]. When a target has multiple scatterers, $d(t, \theta, \phi)$ in (10.17) will be a sum of sinusoids. The 3-D locations and RCS's of the target scatterers are the 3-D target features. (Usually the samples on the t, θ, and ϕ axes are uniformly spaced and hence the samples of t_x, t_y, and t_z occur at the points of a polar grid.)

Assume that a curvilinear aperture consists of M different viewing angles and let $\{\phi_m, \theta_m\}_{m=1}^{M}$ denote the elevation and azimuth angle pairs of the M look angles of the radar. Let $y(n, m)$, $n = 0, 1, \cdots, N-1$, denote the one-dimensional (1-D) data samples obtained after dechirping from the mth viewing angle of the radar. Let

$$\breve{t}_n = -\frac{4(\pi f_0 - \gamma\tau_0 + \gamma t_n)}{c}, \quad n = 0, 1, \cdots, N-1, \tag{10.21}$$

where t_n denotes the time samples. Let $t_{x_m}(n)$, $t_{y_m}(n)$, and $t_{z_m}(n)$ denote the time samples of the mth look angle, where

$$t_{x_m}(n) = \breve{t}_n \cos(\theta_m)\cos(\phi_m), \tag{10.22}$$

$$t_{y_m}(n) = \breve{t}_n \sin(\theta_m)\cos(\phi_m), \tag{10.23}$$

and

$$t_{z_m}(n) = \breve{t}_n \sin(\phi_m), \tag{10.24}$$

with $m = 1, 2, \cdots, M$ and $n = 0, 1, \cdots, N-1$. Then $y(n, m)$ has the form

$$y(n, m) = z(n, m) + e(n, m), \tag{10.25}$$

where $e(n, m)$ denotes the noise and clutter and

$$z(n, m) = \sum_{k=1}^{K} \alpha_k e^{j2\pi[f_k t_{x_m}(n) + \tilde{f}_k t_{y_m}(n) + \bar{f}_k t_{z_m}(n)]}, \tag{10.26}$$

with K being the number of scatterers.

10.2.3 Aperture Errors

For the above CLSAR data model, we have assumed that $\{\theta_m, \phi_m\}_{m=1}^M$ and $R_0(m) = R_0$, $m = 1, 2, \cdots, M$, are known exactly. For a practical curvilinear SAR system, however, the radar positions relative to the XYZ coordinate system may not be known exactly due to atmospheric turbulence and platform position uncertainty. In 2-D SAR imaging, it is generally assumed that the errors in $\{\theta_m, \phi_m\}_{m=1}^M$ are negligible and the errors in $\{R_0(m)\}_{m=1}^M$ cause phase errors along the synthetic aperture. In CLSAR, however, the errors in $\{\theta_m, \phi_m\}_{m=1}^M$ may no longer be negligible since the aperture shape is critical for 3-D target feature extraction.

Our problem of interest herein is to compensate for the curvilinear aperture errors in $\{R_0(m)\}_{m=1}^M$ and $\{\theta_m, \phi_m\}_{m=1}^M$ and extract the 3-D target parameters $\{\alpha_k, f_k, \tilde{f}_k, \bar{f}_k\}_{k=1}^K$ from $\{y(n,m)\}$, $n = 0, 1, \cdots, N-1$, $m = 1, 2, \cdots, M$.

Before we present the AUTORELAX algorithm, let us first consider the approximations and the ambiguity problems in our data model in the presence of aperture errors. For the broadside data collection geometry shown in Figure 10.2, θ_m is very small. For very small θ_m, we have $\sin(\theta_m) \approx \theta_m$ and $\cos(\theta_m) \approx 1$. These approximations also hold for the true look angles of the radar. Then according to (10.22), (10.23), and (10.24), respectively, we have

$$\overset{\circ}{t}_{x_m} \approx \breve{t}(\cos \overset{\circ}{\phi}_m), \tag{10.27}$$

$$\overset{\circ}{t}_{y_m} \approx \breve{t}(\cos \overset{\circ}{\phi}_m)\, \overset{\circ}{\theta}_m, \tag{10.28}$$

and

$$\overset{\circ}{t}_{z_m} \approx \breve{t}(\sin \overset{\circ}{\phi}_m), \tag{10.29}$$

where $\{\overset{\circ}{\theta}_m, \overset{\circ}{\phi}_m\}_{m=1}^M$ denote the true look angles of the radar and

$$\breve{t} = -\frac{4(\pi f_0 - \gamma\tau_0 + \gamma t)}{c}.$$

Consider first the errors in $\{\phi_m\}_{m=1}^{M}$ when $\theta_m = \overset{\circ}{\theta}_m$ and $R_0(m) = R_0$, $m = 1, 2, \cdots, M$. Let

$$\phi_m = \overset{\circ}{\phi}_m + \Delta\phi_m. \tag{10.30}$$

Let

$$\breve{t}_0 = \frac{-4\pi f_0}{c}. \tag{10.31}$$

Then for very small $\Delta\phi_m$ and $\pi f_0 \gg \gamma(t - \tau_0)$, where $|t - \tau_0| < T_0/2$, we have

$$\begin{aligned} t_{x_m} &= \breve{t}(\cos\phi_m) \\ &\approx \breve{t}\cos\overset{\circ}{\phi}_m - \breve{t}(\sin\overset{\circ}{\phi}_m)\Delta\phi_m \\ &\approx \breve{t}\cos\overset{\circ}{\phi}_m - \breve{t}_0(\sin\overset{\circ}{\phi}_0)\Delta\phi_m, \end{aligned} \tag{10.32}$$

$$\begin{aligned} t_{y_m} &= \breve{t}(\cos\phi_m)\,\overset{\circ}{\theta}_m \\ &\approx \breve{t}(\cos\overset{\circ}{\phi}_m)\,\overset{\circ}{\theta}_m \\ &\approx \breve{t}_0(\cos\overset{\circ}{\phi}_0)\,\overset{\circ}{\theta}_m, \end{aligned} \tag{10.33}$$

and

$$\begin{aligned} t_{z_m} &= \breve{t}(\sin\phi_m) \\ &\approx \breve{t}\sin\overset{\circ}{\phi}_m + \breve{t}(\cos\overset{\circ}{\phi}_m)\Delta\phi_m \\ &\approx \breve{t}\sin\overset{\circ}{\phi}_m + \breve{t}_0(\cos\overset{\circ}{\phi}_0)\Delta\phi_m, \end{aligned} \tag{10.34}$$

where $\overset{\circ}{\phi}_0$ is the average of all $\overset{\circ}{\phi}_m$, $m = 1, 2, \cdots, M$, and hence is a constant. Then

$$\begin{aligned} \alpha_k \exp\left\{j2\pi(f_k t_{x_m} + \tilde{f}_k t_{y_m} + \bar{f}_k t_{z_m})\right\} &\approx \alpha_k \exp\left\{j2\pi\left[f_k\breve{t}\cos\overset{\circ}{\phi}_m + \tilde{f}_k\breve{t}_0\cos\overset{\circ}{\phi}_0\overset{\circ}{\theta}_m + \bar{f}_k\breve{t}\sin\overset{\circ}{\phi}_m\right]\right\} \\ &\quad \exp\left\{j2\pi\left[-f_k\breve{t}_0\sin\overset{\circ}{\phi}_0 + \bar{f}_k\breve{t}_0\cos\overset{\circ}{\phi}_0\right]\Delta\phi_m\right\}. \end{aligned} \tag{10.35}$$

Equation (10.35) shows that if $\Delta\phi_m$ is a constant, then the phase error due to $\Delta\phi_m$ and the phase of α_k are ambiguous. Hence the phase of α_k can never be determined exactly in the presence of $\Delta\phi_m$. If $\Delta\phi_m$ is a linear function of $\overset{\circ}{\theta}_m$, then $\tilde{f}_k$ and the linear phase error due to $\Delta\phi_m$ are ambiguous and cannot be determined exactly.

Similarly, we can analyze the errors in $\{\theta_m\}_{m=1}^{M}$ when $\phi_m = \overset{\circ}{\phi}_m$ and $R_0(m) = R_0$, $m = 1, 2, \cdots, M$. Let

$$\theta_m = \overset{\circ}{\theta}_m + \Delta\theta_m. \tag{10.36}$$

For very small $\overset{\circ}{\theta}_m$ and $\Delta\theta_m$ and for $\pi f_0 \gg \gamma(t - \tau_0)$, where $|t - \tau_0| < T_0/2$, we have

$$t_{x_m} = \breve{t} \cos \overset{\circ}{\phi}_m, \tag{10.37}$$

$$t_{y_m} \approx \breve{t}_0 \cos \overset{\circ}{\phi}_0 \, (\overset{\circ}{\theta}_m + \Delta\theta_m), \tag{10.38}$$

and

$$t_{x_m} = \breve{t} \sin \overset{\circ}{\phi}_m \, . \tag{10.39}$$

Then

$$\begin{aligned} \alpha_k \exp\left\{j2\pi(f_k t_{x_m} + \tilde{f}_k t_{y_m} + \bar{f}_k t_{z_m})\right\} \approx \; & \alpha_k \exp\left\{j2\pi\left[f_k \breve{t} \cos \overset{\circ}{\phi}_m + \tilde{f}_k \breve{t}_0 \cos \overset{\circ}{\phi}_0 \overset{\circ}{\theta}_m + \bar{f}_k \breve{t} \sin \overset{\circ}{\phi}_m\right]\right\} \\ & \exp\left\{j2\pi \tilde{f}_k \breve{t}_0 \cos \overset{\circ}{\phi}_0 \, \Delta\theta_m\right\}, \end{aligned} \tag{10.40}$$

which shows that a constant $\Delta\theta_m$ also results in the ambiguity between the phase error due to $\Delta\theta_m$ and the phase of α_k. Hence the phase of α_k can never be determined exactly in the presence of $\Delta\theta_m$. Also, if $\Delta\theta_m$ is a linear function of $\overset{\circ}{\theta}_m$, then $\tilde{f}_k$ and the linear phase error due to $\Delta\theta_m$ are ambiguous and cannot be determined exactly.

Finally, consider the errors in $R_0(m)$ when $\theta_m = \overset{\circ}{\theta}_m$ and $\phi_m = \overset{\circ}{\phi}_m$, $m = 1, 2, \cdots, M$. Let

$$R_0(m) = R_0 + \Delta R_0(m). \tag{10.41}$$

Replacing R_0 in (10.12) with $R_0(m)$ and for small $\Delta R_0(m)$, large $R_0(m)$, and $\pi f_0 \gg \gamma(t - \tau_0)$, where $|t - \tau_0| < T_0/2$, we have from (10.16)

$$d(t, \theta, \phi) \approx \delta_{x,y,z} e^{j(\tilde{x} t_x + \tilde{y} t_y + \tilde{z} t_z)} e^{j\breve{t}_0 \Delta R_0(m)}. \tag{10.42}$$

Hence the errors in $\{R_0(m)\}_{m=1}^{M}$ result in phase errors along the synthetic aperture, which is consistent with the analysis in [13, 11]. Then the $z(n, m)$ in (10.25) should, for this case,

be replaced by

$$x(n,m) = \left\{ \sum_{k=1}^{K} \alpha_k e^{j2\pi[f_k \overset{\circ}{t}_{x_m}(n) + \tilde{f}_k \overset{\circ}{t}_{y_m}(n) + \bar{f}_k \overset{\circ}{t}_{z_m}(n)]} \right\} e^{j\eta_m}, \tag{10.43}$$

where $\eta_m = \breve{t}_0 \Delta R_0(m)$ is the phase error caused by $\Delta R_0(m)$. Note that this phase error differs from the phase errors in (10.35) and (10.40) since it does not depend on the parameters of the kth scatterer and is easier to deal with. If $\Delta R_0(m)$ is a constant, then the phase error due to $\Delta R_0(m)$ and the phase of α_k are ambiguous. Hence the phase of α_k can never be determined exactly in the presence of $\Delta R_0(m)$. If $\Delta R_0(m)$ is a linear function of $\overset{\circ}{\theta}_m$, then $\tilde{f}_k$ and the linear phase error due to $\Delta R_0(m)$ is ambiguous and cannot be determined exactly. We again assume that such linear errors in $\Delta R_0(m)$ are negligible in practice.

10.3 The Relaxation-Based Autofocus Algorithm

The AUTORELAX algorithm obtains the estimates $\{\hat{\theta}_m, \hat{\phi}_m, \hat{\eta}_m\}_{m=1}^{M}$ and $\{\hat{\alpha}_k, \hat{f}_k, \hat{\tilde{f}}_k, \hat{\bar{f}}_k\}_{k=1}^{K}$, respectively, of the true values $\{\overset{\circ}{\theta}_m, \overset{\circ}{\phi}_m, \eta_m\}_{m=1}^{M}$ and $\{\alpha_k, f_k, \tilde{f}_k, \bar{f}_k\}_{k=1}^{K}$ by minimizing the following nonlinear least squares (NLS) criterion:

$$C\left(\left\{\alpha_k, f_k, \tilde{f}_k, \bar{f}_k\right\}_{k=1}^{K}, \{\theta_m, \phi_m, \eta_m\}_{m=1}^{M}\right) = \sum_{m=1}^{M} \sum_{n=0}^{N-1} |y(n,m) - x(n,m)|^2, \tag{10.44}$$

where

$$x(n,m) = \left\{ \sum_{k=1}^{K} \alpha_k e^{j2\pi[f_k t_{x_m}(n) + \tilde{f}_k t_{y_m}(n) + \bar{f}_k t_{z_m}(n)]} \right\} e^{j\eta_m}, \tag{10.45}$$

with $t_{x_m}(n)$, $t_{y_m}(n)$, and $t_{z_m}(n)$ defined in (10.22), (10.23), and (10.24), respectively. Note that as shown below, we will determine θ_m and ϕ_m via a search method. Hence there is no need to use the approximations such as those in (10.32), (10.33), and (10.34) for $t_{x_m}(n)$, $t_{y_m}(n)$, and $t_{z_m}(n)$, respectively. When the noise $e(n,m)$ is a zero-mean white Gaussian random process, the NLS estimates of the unknown parameters coincide with the maximum likelihood (ML) estimates of the parameters. When the noise is colored, the NLS estimates are no longer the ML estimates, but they still possess excellent statistical performance [14]. The minimization of (10.44) is a very complicated optimization problem. AUTORELAX is a relaxation-based optimization method that can be used to minimize (10.44). Before we present the AUTORELAX algorithm, let us consider the following preparations.

Assume first that the target feature estimates $\{\hat{\alpha}_k, \hat{f}_k, \hat{\tilde{f}}_k, \hat{\bar{f}}_k\}_{k=1}^K$ are given. Then $\hat{\theta}_m$, $\hat{\phi}_m$, and $\hat{\eta}_m$, $m = 1, 2, \cdots, M$, can be determined by minimizing the following C_m with respect to θ_m, ϕ_m, and η_m, $m = 1, 2, \cdots, M$, where

$$C_m(\theta_m, \phi_m, \eta_m) = \sum_{n=0}^{N-1} |y(n,m) - \hat{x}_1(n,m)|^2, \tag{10.46}$$

where $\hat{x}_1(n,m)$ has the same form as $x(n,m)$ in (10.45) except that $\{\alpha_k, f_k, \tilde{f}_k, \bar{f}_k\}_{k=1}^K$ are replaced by $\{\hat{\alpha}_k, \hat{f}_k, \hat{\tilde{f}}_k, \hat{\bar{f}}_k\}_{k=1}^K$. To simplify the optimization of C_m, we determine $\{\hat{\theta}_m, \hat{\phi}_m\}_{m=1}^M$ and $\{\hat{\eta}_m\}_{m=1}^M$ iteratively as follows:

Step (1): Obtain $\hat{\theta}_m$ and $\hat{\phi}_m$, $m = 1, 2, \cdots, M$, by minimizing the following C_{m_1} with respect to θ_m and ϕ_m, where

$$C_{m_1}(\theta_m, \phi_m) = \sum_{n=0}^{N-1} |y(n,m) - \hat{z}_1(n,m)|^2, \tag{10.47}$$

where $\hat{z}_1(n,m)$ has the same form as $z(n,m)$ in (10.26) except that $\{\alpha_k, f_k, \tilde{f}_k, \bar{f}_k\}_{k=1}^K$ are replaced by $\{\hat{\alpha}_k, \hat{f}_k, \hat{\tilde{f}}_k, \hat{\bar{f}}_k\}_{k=1}^K$.

When there are errors in both the elevation and azimuth directions, we can estimate $\overset{\circ}{\phi}_m$ and $\overset{\circ}{\theta}_m$ by the alternating minimization approach, i.e., by iteratively fixing the estimate $\hat{\phi}_m$ of $\overset{\circ}{\phi}_m$ and minimizing C_{m_1} with respect to θ_m, and then fixing the estimate $\hat{\theta}_m$ of $\overset{\circ}{\theta}_m$ and minimizing C_{m_1} with respect to ϕ_m until "practical convergence". The "practical convergence" in the alternating minimization approach is determined by the relative change of the cost function C_{m_1}. In the numerical examples, we terminate the iteration when the relative change of the cost function C_{m_1} between two consecutive iterations is less than 10^{-3}. When there are errors in only one angular direction, i.e., either $\hat{\theta}_m$ or $\hat{\phi}_m$ is to be determined, then the minimization of C_{m_1} is a simple one-dimensional search problem.

Step (2): Determine $\{\hat{\eta}_m\}_{m=1}^M$ by minimizing the following cost functions C_{m_2}:

$$C_{m_2}(\eta_m) = \sum_{n=0}^{N-1} \left| y(n,m) - \hat{z}_2(n,m) e^{j\eta_m} \right|^2, \tag{10.48}$$

where

$$\hat{z}_2(n,m) = \sum_{k=1}^{K} \hat{\alpha}_k e^{j2\pi[\hat{f}_k \hat{t}_{x_m}(n) + \hat{\tilde{f}}_k \hat{t}_{y_m}(n) + \hat{\bar{f}}_k \hat{t}_{z_m}(n)]}, \quad n = 0, 1, \cdots, N-1, \quad m = 1, 2, \cdots, M, \tag{10.49}$$

and $\hat{t}_{x_m}(n)$, $\hat{t}_{y_m}(n)$, and $\hat{t}_{z_m}(n)$ are the same as $t_{x_m}(n)$, $t_{y_m}(n)$, and $t_{z_m}(n)$, respectively, except that $\{\theta_m, \phi_m\}_{m=1}^{M}$ are replaced by $\{\hat{\theta}_m, \hat{\phi}_m\}_{m=1}^{M}$, obtained in Step (1). This step is similar to [11] and we have

$$\hat{\eta}_m = \text{angle}(\hat{\mathbf{z}}_{2_m}^H \mathbf{y}_m), \quad m = 1, 2, \cdots, M, \tag{10.50}$$

where

$$\mathbf{y}_m = \begin{bmatrix} y(0,m) & y(1,m) & \cdots & y(N-1,m) \end{bmatrix}^T, \tag{10.51}$$

and

$$\hat{\mathbf{z}}_{2_m} = \begin{bmatrix} \hat{z}_2(0,m) & \hat{z}_2(1,m) & \cdots & \hat{z}_2(N-1,m) \end{bmatrix}^T. \tag{10.52}$$

Note from (10.50) that we do not need the search over parameter space to determine $\{\hat{\eta}_m\}_{m=1}^{M}$ and hence the errors in $\{R_0(m)\}_{m=1}^{M}$ are easier to deal with than those in $\{\theta_m, \phi_m\}_{m=1}^{M}$.

Step (3): Repeat Step (1) by replacing $y(n,m)$ with $\hat{y}(n,m) = y(n,m)e^{-j\hat{\eta}_m}$, $m = 1, 2, \cdots, M$, where $\{\hat{\eta}_m\}_{m=1}^{M}$ are determined in Step (2).

Step (4): Repeat Steps (2) and (3) until "practical convergence", which is determined by the relative change of the cost function C_m in (10.46) between two consecutive iterations. In the numerical examples, we terminate the repetition of Steps (2) and (3) when the relative change of C_m is less than 10^{-3} between two consecutive iterations.

If the errors in $\{R_0(m)\}_{m=1}^{M}$ are known to be negligible, then Step (1) alone is sufficient for the autofocusing.

Assume next that the aperture parameter estimates $\{\hat{\theta}_m, \hat{\phi}_m, \hat{\eta}_m\}_{m=1}^{M}$ are given. Then the problem becomes the target feature extraction problem considered in [7]. As shown in [7], the estimates $\{\hat{\alpha}_k, \hat{f}_k, \hat{\tilde{f}}_k, \hat{\bar{f}}_k\}_{k=1}^{K}$ of $\{\alpha_k, f_k, \tilde{f}_k, \bar{f}_k\}_{k=1}^{K}$ can be obtained with the RELAX algorithm by equivalently minimizing the following NLS criterion [7]:

$$F\left(\left\{\alpha_k, f_k, \tilde{f}_k, \bar{f}_k\right\}_{k=1}^{K}\right) = \sum_{m=1}^{M} \sum_{n=0}^{N-1} |\hat{y}(n,m) - \hat{z}_3(n,m)|^2, \tag{10.53}$$

where $\hat{y}(n,m) = y(n,m)e^{-j\hat{\eta}_m}$ and $\hat{z}_3(n,m)$ has the same form as $z(n,m)$ in (10.26) except that $t_{x_m}(n)$, $t_{y_m}(n)$, and $t_{z_m}(n)$ are replaced by $\hat{t}_{x_m}(n)$, $\hat{t}_{y_m}(n)$, and $\hat{t}_{z_m}(n)$, respectively, which are determined by $\{\hat{\theta}_m, \hat{\phi}_m\}_{m=1}^M$ instead of $\{\theta_m, \phi_m\}_{m=1}^M$.

The RELAX algorithm minimizes the cost function F in (10.53) efficiently by letting only the parameters of one target scatterer vary and fixing all others at their most recently determined values. Let

$$F_k\left(\alpha_k, f_k, \tilde{f}_k, \bar{f}_k\right) = \sum_{m=1}^{M}\sum_{n=0}^{N-1}\left|\hat{y}_k(n,m) - \alpha_k e^{j2\pi[f_k\hat{t}_{x_m}(n)+\tilde{f}_k\hat{t}_{y_m}(n)+\bar{f}_k\hat{t}_{z_m}(n)]}\right|^2, \tag{10.54}$$

where $\hat{t}_{x_m}(n)$, $\hat{t}_{y_m}(n)$, and $\hat{t}_{z_m}(n)$ are the same as $t_{x_m}(n)$, $t_{y_m}(n)$, and $t_{z_m}(n)$, respectively, except that $\{\theta_m, \phi_m\}_{m=1}^M$ are replaced by $\{\hat{\theta}_m, \hat{\phi}_m\}_{m=1}^M$, and

$$\hat{y}_k(n,m) = \hat{y}(n,m) - \sum_{i=1, i\neq k}^{K}\hat{\alpha}_i e^{j2\pi[\hat{f}_i\hat{t}_{x_m}(n)+\hat{\tilde{f}}_i\hat{t}_{y_m}(n)+\hat{\bar{f}}_i\hat{t}_{z_m}(n)]}, \tag{10.55}$$

Then minimizing F_k in (10.54) with respect to α_k yields the estimate $\hat{\alpha}_k$ of α_k:

$$\hat{\alpha}_k = \left.\frac{\sum_{m=1}^{M}\sum_{n=0}^{N-1}\hat{y}_k(n,m)e^{-j2\pi[f_k\hat{t}_{x_m}(n)+\tilde{f}_k\hat{t}_{y_m}(n)+\bar{f}_k\hat{t}_{z_m}(n)]}}{MN}\right|_{f_k=\hat{f}_k, \tilde{f}_k=\hat{\tilde{f}}_k, \bar{f}_k=\hat{\bar{f}}_k}, \tag{10.56}$$

and

$$\{\hat{f}_k, \hat{\tilde{f}}_k, \hat{\bar{f}}_k\} = \arg\max_{f_k, \tilde{f}_k, \bar{f}_k}\left|\sum_{m=1}^{M}\sum_{n=0}^{N-1}\hat{y}_k(n,m)e^{-j2\pi[f_k\hat{t}_{x_m}(n)+\tilde{f}_k\hat{t}_{y_m}(n)+\bar{f}_k\hat{t}_{z_m}(n)]}\right|^2. \tag{10.57}$$

To speed up the RELAX algorithm via utilizing FFT, we proposed in [7] a Cartesian grid mapping approach by approximating $\{\hat{t}_{x_m}(n), \hat{t}_{y_m}(n), \hat{t}_{z_m}(n)\}_{m=1}^M$ on a Cartesian grid. When the Cartesian grid is fine enough, the errors introduced by the approximation are negligible. In this paper, we use a crude Cartesian grid and the FFT method to obtain an initial estimate of the parameters and then minimize (10.57) via a multidimensional search method, such as the FMINV function in PV-WAVE. The latter approach uses less computer memory and is more accurate.

Let $\bar{K}$ denote the intermediate number of scatterers. Then the steps of the RELAX algorithm for 3-D target feature extraction via the curvilinear SAR are:

Step [1] : Assume $\bar{K} = 1$. Obtain $\{\hat{\alpha}_1, \hat{f}_1, \hat{\tilde{f}}_1, \hat{\bar{f}}_1\}$ from $\hat{y}(n,m)$, $n = 0, 1, \cdots, N-1$, $m = 1, \cdots, M$.

Step [2] : Assume $\bar{K} = 2$. Obtain $\{\hat{y}_2(m,n)\}$ with (10.55) by using $\{\hat{\alpha}_1, \hat{f}_1, \hat{\tilde{f}}_1, \hat{\bar{f}}_1\}$ obtained in Step [1]. Obtain $\{\hat{\alpha}_2, \hat{f}_2, \hat{\tilde{f}}_2, \hat{\bar{f}}_2\}$ from $\{\hat{y}_2(n,m)\}$. Next, compute $\{\hat{y}_1(n,m)\}$ with (10.55) by using $\{\hat{\alpha}_2, \hat{f}_2, \hat{\tilde{f}}_2, \hat{\bar{f}}_2\}$ and redetermine $\{\hat{\alpha}_1, \hat{f}_1, \hat{\tilde{f}}_1, \hat{\bar{f}}_1\}$ from $\{\hat{y}_1(n,m)\}$.

Iterate the previous two substeps until "practical convergence" is achieved (to be discussed later on).

Step [3] : Assume $\bar{K} = 3$. Compute $\{\hat{y}_3(n,m)\}$ by using $\{\hat{\alpha}_k, \hat{f}_k, \hat{\tilde{f}}_k, \hat{\bar{f}}_k\}_{k=1}^{2}$ obtained in Step [2]. Obtain $\{\hat{\alpha}_3, \hat{f}_3, \hat{\tilde{f}}_3, \hat{\bar{f}}_3\}$ from $\{\hat{y}_3(n,m)\}$. Next, compute $\{\hat{y}_1(n,m)\}$ by using $\{\hat{\alpha}_k, \hat{f}_k, \hat{\tilde{f}}_k, \hat{\bar{f}}_k\}_{k=2}^{3}$ and redetermine $\{\hat{\alpha}_1, \hat{f}_1, \hat{\tilde{f}}_1, \hat{\bar{f}}_1\}$ from $\{\hat{y}_1(n,m)\}$. Then compute $\{\hat{y}_2(n,m)\}$ by using $\{\hat{\alpha}_k, \hat{f}_k, \hat{\tilde{f}}_k, \hat{\bar{f}}_k\}_{k=1,3}$ and redetermine $\{\hat{\alpha}_2, \hat{f}_2, \hat{\tilde{f}}_2, \hat{\bar{f}}_2\}$ from $\{\hat{y}_2(n,m)\}$.

Iterate the previous three substeps until "practical convergence".

Remaining Steps: Continue similarly until $\bar{K} = K$. (Whenever K is unknown, it can be estimated from the available data, for instance, by using generalized AIC rules which are particularly tailored to the RELAX method of parameter estimation. See, e.g., [14].)

The "practical convergence" in the iterations of the above RELAX method may be determined by checking the relative change of the cost function $F\left(\{\hat{\alpha}_k, \hat{f}_k, \hat{\tilde{f}}_k, \hat{\bar{f}}_k\}_{k=1}^{K}\right)$ in (10.53) between two consecutive iterations. In our numerical examples, we terminate the iterative process in each of the above steps when the aforementioned relative change is less than 10^{-3}. Our numerical examples show that the iterations usually converge in a few steps.

Finally, the relaxation-based autofocus (AUTORELAX) algorithm can be described with the following steps:

Step 1: Extract the target features $\{\hat{\alpha}_k, \hat{f}_k, \hat{\tilde{f}}_k, \hat{\bar{f}}_k\}_{k=1}^{K}$ with the RELAX algorithm from an initial curvilinear aperture by assuming that $\{\eta_m = 0\}_{m=1}^{M}$.

Step 2: Update the curvilinear aperture $\{\theta_m, \phi_m\}_{m=1}^{M}$ and the phase error $\{\eta_m\}_{m=1}^{M}$ with Steps (1) – (4). (If $\{\eta_m\}_{m=1}^{M}$ is known to be negligible, then only update $\{\theta_m, \phi_m\}_{m=1}^{M}$ with Step (1).)

Step 3: Redetermine the target parameters with the RELAX algorithm by using the curvilinear aperture $\{\hat{\theta}_m, \hat{\phi}_m\}_{m=1}^{M}$ and $\{\hat{\eta}_m\}_{m=1}^{M}$ obtained in Step 2.

Step 4: Repeat Steps 2 and 3 until "practical convergence".

The "practical convergence" of AUTORELAX is determined similarly to that of the RELAX algorithm. Since a minimization is performed at every iteration, the value of the cost function C in (10.44) cannot increase. As a result, under mild conditions, the AUTORELAX algorithm is bound to converge to at least a local minimum of C [14]. The local minimum may or may not be the global one depending on the data parameters. Our examples below show that AUTORELAX can be quite effective for autofocusing and feature extraction.

10.4 Experimental and Numerical Results

We first present an experimental example to show the performance of the AUTORELAX algorithm. The field data was obtained by the Deployable Signature Measurement System (DSMS), Carderock Division, Naval Surface Warfare Center, Bethesda, Maryland. The radar was carried on board of a helicopter. The radar carrier frequency is 9.449 GHz and the bandwidth 0.498 GHz. The data set was obtained with a curved aperture not exactly known but is roughly the same as the one shown in Figure 10.4(a), where there are 64 look angles and 64 samples per look angle. The radar was about 300 meters away from the ground target. The ground target consists of 13 corner reflectors on the ground plane and 7 corner reflectors mounted on a wooden tripod that is about 2.65 meters tall. The true distribution of the scatterers is shown in Figure 10.3, where the centers of the squares denote the locations of the scatterers in 3-D space and the length of each square is proportional to the modulus of the RCS of the corresponding scatterer. The triangles show the projections of the scatterer locations onto the ground plane and their sizes are also scaled to be proportional to the RCS's of the scatterers.

Figure 10.4(b) shows the scatterer distribution obtained with the RELAX algorithm from the initial aperture shown in Figure 10.4(a). Figure 10.5(b) shows the scatterer distribution obtained with the RELAX algorithm from a manually adjusted aperture shown in Figure 10.5(a). (This manually adjusted aperture was used in [7].) We note that as compared to Figure 10.3, the results in Figure 10.5(b) are obviously better than those in Figure 10.4(b).

We now first consider autofocusing only in the elevation direction by assuming that

no errors exist in both the azimuth angles $\{\theta_m\}_{m=1}^M$ and the distances $\{R_0(m)\}_{m=1}^M$. The AUTORELAX algorithm converges after 6 iterations. Figure 10.6 shows the target scatterers extracted by AUTORELAX with the search interval for $\{\Delta\phi_m\}_{m=1}^M$ being $\pm 0.34°$. Comparing Figures 10.4(b) and 10.5(b) with Figure 10.3, we find that AUTORELAX works well and the AUTORELAX results are slightly better than those obtained by using the manually adjusted aperture. Figure 10.7 shows that the manually adjusted and autofocused apertures fit quite well after adding a line to the former. Since the results in Figure 10.6(b) are better than those in Figure 10.5(b), it appears that a linear phase error (in ϕ_m as a function of θ_m) was introduced in the manually adjusted aperture. The linear phase error can cause shifts in $\tilde{f}_k$ and the amount of shift is different for different scatterer k, as can be seen from (10.35).

Consider next autofocusing in both the elevation and azimuth directions by assuming that no errors exist in $\{R_0(m)\}_{m=1}^M$. The AUTORELAX algorithm converges after 7 iterations. Figure 10.8 shows the results obtained with the search intervals for $\{\Delta\phi_m\}_{m=1}^M$ and $\{\Delta\theta_m\}_{m=1}^M$ being $\pm 0.34°$ and $\pm 0.006°$, respectively. Compared with the results obtained by autofocusing only in the elevation direction, we find that autofocusing in both directions provides little further improvement to the accuracy of the estimated target parameters in this example. Hence for this example, it appears that the aperture errors mainly occur in the elevation direction.

For this experimental example, we have also used AUTORELAX to extract target features when we assumed that errors exist in both $\{\theta_m, \phi_m\}_{m=1}^M$ and $\{R_0(m)\}_{m=1}^M$. Again, as compared with Figure 10.6(b), we have noticed little change in the extracted target parameters. It could be that DSMS has already done a good job compensating for the errors in $\{R_0(m)\}_{m=1}^M$ with some traditional method so that they are now negligible as compared with the errors in $\{\phi_m\}_{m=1}^M$.

Finally, we use a simulation example to show that the accuracy of target feature extraction via CLSAR is very sensitive to the accuracy of the curvilinear aperture. We assume that there are 20 scatterers with a similar distribution to that in the experimental example above. The true aperture is shown in Figure 10.9(a) and the scatterer distribution is shown in Figure 10.9(b). Here we consider the case where the aperture errors exist only in the ele-

vation direction. Figure 10.10(b) shows the scatterer distribution obtained with the RELAX algorithm from the initial aperture shown in Figure 10.10(a). We note that the scatterer distribution in Figure 10.10(b) is quite different from the true one in Figure 10.9(b).

When used with the simulation data, AUTORELAX converges after four iterations. Figure 10.11 shows the autofocused aperture and the scatterer distribution obtained with AUTORELAX. We see that the scatterer distribution obtained by AUTORELAX is almost the same as the true one. We also notice that the shape of the autofocused aperture is closer to the true one than the initial aperture. The constant difference between the autofocused and the true apertures will cause phase errors in $\{\alpha_k\}_{k=1}^{K}$, which cannot be eliminated due to the ambiguity problems discussed in Section 10.2.3.

10.5 Conclusions

This chapter introduces a relaxation-based autofocus (AUTORELAX) algorithm to estimate the aperture errors in CLSAR. The AUTORELAX algorithm can be used to extract the 3-D target features more accurately via CLSAR in the presence of unknown curvilinear aperture errors. The experimental and numerical results have shown that AUTORELAX is quite an effective method for autofocusing and 3-D target feature extraction.

Reference

[1] L. C. Graham, "Synthetic interferometer radar for topographic mapping," *Proceedings of the IEEE*, vol. 62, pp. 763–768, June 1974.

[2] R. Gens and J. L. V. Gendren, "SAR interferometry - issues, techniques, applications," *International Journal of Remote Sensing*, vol. 17, pp. 1803–1835, July 1996.

[3] J. Li, Z.-S. Liu, and P. Stoica, "3-D target feature extraction via interferometric SAR," *IEE Proceedings on Radar, Sonar and Navigation*, vol. 144, April 1997.

[4] K. Knaell, "Three-dimensional SAR from curvilinear apertures," *SPIE Proceedings on Optical Engineering in Aerospace Sensing*, vol. 2230, pp. 120–134, Orlando, FL, April 1994.

[5] K. Knaell, "Three-dimensional SAR from practical apertures," *SPIE Proceedings on Optical Engineering in Aerospace Sensing, San Diego*, vol. 2562, pp. 31–41, 1995.

[6] K. Knaell, "Three-dimensional SAR from curvilinear apertures," *Proceedings of the 1996 IEEE National Radar Conference*, pp. 220–225, Ann Arbor, MI, May 1996.

[7] J. Li, Z. Bi, Z.-S. Liu, and K. Knaell, "Using curvilinear SAR for three-dimensional target feature extraction," *IEE Proceedings on Radar, Sonar and Navigation*, vol. 144, pp. 275–283, October 1997.

[8] P. H. Eichel, D. C. Ghiglia, and C. V. Jakowatz, Jr., "Speckle processing methods for synthetic-aperture -radar phase correction," *Optics Letters*, vol. 14, pp. 1–3, January 1989.

[9] P. H. Eichel and C. V. Jakowatz, Jr., "Phase-gradient algorithm as an optimal estimator of the phase derivative," *Optics Letters*, vol. 14, pp. 1101–1103, October 1989.

[10] D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and C. V. Jakowatz, Jr., "Phase gradient autofocus - a robust tool for high resolution SAR phase correction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, pp. 827–835, July 1994.

[11] Z.-S. Liu and J. Li, "SAR motion compensation and feature extraction via MCRELAX," *Journal of the Optical Society of America A*, March 1998.

[12] D. C. Munson, Jr., J. D. O'Brien, and W. K. Jenkins, "A tomographic formulation of spotlight-mode synthetic aperture radar," *Proceedings of the IEEE*, vol. 71, pp. 917 –925, August 1983.

[13] C. V. Jakowatz, Jr., D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. A. Thompson, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach.* Norwell, MA: Kluwer Academic Publishers, 1996.

[14] J. Li and P. Stoica, "Efficient mixed-spectrum estimation with applications to target feature extraction," *IEEE Transactions on Signal Processing*, vol. 44, pp. 281–295, February 1996.

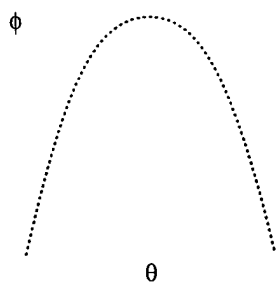


Figure 10.1: A possible curvilinear aperture for CLSAR.

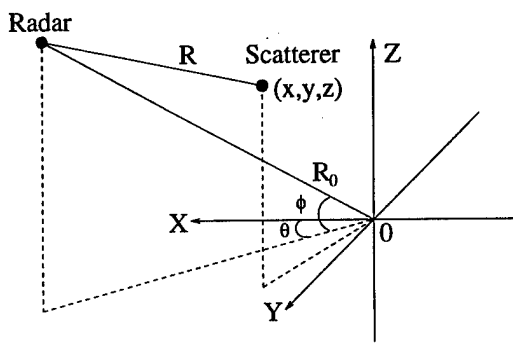


Figure 10.2: A broadside data collection geometry.

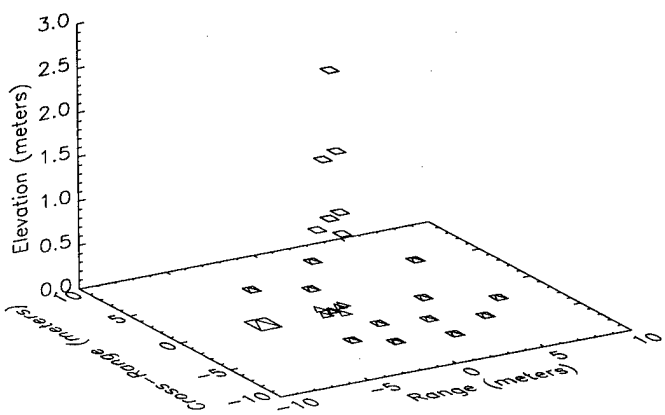
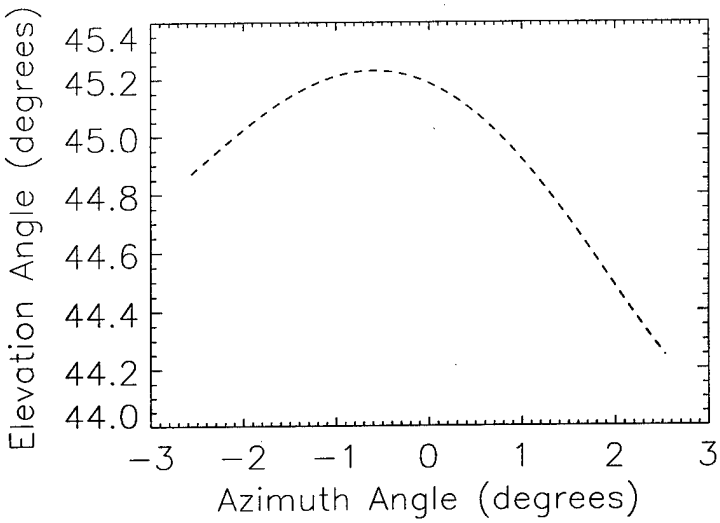
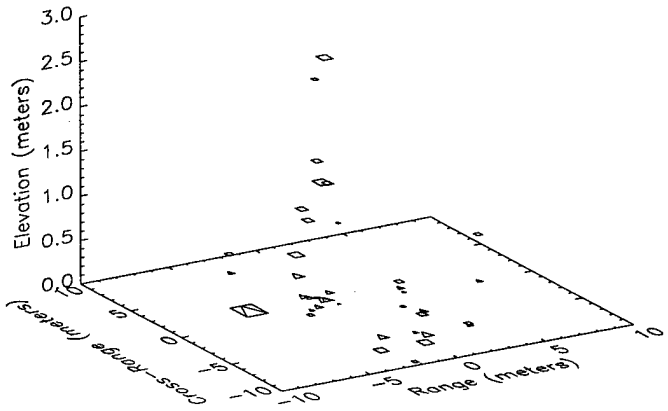


Figure 10.3: True scatterer distribution for the experimental example.

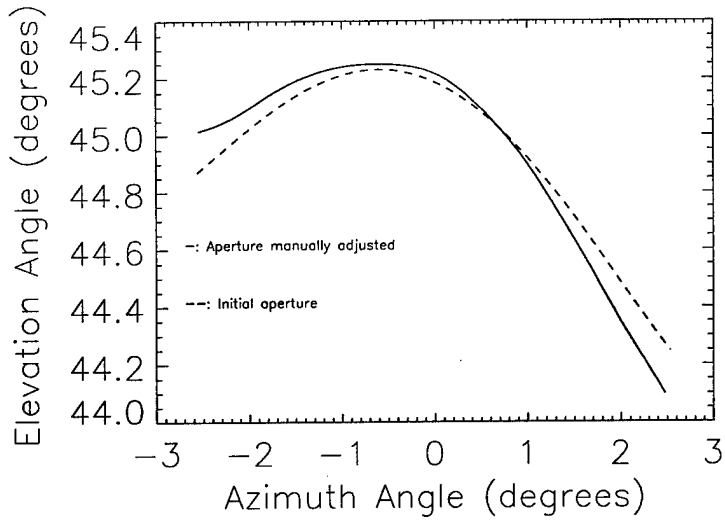


(a)

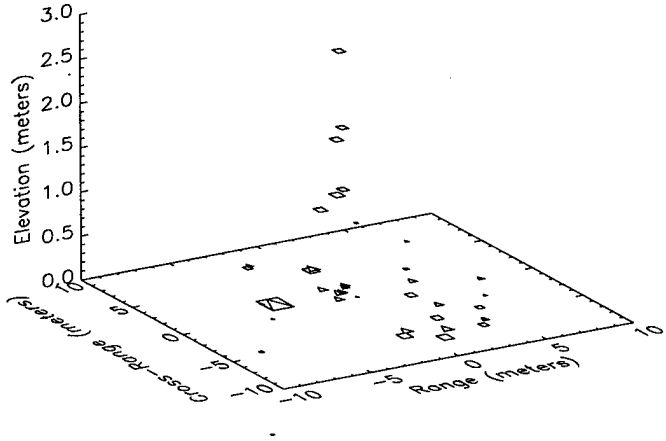


(b)

Figure 10.4: (a) Initial curvilinear aperture for the experimental example. (b) 3-D plot of $K = 20$ scatterers extracted from the initial aperture in (a) with RELAX.

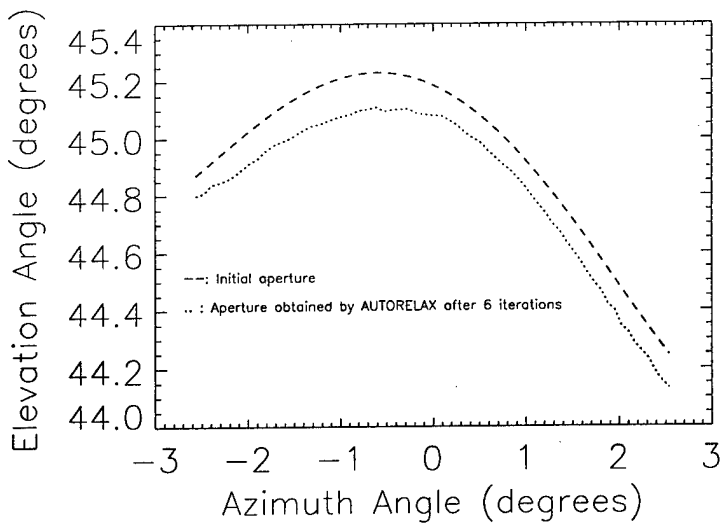


(a)

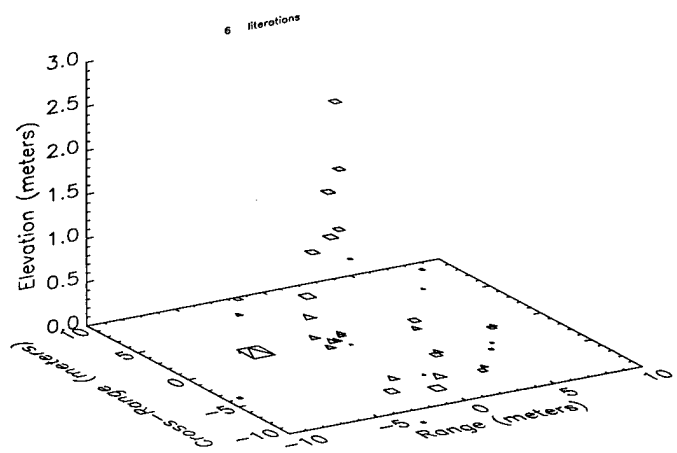


(b)

Figure 10.5: (a) Manually adjusted aperture for the experimental example. (b) Scatterer distribution obtained by using RELAX with $K = 20$ and the aperture in (a).

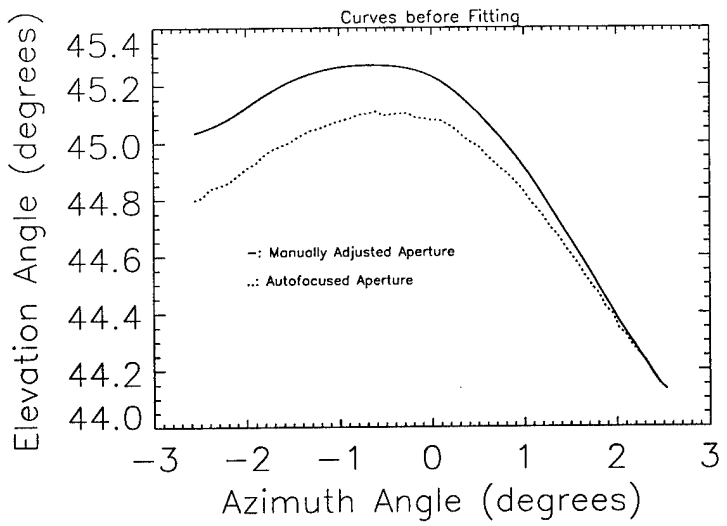


(a)

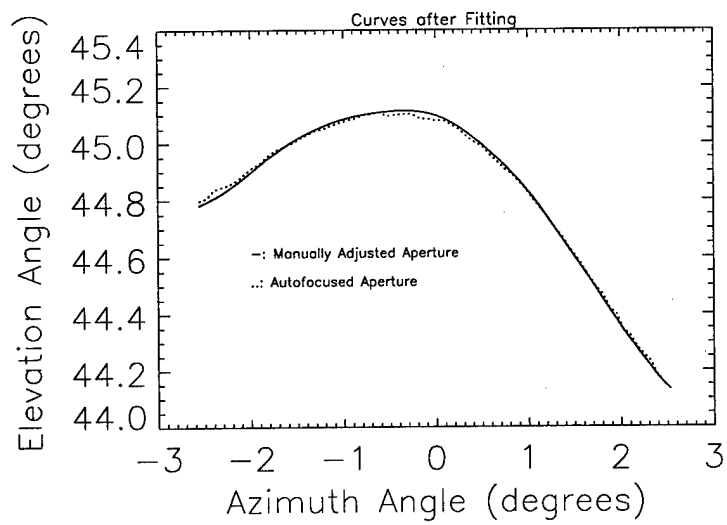


(b)

Figure 10.6: Autofocused curvilinear aperture and scatterer distribution obtained with AUTORELAX by autofocusing only in the elevation direction and using $K = 20$ for the experimental example. (a) Autofocused curvilinear aperture. (b) Scatterer distribution.

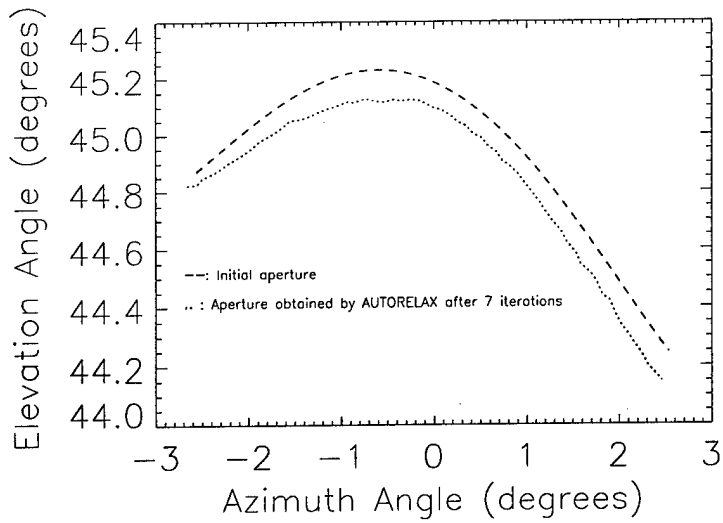


(a)

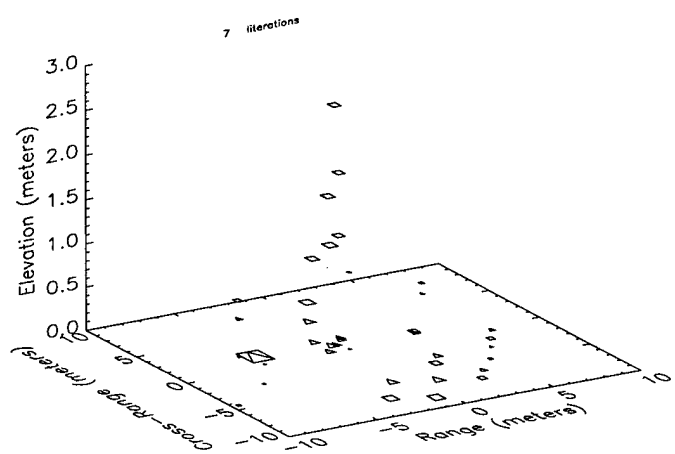


(b)

Figure 10.7: (a) Manually adjusted and autofocused curvilinear apertures for the experimental example. (b) Fitting the manually adjusted aperture to the autofocused aperture by adding a line to the former.

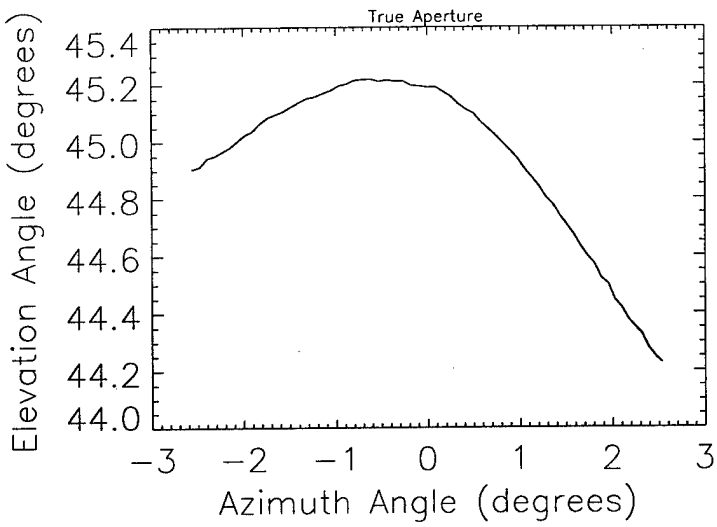


(a)

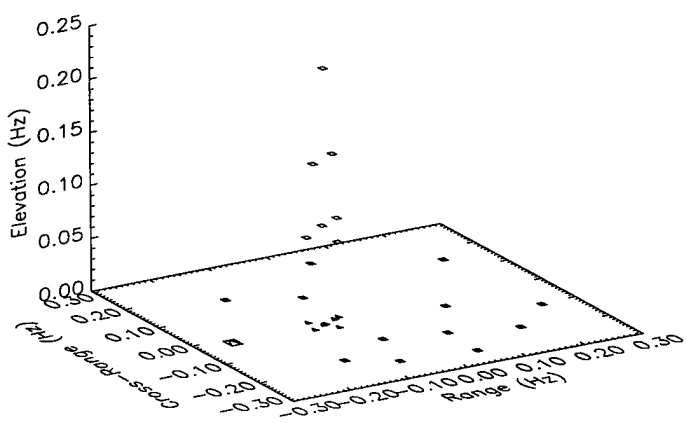


(b)

Figure 10.8: Autofocused curvilinear aperture and scatterer distribution obtained with AUTORELAX by autofocusing in both the elevation and azimuth directions and using $K = 20$ for the experimental example. (a) Autofocused curvilinear aperture. (b) Scatterer distribution.

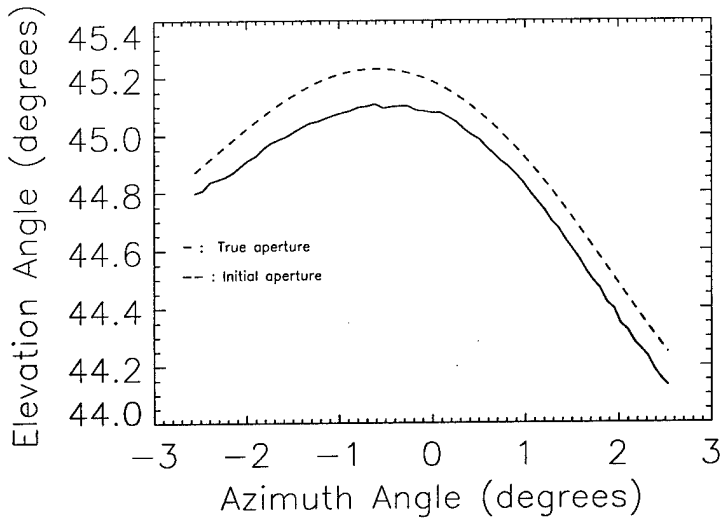


(a)

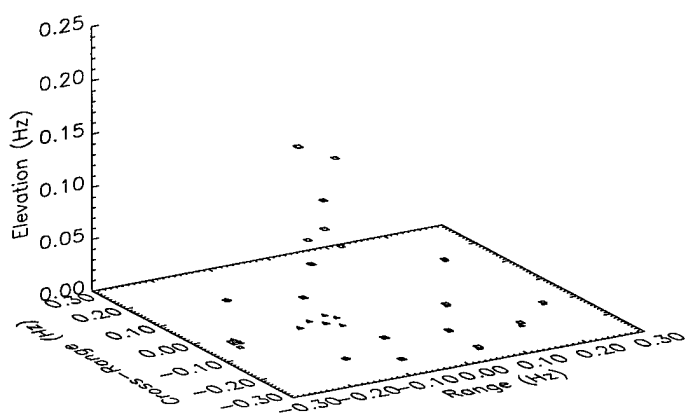


(b)

Figure 10.9: (a) True curvilinear aperture for the simulation example. (b) True scatterer distribution for the simulation example.

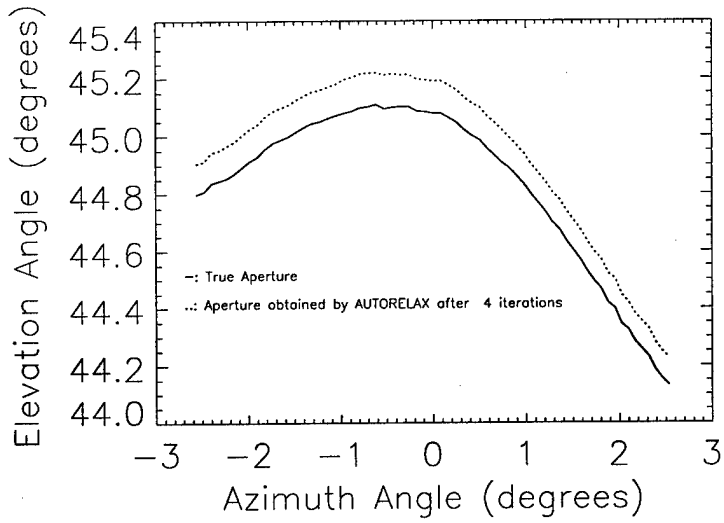


(a)

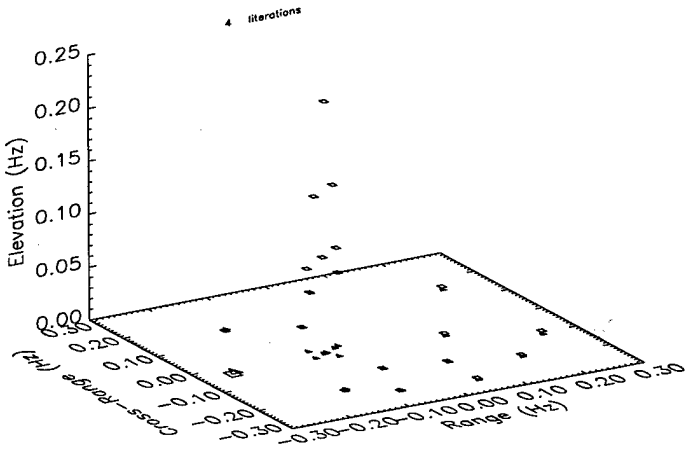


(b)

Figure 10.10: (a) Initial curvilinear aperture for the simulation example. (b) Scatterer distribution obtained from the initial aperture in (a) by using RELAX with $K = 20$.



(a)



(b)

Figure 10.11: Autofocused curvilinear aperture and scatterer distribution obtained by using AUTORELAX with $K = 20$ for the simulation example. (a) Autofocused curvilinear aperture. (b) Scatterer distribution.

11. An Efficient Algorithm for Time Delay Estimation

11.1 Introduction

Suppose that we have a single sensor receiving a superimposition of attenuated and delayed replicas of a known signal plus noise. From the received data we want to estimate the arrival times of the various replicas and their (complex or real) attenuation coefficients (gains). This is the well-known time delay estimation problem which occurs in many fields including radar, active sonar, propagation modeling in wireless communications, nondestructive testing, geophysical/seismic exploration, and medical imaging. Another related problem is Time Difference of Arrival (TDOA) estimation of a signal that has been intercepted by multiple sensors where the signal waveform is unknown or random in nature, which occurs in passive source localization systems. In this paper, we will concentrate only on time delay estimation based on one sensor with known signal shapes.

The most well-known time delay estimator is the matched filter approach. If there is only one signal or the overlapped signals are separated in time by an interval that is much greater than the width of the signal autocorrelation function, then the matched filter is the optimal estimator when the noise is white Gaussian [1] . The resolution capability of the matched filter approach depends on the signal bandwidth and the larger the signal bandwidth, the better the resolution. However, in many situations there exist some practical limitations on increasing the bandwidth of the transmitted signals. How to resolve closely spaced overlapping noisy echoes has attracted the attention of researchers from many fields for several decades.

Several approaches have been suggested for this problem and many of them benefit from the recent development of high resolution sinusoidal frequency estimation and Direction of Arrival (DOA) estimation techniques. For example, MUSIC [2] is employed in [3] to estimate time delays with multiple experimental data and the approach requires that the signal gains be random in order to get a nonsingular covariance matrix. Sinusoidal frequency estimation techniques such as MUSIC [2], Linear Prediction [4], and MODE [5] are applied to the

time delay estimation problem in [6, 7, 8]. However, these approaches are only applicable to signals with flat (rectangular) band-limited spectra. Several Maximum Likelihood (ML) approaches have also been suggested for this problem. Multidimensional global optimization algorithms are presented in [9, 10, 11] to analyze a special class of ocean acoustic data that has very oscillatory autocorrelation functions. An efficient approach based on the Expectation Maximization (EM) algorithm [12] is proposed in [13] that decouples the complicated multidimensional optimization problem into a sequence of multiple separate one-dimensional optimization problems. However, its convergence depends highly on the initialization method used and no systematic initialization method is given in [13].

In this chapter, we first formulate the time delay estimation problem as a nonlinear least squares (NLS) fitting problem in the frequency domain. Then a weighted Fourier transform based relaxation method (referred to as WRELAX) is presented for finding the global minimum of the complicated multimodal NLS cost function. The most striking feature of the WRELAX algorithm is that it decouples the multidimensional optimization problem into a series of one-dimensional optimization problems in a conceptually and computationally simple way. Compared with other existing algorithms, WRELAX is more systematic and efficient and has less limitations on the signal shapes. The WRELAX algorithm is also extended to the case of multiple looks for different scenarios (i.e., fixed delays but arbitrary gains and fixed delays and gains). Simulation results show that the mean square error (MSE) of WRELAX is very close to the corresponding Cramér-Rao bound (CRB) for a wide range of signal-to-noise ratios (SNRs). The new algorithm is also successfully applied to detecting and classifying roadway subsurface anomalies by using an ultra wideband ground penetrating radar.

The remainder of the chapter is organized as follows. In Section 11.2, we describe the data model and formulate the problem of interest. Section 11.3 presents the WRELAX algorithm. In Section 11.4, we extend the WRELAX algorithm to two cases of multiple looks. Some numerical and experimental results are given in Section 11.5. Section 11.6 concludes the chapter. The CRB analysis is included in the appendix.

11.2 Problem Formulation

Time delay estimation is a well-known traditional problem occurring frequently in radar, active sonar, and many other fields. In this problem, the waveform received at a single sensor consists of delayed replicas of the transmitted signal with different gains. The gains reflect the scattering property of the targets or multipath channel transmission features. The received signal waveform $y(t)$ can be described as

$$y(t) = \sum_{l=1}^{L} \alpha_l s(t - \tau_l) + e(t) \ , \quad 0 \leq t \leq T, \tag{11.1}$$

where $s(t)$, $0 \leq t \leq T_0$, represents the known transmitted signal, $y(t)$ denotes the received signal, which is composed of L replicas of $s(t)$ with different (complex or real valued) gains $\{\alpha_l\}_{l=1}^{L}$ and real valued delays $\{\tau_l\}_{l=1}^{L}$, and $e(t)$ is the receiver noise, which is modeled as a zero-mean Gaussian random process.

Usually, the above received analog signal is sampled for digital signal processing. To avoid aliasing, we must sample $y(t)$ according to the bandwidth of $s(t)$. Let B_s denote the double-sided bandwidth of $s(t)$. Then $y(t)$ must be sampled with the sampling frequency f_s satisfying

$$f_s \geq B_s \ . \tag{11.2}$$

After A/D conversion, the sampled received signal has the form

$$y(nT_s) = \sum_{l=1}^{L} \alpha_l s(nT_s - \tau_l) + e(nT_s), \quad n = 0, 1, \cdots, N-1, \tag{11.3}$$

where T_s is the sampling period and is equal to the reciprocal of the sampling frequency f_s.

Our problem of interest herein is to estimate $\{\alpha_l, \tau_l\}_{l=1}^{L}$ from $\{y(nT_s)\}_{n=0}^{N-1}$ with known $s(t)$, $0 \leq t \leq T_0$, or $\{s(nT_s)\}_{n=0}^{N-1}$.

Although we could solve the estimation problem in the time-domain [3, 9, 10, 13], we shall consider below solving the problem in the frequency domain and propose a relaxation based algorithm that requires a sequence of Fourier transforms on some weighted data vectors. This algorithm is referred to as the WRELAX algorithm.

Let $Y(k)$, $S(k)$, and $E(k)$, $k = -N/2, -N/2+1, ..., N/2-1$, denote the discrete Fourier transforms (DFT's) of $y(nT_s)$, $s(nT_s)$, and $e(nT_s)$, respectively. Provided that aliasing is negligible, then $Y(k)$ can be written as :

$$Y(k) = S(k) \sum_{l=1}^{L} \alpha_l e^{j\omega_l k} + E(k), \tag{11.4}$$

where

$$\omega_l = -\frac{2\pi\tau_l}{NT_s} . \tag{11.5}$$

Note that the time delay estimation problem is similar to the sinusoidal parameter estimation problem except that the exponential signals are weighted by the known signal spectrum. If we divided both sides of (11.4) by $S(k)$, the problem would become identical to the sinusoidal parameter estimation problem. Yet we should not do so for the following reasons: first, $S(k)$ could be zero for some k; second, the noise $E(k)/S(k)$ will no longer be a white noise even when $E(k)$ is white; third, when $E(k)$ is a white noise, the larger the $S(k)$ at sample k, the higher the signal-to-noise ratio (SNR) of the corresponding $Y(k)$ and hence dividing $Y(k)$ by $S(k)$ will de-emphasize those $Y(k)'s$ that have high SNRs. Because of this, many well-known sinusoidal parameter estimation algorithms, such as MUSIC [2], ESPRIT [14], PRONY [15], are not directly applicable to our problem of interest. Using MODE [5] would require a multidimensional search over a parameter space because we can no longer reparameterize the MODE cost function via the coefficients of a polynomial.

11.3 The WRELAX Algorithm

We consider below estimating the unknown parameters by minimizing the following NLS criterion:

$$C_1(\{\alpha_l, \omega_l\}_{l=1}^{L}) = \sum_{k=-N/2}^{N/2-1} \left| Y(k) - S(k) \sum_{l=1}^{L} \alpha_l e^{j\omega_l k} \right|^2 . \tag{11.6}$$

When $e(nT_s)$ is a zero-mean white Gaussian random process, $E(k)$ is also white since DFT is a unitary transformation. For this white noise case, the NLS approach is the same as the ML method. When $E(k)$ is not white, however, the NLS approach is no longer the ML method. However, it has been shown in [16] that the NLS approach can still have excellent statistical accuracy.

Minimizing $C_1(\{\alpha_l, \omega_l\}_{l=1}^L)$ with respect to the unknown parameters is a highly nonlinear optimization problem. The cost function has a complicated multimodal shape with a very small attraction domain, which makes it very difficult to find the global minimum. Below, we present a relaxation based optimization algorithm to obtain the NLS parameter estimates. Before we present our approach, let us consider the following preparations. Let

$$\mathbf{Y} = \begin{bmatrix} Y(-N/2) & Y(-N/2+1) & \cdots & Y(N/2-1) \end{bmatrix}^T, \tag{11.7}$$

$$\mathbf{S} = \text{diag}\left\{ S(-N/2) \quad S(-N/2+1) \quad \cdots \quad S(N/2-1) \right\}, \tag{11.8}$$

$$\mathbf{E} = \begin{bmatrix} E(-N/2) & E(-N/2+1) & \cdots & E(N/2-1) \end{bmatrix}^T, \tag{11.9}$$

and

$$\mathbf{a}(\omega_l) = \begin{bmatrix} e^{j\omega_l(-N/2)} & e^{j\omega_l(-N/2+1)} & \ldots & e^{j\omega_l(N/2-1)} \end{bmatrix}^T, \tag{11.10}$$

where $(\cdot)^T$ denotes the transpose. Denote

$$\mathbf{Y}_l = \mathbf{Y} - \sum_{i=1, i\neq l}^{L} \hat{\alpha}_i[\mathbf{S}\mathbf{a}(\hat{\omega}_i)] \tag{11.11}$$

where $\{\hat{\alpha}_i, \hat{\omega}_i\}_{i=1, i\neq l}$ are assumed to be given. Consider first the case where $\{\alpha_l\}_{l=1}^L$ are complex valued. Let

$$\mathbf{b}(\omega_l) = \mathbf{S}\mathbf{a}(\omega_l), \quad l = 1, 2, \cdots, L. \tag{11.12}$$

Then (11.6) becomes

$$C_2(\alpha_l, \omega_l) = \| \mathbf{Y}_l - \alpha_l \mathbf{b}(\omega_l) \|^2, \tag{11.13}$$

where $\|\cdot\|$ denotes the Euclidean norm. Minimizing $C_2(\alpha_l, \omega_l)$ with respect to α_l yields the estimate $\hat{\alpha}_l$ of α_l

$$\begin{aligned}\hat{\alpha}_l &= \frac{\mathbf{b}^H(\omega_l)\mathbf{Y}_l}{\mathbf{b}^H(\omega_l)\mathbf{b}(\omega_l)} \\ &= \frac{\mathbf{a}^H(\omega_l)(\mathbf{S}^*\mathbf{Y}_l)}{\| \mathbf{S} \|_F^2},\end{aligned} \tag{11.14}$$

where $(\cdot)^H$ represents the conjugate transpose and $\|\cdot\|_F$ denotes the *Frobenius* norm [17]. Then the estimate $\hat{\omega}_l$ of ω_l is obtained as follows:

$$\begin{aligned}\hat{\omega}_l &= \arg\min_{\omega_l} \left\| \mathbf{Y}_l - \frac{\mathbf{b}(\omega_l)\mathbf{b}^H(\omega_l)}{\mathbf{b}^H(\omega_l)\ bbf(\omega_l)}\mathbf{Y}_l \right\|^2 \\ &= \arg\max_{\omega_l} \frac{\mathbf{Y}_l^H\mathbf{b}(\omega_l)\mathbf{b}^H(\omega_l)\mathbf{Y}_l}{\mathbf{b}^H(\omega_l)\mathbf{b}(\omega_l)} \\ &= \arg\max_{\omega_l} \left|\mathbf{a}^H(\omega_l)(\mathbf{S}^*\mathbf{Y}_l)\right|^2,\end{aligned} \tag{11.15}$$

where we have used the fact that $\mathbf{b}^H(\omega_l)\mathbf{b}(\omega_l) = \| \mathbf{S} \|_F^2$ and hence is independent of ω_l. Hence $\hat{\omega}_l$ is obtained as the location of the dominant peak of the magnitude squared of the Fourier transform, $|\mathbf{a}^H(\omega_l)(\mathbf{S}^*\mathbf{Y}_l)|^2$, which can be efficiently computed by using the fast Fourier transform (FFT) with the weighted data vector $\mathbf{S}^*\mathbf{Y}_l$ padded with zeros. An alternative scheme to zero-padding FFT is to find an approximate peak location first by using FFT without much zero-padding and then perform a fine search nearby the approximate peak location by, for example, the *fmin* function in MATLAB, which uses the Golden section search algorithm. With the estimate of ω_l at hand, $\hat{\alpha}_l$ is easily computed from the corresponding complex height:

$$\hat{\alpha}_l = \left.\frac{\mathbf{a}^H(\omega_l)(\mathbf{S}^*\mathbf{Y}_l)}{\| \mathbf{S} \|_F^2}\right|_{\omega_l=\hat{\omega}_l}. \tag{11.16}$$

With the above simple preparations, we now present the WRELAX algorithm.

Step (1): Assume $L = 1$. Obtain $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1}$ from $\mathbf{Y}$ by using (11.15) and (11.16).

Step (2): Assume $L = 2$. Compute $\mathbf{Y}_2$ with (11.11) by using $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1}$ obtained in Step (1). Obtain $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=2}$ from $\mathbf{Y}_2$. Next, compute $\mathbf{Y}_1$ by using $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=2}$ and then redetermine $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1}$ from $\mathbf{Y}_1$.

Iterate the previous two substeps until "practical convergence" is achieved (to be discussed later on).

Step (3): Assume $L = 3$. Compute $\mathbf{Y}_3$ by using $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1}^{2}$ obtained in Step (2). Obtain $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=3}$ from $\mathbf{Y}_3$. Next, compute $\mathbf{Y}_1$ by using $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=2}^{3}$ and redetermine $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1}$ from $\mathbf{Y}_1$. Then compute $\mathbf{Y}_2$ by using $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1,3}$ and redetermine $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=2}$ from $\mathbf{Y}_2$.

Iterate the previous three substeps until "practical convergence".

Remaining Steps: Continue similarly until L is equal to the desired or estimated number of signals. (Whenever L is unknown, it can be estimated from the available data, for instance, by using the generalized AIC rules which are particularly tailored to the WRELAX method of parameter estimation. See, for example, [16].)

The "practical convergence" in the iterations of the above WRELAX method may be determined by checking the relative change of the cost function $C_1(\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1}^{L})$ in (11.6) between two consecutive iterations. The algorithm is bound to converge to at least some local minimum point [18]. The convergence speed depends on the time delay spacing of the signals. If the spacing between any two signals is larger than the reciprocal of the signal bandwidth, the algorithm converges in a few steps. As the spacing of the signals becomes closer, the convergence speed becomes slower.

Once we have obtained the estimates $\{\hat{\omega}_l\}_{l=1}^{L}$, the estimates $\{\hat{\tau}_l\}_{l=1}^{L}$ of $\{\tau_l\}_{l=1}^{L}$ can be determined by using (11.5).

At this point, we would like to point out the relationship between WRELAX and the conventional matched filter approach. The matched filter approach can also be formulated in the frequency domain. Let

$$F(\omega) = \left|\mathbf{a}^H(\omega)(\mathbf{S}^*\mathbf{Y})\right|^2 . \tag{11.17}$$

The matched filter method searches for the L largest peak positions of $F(\omega)$ as the estimates of $\{\omega_l\}_{l=1}^{L}$, and then the gains are determined as follows

$$\hat{\alpha}_l = \left.\frac{\mathbf{a}^H(\omega_l)(\mathbf{S}^*\mathbf{Y})}{\| \mathbf{S} \|_F^2}\right|_{\omega_l=\hat{\omega}_l} , \quad l = 1, 2, \cdots, L . \tag{11.18}$$

Hence when there is only one signal, this one-dimensional matched filter approach is equiv-

alent to the WRELAX algorithm. However, when there are multiple signals that are not well separated, this conventional matched filter approach will perform poorly. In this case, a multidimensional matched filter method [9] could be used and the method is equivalent to the NLS fitting approach [9]. The WRELAX algorithm decouples the multidimensional matched filters into a sequence of one-dimensional matched filters. Thus the excellent parameter estimation performance of the NLS fitting approach can be achieved at a much lower implementation cost.

Similar to the WRELAX algorithm, the EM algorithm proposed in [13] also transforms the multidimensional optimization problem into a series of one-dimensional optimization problems. The detailed implementations of the algorithms, however, are quite different. The EM algorithm consists of two steps, the E (Estimate) step and the M (Maximize) step. The idea is to decompose the observed data into their signal components (the E step) and then to estimate the parameters of each signal component separately (the M step). The algorithm is iterative, using the current parameter estimates to decompose the observed data. At each E step, the residue error corresponding to the current estimates is also decomposed among different signal components. Although initial conditions are needed by EM, no systematic initialization method is given in [13]. We have also found that the performance of EM is very sensitive to the initial conditions used. Even with the same initial conditions, our numerical examples show that the convergence speed of EM can be much slower than the last step of WRELAX. Further, WRELAX does not require any initial condi tions before its iterations and the first $L-1$ steps of WRELAX can provide an excellent initial condition for Step L.

Consider next the case where $\{\alpha_l\}_{l=1}^{L}$ are real-valued. Minimizing $C_2(\alpha_l, \omega_l)$ with respect to α_l and ω_l yields

$$\hat{\alpha}_l = \left. \frac{\mathrm{Re}\left[\mathbf{a}^H(\omega_l)(\mathbf{S}^*\mathbf{Y}_l)\right]}{\| \mathbf{S} \|_F^2} \right|_{\omega_l=\hat{\omega}_l}, \tag{11.19}$$

where $\mathrm{Re}(\mathbf{X})$ denotes the real part of $\mathbf{X}$, and

$$\hat{\omega}_l = \arg\max_{\omega_l} \mathrm{Re}^2\left[\mathbf{a}^H(\omega_l)(\mathbf{S}^*\mathbf{Y}_l)\right]. \tag{11.20}$$

The WRELAX algorithm could also be implemented in the time domain, which is based

on the correlations. However, we prefer to use the frequency domain version of WRELAX. For the time domain version, we could be restricted to using the discrete values of $\{\tau_l\}_{l=1}^{L}$ if we only know the sampled version of $s(t)$. For this case, if a more accurate delay estimate is required, then one has to resort to interpolation [9]. This inconvinence can be avoided by transforming the problem to the frequency domain, where $\{\tau_l\}_{l=1}^{L}$ can take on a continuum of values. Even without considering the additional interpolation cost, the computational load of the time domain correlation-based WRELAX is heavier than that of the frequency domain WRELAX, which can be easily implemented by using the currently available dedicated high performance FFT chips, such as TMC2310 [19] and A41102 [20].

11.4 The Extended WRELAX Algorithms for Multiple Looks

Next we extend the above WRELAX algorithm to the case of multiple looks. Two scenarios will be considered, which include 1) fixed delays but arbitrary gains and 2) fixed delays and fixed gains. In radar applications, the two cases correspond to two target fluctuation models [21].

11.4.1 Fixed Delays but Arbitrary Gains

Consider the case where multiple pulses are transmitted and the ranges of target scatterers remain the same but their gains change randomly during the observation interval.

Let $\mathbf{Y}^{(m)}$ be the DFT of the received vector due to the mth pulse. Then

$$\mathbf{Y}^{(m)} = \sum_{l=1}^{L} \alpha_l^{(m)} [\mathbf{Sa}(\omega_l)] + \mathbf{E}^{(m)}, \quad m = 1, 2, ..., M, \tag{11.21}$$

where $\alpha_l^{(m)}$ denotes the gain of the lth scatterers due to the mth pulse and the noise vectors $\{\mathbf{E}^{(m)}\}_{m=1}^{M}$ are assumed independent of each other. Our problem of interest is to estimate $\left\{\alpha_l^{(m)}, \omega_l\right\}_{l=1,..,L;m=1,...,M}$ from $\left\{\mathbf{Y}^{(m)}\right\}_{m=1}^{M}$.

We now extend the WRELAX algorithm to this multiple look case. The extended WRE-

LAX algorithm minimizes the following NLS criterion:

$$C_3(\{\alpha_l^{(m)}, \omega_l\}_{l=1,..,L;m=1,...,M}) = \sum_{m=1}^{M} \left\| \mathbf{Y}^{(m)} - \sum_{l=1}^{L} \alpha_l^{(m)} \mathbf{b}(\omega_l) \right\|^2, \tag{11.22}$$

where $\mathbf{b}(\omega_l)$ is defined in (11.12).

Before we present the extended WRELAX algorithm, let us consider the following preparations. Let

$$\mathbf{Y}_l^{(m)} = \mathbf{Y}^{(m)} - \sum_{i=1, i\neq l}^{L} \hat{\alpha}_i^{(m)} \mathbf{b}(\hat{\omega}_i), \quad m = 1, ..., M, \tag{11.23}$$

where $\{\hat{\alpha}_i^{(m)}, \hat{\omega}_i\}_{i=1,...,L, i\neq l, m=1,...,M}$ are assumed given. Then the cost function $C_3(\{\alpha_l^{(m)}, \omega_l\}_{l=1,..,L;m=1,...,M})$ becomes

$$C_4(\{\alpha_l^{(m)}\}_{m=1}^{M}, \omega_l) = \sum_{m=1}^{M} \left\| \mathbf{Y}_l^{(m)} - \alpha_l^{(m)} \mathbf{b}(\omega_l) \right\|^2. \tag{11.24}$$

Minimizing $C_4(\{\alpha_l\}_{l=1}^{L}, \omega_l)$ with respect to the complex-valued $\{\alpha_l^{(m)}\}_{m=1}^{M}$ and ω_l yields

$$\hat{\alpha}_l^{(m)} = \left. \frac{\mathbf{a}^H(\omega_l)(\mathbf{S}^* \mathbf{Y}_l^{(m)})}{\| \mathbf{S} \|_F^2} \right|_{\omega_l = \hat{\omega}_l}, \tag{11.25}$$

and

$$\hat{\omega}_l = \arg\max_{\omega_l} \left[\sum_{m=1}^{M} \left| \mathbf{a}^H(\omega_l)(\mathbf{S}^* \mathbf{Y}_l^{(m)}) \right|^2 \right]. \tag{11.26}$$

Minimizing $C_4(\{\alpha_l^{(m)}\}_{m=1}^{M}, \omega_l)$ with respect to the real-valued $\{\alpha_l^{(m)}\}_{m=1}^{M}$ and ω_l yields

$$\hat{\alpha}_l^{(m)} = \left. \frac{\mathrm{Re}\left[\mathbf{a}^H(\omega_l)(\mathbf{S}^* \mathbf{Y}_l^{(m)})\right]}{\| \mathbf{S} \|_F^2} \right|_{\omega_l = \hat{\omega}_l}, \tag{11.27}$$

and

$$\hat{\omega}_l = \arg\max_{\omega_l} \left\{ \sum_{m=1}^{M} \mathrm{Re}^2 \left[\mathbf{a}^H(\omega_l)(\mathbf{S}^* \mathbf{Y}_l^{(m)})\right] \right\}. \tag{11.28}$$

11.4.2 Fixed Delays and Gains

When both the delays and gains of the target scatterers remain the same during the multiple look interval, we can derive an ML estimator when the noise is assumed to be a zero-mean colored Gaussian noise with an unknown covariance matrix $\mathbf{Q}$. Note that although we could continue to use the NLS approach for the current problem, we prefer to take the noise statistics into account since we will show below that doing so in this case introduces little difficulties for sufficiently large M. For the former problems, modeling the noise with an unknown covariance matrix $\mathbf{Q}$ makes the ML approach ill-defined due to too many unknowns [22].

Let $\mathbf{Y}^{(m)}$ be the DFT of the received data vector due to the mth pulse which can be written as

$$\mathbf{Y}^{(m)} = \sum_{l=1}^{L} \alpha_l \mathbf{b}(\omega_l) + \mathbf{E}^{(m)}, \quad m = 1, ..., M, \tag{11.29}$$

where the noise vectors $\{\mathbf{E}^{(m)}\}_{m=1}^{M}$ are assumed to be zero-mean colored Gaussian random vectors with an unknown covariance matrix $\mathbf{Q}$ that are independent of each other. Let

$$\mathbf{B} = [\ \mathbf{b}(\omega_1) \quad \mathbf{b}(\omega_2) \quad \cdots \quad \mathbf{b}(\omega_L)\ ]^T, \tag{11.30}$$

and

$$\boldsymbol{\alpha} = [\ \alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_L\ ]^T\ . \tag{11.31}$$

Then

$$\mathbf{Y}^{(m)} = \mathbf{B}\boldsymbol{\alpha} + \mathbf{E}^{(m)}, \quad m = 1, 2, ..., M. \tag{11.32}$$

The log-likelihood function of $\mathbf{Y}^{(m)}$ is proportional to (within an additive constant):

$$-\ln[\det(\mathbf{Q})] - \mathrm{tr}\left\{\mathbf{Q}^{-1}\frac{1}{M}\sum_{m=1}^{M}\left[\mathbf{Y}^{(m)} - \mathbf{B}\boldsymbol{\alpha}\right]\left[\mathbf{Y}^{(m)} - \mathbf{B}\boldsymbol{\alpha}\right]^H\right\} \tag{11.33}$$

where $\det(\cdot)$ denotes the determinant of a matrix and $\mathrm{tr}(\cdot)$ denotes the trace of a matrix. Consider first the estimate of $\mathbf{Q}$ and the unstructured estimate of $\mathbf{C} = \mathbf{B}\boldsymbol{\alpha}$. It is easy to

show that the estimate $\hat{\mathbf{Q}}$ of $\mathbf{Q}$ is

$$\hat{\mathbf{Q}} = \frac{1}{M}\sum_{m=1}^{M}\left[\mathbf{Y}^{(m)} - \mathbf{C}\right]\left[\mathbf{Y}^{(m)} - \mathbf{C}\right]^{H}, \tag{11.34}$$

where $\hat{\mathbf{C}}$ may be obtained by minimizing the following cost function:

$$C_5 = \det\left[\frac{1}{M}\sum_{m=1}^{M}\left(\mathbf{Y}^{(m)} - \mathbf{C}\right)\left(\mathbf{Y}^{(m)} - \mathbf{C}\right)^{H}\right]. \tag{11.35}$$

Let

$$\hat{\mathbf{R}}_{Y1} = \frac{1}{M}\sum_{m=1}^{M}\mathbf{Y}^{(m)}, \tag{11.36}$$

and

$$\hat{\mathbf{R}}_{YY} = \frac{1}{M}\sum_{m=1}^{M}\mathbf{Y}^{(m)}(\mathbf{Y}^{(m)})^{H}. \tag{11.37}$$

Then

$$\begin{aligned}
\mathbf{G} &= \frac{1}{M}\sum_{m=1}^{M}\left[\mathbf{Y}^{(m)} - \mathbf{C}\right]\left[\mathbf{Y}^{(m)} - \mathbf{C}\right]^{H} \\
&= \hat{\mathbf{R}}_{YY} - \mathbf{C}\hat{\mathbf{R}}_{Y1}^{H} - \hat{\mathbf{R}}_{Y1}\mathbf{C}^{H} + \mathbf{C}\mathbf{C}^{H} \\
&= \left[\mathbf{C} - \hat{\mathbf{R}}_{Y1}\right]\left[\mathbf{C} - \hat{\mathbf{R}}_{Y1}\right]^{H} + \hat{\mathbf{R}}_{YY} - \hat{\mathbf{R}}_{Y1}\hat{\mathbf{R}}_{Y1}^{H}.
\end{aligned} \tag{11.38}$$

To minimize $\det(\mathbf{G})$, we have

$$\hat{\mathbf{C}} = \hat{\mathbf{R}}_{Y1}. \tag{11.39}$$

Then using the $\hat{\mathbf{C}}$ in (11.39) to replace the $\mathbf{C}$ in (11.34) yields

$$\hat{\mathbf{Q}} = \hat{\mathbf{R}}_{YY} - \hat{\mathbf{R}}_{Y1}\hat{\mathbf{R}}_{Y1}^{H}. \tag{11.40}$$

With these notations, the above C_5 can be rewritten as

$$\begin{aligned}
C_5 &= \det\left[\hat{\mathbf{R}}_{YY} - \mathbf{C}\hat{\mathbf{R}}_{Y1}^{H} - \hat{\mathbf{R}}_{Y1}\mathbf{C}^{H} + \mathbf{C}\mathbf{C}^{H}\right] \\
&= \det\left[\hat{\mathbf{R}}_{YY} - \hat{\mathbf{R}}_{Y1}\hat{\mathbf{R}}_{Y1}^{H} + (\mathbf{C} - \hat{\mathbf{C}})(\mathbf{C} - \hat{\mathbf{C}})^{H}\right] \\
&= \det(\hat{\mathbf{Q}})\det\left[\mathbf{I} + \hat{\mathbf{Q}}^{-1}(\mathbf{C} - \hat{\mathbf{C}})(\mathbf{C} - \hat{\mathbf{C}})^{H}\right] \\
&= \det(\hat{\mathbf{Q}})\left[1 + (\mathbf{C} - \hat{\mathbf{C}})^{H}\hat{\mathbf{Q}}^{-1}(\mathbf{C} - \hat{\mathbf{C}})\right],
\end{aligned} \tag{11.41}$$

where we have used the fact that $\det(\mathbf{I}+\mathbf{AB}) = \det(\mathbf{I}+\mathbf{BA})$ if the dimensions of $\mathbf{A}$ and $\mathbf{B}$ permit. Hence minimizing C_5 is equivalent to minimizing

$$\begin{aligned} C_6(\{\alpha_l, \omega_l\}_{l=1}^L) &= \left[\mathbf{C} - \hat{\mathbf{C}}\right]^H \hat{\mathbf{Q}}^{-1} \left[\mathbf{C} - \hat{\mathbf{C}}\right] \\ &= \left[\mathbf{B}\boldsymbol{\alpha} - \hat{\mathbf{C}}\right]^H \hat{\mathbf{Q}}^{-1} \left[\mathbf{B}\boldsymbol{\alpha} - \hat{\mathbf{C}}\right], \end{aligned} \tag{11.42}$$

which is again a highly nonlinear optimization problem.

We consider below using the relaxation based approach to minimize $C_6(\{\alpha_l, \omega_l\}_{l=1}^L)$. Let

$$\hat{\mathbf{C}}_l = \hat{\mathbf{C}} - \sum_{i=1, i\neq l}^{L} \hat{\alpha}_i \mathbf{b}(\hat{\omega}_i), \tag{11.43}$$

where $\{\hat{\alpha}_i, \hat{\omega}_i\}_{i=1, i\neq l}^L$ are assumed given. Then minimizing C_6 becomes minimizing

$$C_7(\alpha_l, \omega_l) = \left[\hat{\mathbf{C}}_l - \alpha_l \mathbf{b}(\omega_l)\right]^H \hat{\mathbf{Q}}^{-1} \left[\hat{\mathbf{C}}_l - \alpha_l \mathbf{b}(\omega_l)\right]. \tag{11.44}$$

Consider first the case of complex-valued $\{\alpha_l\}_{l=1}^L$. Minimizing $C_7(\alpha_l, \omega_l)$ with respect to α_l and ω_l yields:

$$\begin{aligned} \hat{\alpha}_l &= \left.\frac{\mathbf{b}^H \hat{\mathbf{Q}}^{-1} \hat{\mathbf{C}}_l}{\mathbf{b}^H(\omega_l) \hat{\mathbf{Q}}^{-1} \mathbf{b}(\omega_l)}\right|_{\omega_l = \hat{\omega}_l} \\ &= \left.\frac{\mathbf{a}^H(\omega_l) \left(\mathbf{S}^* \hat{\mathbf{Q}}^{-1} \hat{\mathbf{C}}_l\right)}{\left\|\hat{\mathbf{Q}}^{-\frac{1}{2}} \mathbf{S}\mathbf{a}(\omega_l)\right\|^2}\right|_{\omega_l = \hat{\omega}_l}, \end{aligned} \tag{11.45}$$

and

$$\hat{\omega}_l = \arg\max_{\omega_l} \frac{\left|\mathbf{a}^H(\omega_l) \mathbf{S}^* \hat{\mathbf{Q}}^{-1} \hat{\mathbf{C}}_l\right|^2}{\left\|\hat{\mathbf{Q}}^{-\frac{1}{2}} \mathbf{S}\mathbf{a}(\omega_l)\right\|^2}. \tag{11.46}$$

Consider next the case of real-valued $\{\alpha_l\}_{l=1}^L$. Minimizing $C_7(\alpha_l, \omega_l)$ with respect to α_l and ω_l yields

$$\hat{\alpha}_l = \left.\frac{\mathrm{Re}\left[\mathbf{a}^H(\omega_l) \mathbf{S}^* \hat{\mathbf{Q}}^{-1} \hat{\mathbf{C}}_l\right]}{\left\|\hat{\mathbf{Q}}^{-\frac{1}{2}} \mathbf{S}\mathbf{a}(\omega_l)\right\|^2}\right|_{\omega_l = \hat{\omega}_l}, \tag{11.47}$$

and

$$\hat{\omega}_l = \arg\max_{\omega_l} \frac{\mathrm{Re}^2\left[\mathbf{a}^H(\omega_l)\mathbf{S}^*\hat{\mathbf{Q}}^{-1}\hat{\mathbf{C}}_l\right]}{\left\|\hat{\mathbf{Q}}^{-\frac{1}{2}}\mathbf{S}\mathbf{a}(\omega_l)\right\|^2}. \tag{11.48}$$

In the above derivations, $\hat{\mathbf{Q}}^{-1}$ plays the role of whitening the noise. A good estimate of $\mathbf{Q}$ requires a large number of independent data vectors (i.e., M should be large enough as compared with N). When M is small, the noise covariance matrix estimated from (11.40) is singular or near singular. At least N data vectors are needed to guarantee that the matrix $\hat{\mathbf{Q}}$ is non-singular with probability one.

Usually, the receiver noise $e(t)$ in (11.1) can be modeled as a zero-mean stationary and ergodic Gaussian stochastic process. Let the covariance matrix corresponding to the sampled noise vector $[e(0),\ e(T_s)\ \cdots\ e((N-1)T_s)]^T$ be $\mathbf{Q}_t$, where the subscript "t" represents the covariance matrix of the noise in the time domain. Then $\mathbf{Q}_t$ is a Hermitian and Toeplitz matrix. The frequency domain noise covariance matrix $\mathbf{Q}$ is related to $\mathbf{Q}_t$ as follows

$$\mathbf{Q} = \mathbf{\Gamma}\ \mathbf{Q}_t\ \mathbf{\Gamma}^H, \tag{11.49}$$

where $\mathbf{\Gamma}$ is the DFT matrix,

$$\mathbf{\Gamma} = \frac{1}{\sqrt{N}}\left[\ \mathbf{a}(-\pi)\quad \mathbf{a}(-\pi+2\pi/N)\quad \cdots\quad \mathbf{a}(\pi-2\pi/N)\ \right]^H, \tag{11.50}$$

and

$$\mathbf{a}(\omega) = \left[\ e^{j\omega(-N/2)}\quad e^{j\omega(-N/2+1)}\quad \ldots\quad e^{j\omega(N/2-1)}\ \right]^T. \tag{11.51}$$

It can be shown that, in general, $\mathbf{Q}$ is no longer a Toeplitz matrix. However, we can use the Toeplitz property of $\mathbf{Q}_t$ to improve the estimation performance. First, we can obtain $\hat{\mathbf{Q}}$ by using (11.40). Then the estimate $\hat{\mathbf{Q}}_t$ of $\mathbf{Q}_t$ can be obtained by using (11.49), which is

$$\hat{\mathbf{Q}}_t = \mathbf{\Gamma}^H\ \hat{\mathbf{Q}}\ \mathbf{\Gamma}. \tag{11.52}$$

Due to a finite number of data vectors, $\hat{\mathbf{Q}}_t$ is no longer a Toeplitz matrix. Although there are many ways to to modify $\hat{\mathbf{Q}}_t$ to obtain a Toeplitz matrix $\hat{\mathbf{Q}}_t^{(T)}$, in this paper we use the

following simple approach. Let $\hat{q}_t(i,j)$ be the (i,j)th element of $\hat{\mathbf{Q}}_t$. Define

$$\hat{r}(k) = \frac{1}{N-k}\sum_{i=1}^{N-k}\hat{q}_t(i,i+k), \quad k = 0,1,\cdots,N-1. \tag{11.53}$$

Then

$$\hat{\mathbf{Q}}_t^{(T)} = \begin{bmatrix} \hat{r}(0) & \hat{r}(1) & \cdots & \hat{r}(N-1) \\ \hat{r}^*(1) & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \hat{r}(1) \\ \hat{r}^*(N-1) & \cdots & \hat{r}^*(1) & \hat{r}(0) \end{bmatrix}. \tag{11.54}$$

Using $\hat{\mathbf{Q}}^{(T)}$ instead of $\hat{\mathbf{Q}}$ in (11.45)-(11.48), where

$$\hat{\mathbf{Q}}^{(T)} = \mathbf{\Gamma}\hat{\mathbf{Q}}_t^{(T)}\mathbf{\Gamma}^H, \tag{11.55}$$

we obtain a new algorithm referred to as TWRELAX. The TWRELAX algorithm can greatly improve the estimation performance of WRELAX, especially when M is small as compared with N, as can be seen from the numerical examples below.

11.5 Simulation and Experimental Results

In this section, we will present numerical and experimental examples to demonstrate the performance of the proposed WRELAX algorithm. We first present several numerical examples for both single look and multiple look cases. The performance of the proposed algorithms is compared with the EM algorithm [13] and the CRB, which gives the minimum attainable variances for any unbiased estimators. Next, we test the new algorithm with the experimental data collected to analyze the subsurface structures of highways.

In the numerical examples below, we use a windowed chirp signal,

$$s(t) = w(t)e^{j\beta(t-\frac{T_0}{2})^2}, \quad 0 \le t \le T_0, \tag{11.56}$$

where β is the chirp rate and

$$w(t) = \begin{cases} 0.5 - 0.5\cos(\pi t/T_w), & 0 < t < T_w, \\ 1, & T_w \le t \le T_0 - T_w, \\ 0.5 - 0.5\cos[\pi(t-T_0)/T_w], & T_0 - T_w < t \le T_0\ , \end{cases} \tag{11.57}$$

with $T_w = T_0/10$.

In the following simulations, we use $N = 64$, $\beta = \pi \times 10^{12}$, the signal bandwidth $B_s = \beta T_0/\pi$, and the sampling frequency $f_s = 2B_s$ (to avoid aliasing). T_0 is chosen in such a way that $T_0 = (N/2 - 1)T_s$. In this case, it can be shown that $T_0 = \sqrt{\frac{(N-1)\pi}{2\beta}} = 3.937\ \mu s$, $T = 8.001\ \mu s$, $T_s = 0.127\ \mu s$, $B_s = 3.937$ MHz, $f_s = 7.874$ MHz, and the resolution limit of the conventional matched filter method is generally considered to be around $\tau_e = 1/B_s = 0.254\ \mu s$.

In all of the examples below, we have used $\epsilon = 0.001$ to test the convergence of WRELAX. All data sequences are zero-padded to the nearest power of 2. For the simulation examples, we have $N = 64$, so zero-padding is not used. The one-dimensional search is performed in two steps, a coarse search using FFT followed by a fine search using the *fmin* function of MATLAB. In all of the simulation examples, complex valued gains are assumed. Real valued gains are used for the experimental example. The SNR of each signal replica is defined to be $10\log_{10}(|\alpha|^2/\sigma_e^2)$ for the case of fixed gains, where σ_e^2 denotes the average noise power. For the case of arbitrary gains, $\{\alpha_l^{(m)}\}_{m=1}^M$ are assumed to be independent complex Gaussian random variables with zero-mean and variance σ_α^2 (they are fixed for all Monte-Carlo trials) and the SNR is defined to be $10\log_{10}[\sigma_\alpha^2/\sigma_e^2]$. The MSE is obtained through 100 Monte-Carlo trials.

Case A: Single Look

Assume $L = 2$ signals are superimposed together with $\alpha_1 = e^{j\pi/8}$, $\alpha_2 = e^{j\pi/4}$, $\tau_1 = T_0/8$, $\tau_2 = T_0/8 + \tau_e$. The additive noise is zero-mean white Gaussian and SNR = $\text{SNR}_1 = \text{SNR}_2 = 10$ dB. Hence the time delay spacing between the two signals is τ_e. Even in this case, the conventional matched filter method fails to resolve the two signals, as can be seen from Figure 11.1(a), where the horizontal axis denotes the normalized time delay τ/T and the two vertical lines indicate the true time delays of the two signals. However, using WRELAX we can resolve them very well. As pointed out before, the WRELAX algorithm can be viewed as transforming the multi-dimensional matched filters into a sequence of one-dimensional matched filters. The outputs of the two matched filters for all iterations are plotted in Figure 11.1(b), which illustrates the convergence process of WRELAX. At the

beginning of the iteration, the peak positions and the corresponding gain estimates obtained from the filter outputs differ from their true values (one gain estimate is larger and the other is smaller than the corresponding true gains). After several steps, they converge to the true time delays and gains.

Now we change the time delay spacing of the two signals to $0.5\tau_e$ with the same gains as above. The MSEs using WRELAX are compared with the corresponding CRBs in Figure 11.2, where $\text{SNR} = \text{SNR}_1 = \text{SNR}_2$. From Figure 11.2, it can be noted that the MSEs obtained by using WRELAX approach the corresponding CRBs as the SNR increases.

We have also tested the EM algorithm for this situation and found that it is very sensitive to the initial conditions used. If we use the conventional matched filter approach to obtain the initial conditions, EM will converge to some local minimum instead of the global one and the estimation performance is very poor. When we skip the first $L-1$ steps of WRELAX and use the same initial values used for EM, WRELAX converges much faster than EM. The speedup ratio is 2.5 for the same signal used in Figure 11.2. If we add one more signal that is separated from the above two signals by τ_e and $1.5\tau_e$ then the speedup ratio goes up to 4.0.

Case B: Multiple Looks with Fixed Delays but Arbitrary Gains

Consider the case of multiple looks with fixed time delays but arbitrary gains when the noise is a zero-mean white Gaussian noise with variance σ_e^2. In this example, $L = 2$, $\tau_1 = T_0/8$, and $\tau_2 = T_0/8 + 0.5\tau_e$. We use two data vectors obtained by $M = 2$ looks to estimate the fixed delays. The gains of the signals are generated randomly but fixed from trial to trial and $\sigma_\alpha^2 = 1.0$. This case is similar to the data model used by the deterministic (or conditional) ML method for DOA estimation in array signal processing [5]. We assume $\text{SNR} = \text{SNR}_1 = \text{SNR}_2$ and the generated arbitrary gains in this example are $\alpha_1^{(1)} = -0.5111 + 0.3922j$, $\alpha_2^{(1)} = 0.9157 - 0.4958j$, $\alpha_1^{(2)} = 0.3268 + 0.6921j$ and $\alpha_2^{(2)} = -1.8936 - 0.3220j$. We use the extended WRELAX algorithm presented in Section 4.1 to estimate the delays and the MSEs are compared with the corresponding CRBs in Figure 11.3. It can be noted that the MSEs again approach the CRBs as SNR increases.

Case C: Multiple Looks with Fixed Delays and Gains

Now we consider the problem of estimating fixed delays and gains of two signals from multiple data vectors collected by multiple looks in a colored Gaussian noise environment. In this example, the colored noise is modeled as a first-order autoregressive (AR) process with coefficient a_1=-0.85. The delays and gains of the two signals are the same as those in Figure 11.2 and SNR = SNR_1 = SNR_2 = 0 dB. The MSEs of WRELAX and TWRELAX are shown and compared with the CRBs in Figure 11.4 as a function of the normalized multiple look numbers $\log_2(M/N)$. From Figure 11.4, it can be noted that the improvement of TWRELAX over WRELAX is significant, especially for small M as compared to N. For this example, when $M = N/2$, the MSEs of TWRELAX are very close to the CRBs, while $M = 4N$ is required before the MSEs of WRELAX approach the CRBs.

Case D: Application to Ultra Wideband Ground Penetrating Radar

The detection and classification of roadway subsurface anomalies are very important for the design and quality evaluation of highways. Ultra wideband ground penetrating radar is very suitable for this application because of its extremely large bandwidth (several GHz) and high range resolution (on the order of several centimeters). The returned echoes of the ultra wideband ground penetrating radar are superimposed signals reflected from the boundaries of different media (layers, voids, etc.), which can be described by (11.1). Unlike the data models used in the above simulations, the probing signal $s(t)$ and the gains here are all real. Both the delays and gains are very useful for the detection and classification of roadway subsurface anomalies. The delays can be used to determine the layer thickness or anomaly location and the gains can be used to classify the type of media because the gains are related to the reflection coefficient at the boundary between two media with different dielectric constants. Once we get the estimates of the media dielectric constants, we can judge the type of the media.

The current method of detecting and classifying roadway subsurface anomalies requires manual inspection of each radar trace by a qualified engineer or technician. This method is neither accurate nor practical. Although the range resolution of the ultra wideband ground penetrating radar is pretty high, it is still very difficult, if not impossible, to identify closely spaced echoes from different layers by visual examination. Yet the closely spaced echoes

may be more important for the detection and classification of the anomalies. Due to the voluminous amount of data collected, manual inspection seems not feasible. However, using the proposed WRELAX algorithm, the detection and classification can be implemented automatically with high accuracy. In this paper, we will only consider the estimation of the delays and gains from the experimental data collected by an ultra wideband ground penetrating radar. The classification and many other practical issues (such as end reflection removal and sensor motion compensation) are beyond the scope of this paper.

The sampled version of the signal waveform $s(t)$ is depicted in Figure 11.5(a) as a function of the sample points, where the sampling interval is T_s=0.07 ns. No explicit expression is available for the transmitted signal $s(t)$, which can only be measured by specially designed experiment. The discrete time Fourier transform (magnitude) of the signal in Figure 11.5(a) is shown in Figure 11.5(b), where $f_s = 1/T_s$=14.28 GHz. From Figure 11.5(b), it can be seen that the signal spectrum covers a wide range (from 0 to 2.5 GHz). Figure 11.5(c) shows the autocorrelation function (magnitude) of the signal $s(t)$ in Figure 11.5(a), from which high sidelobes can be observed. These high sidelobes will greatly degrade the performance of the conventional matched filter approach, as can be seen from the matched filter output (shown in Figure 11.5(d)) of the sampled version of the observed signal $y(t)$ (shown in Figure 11.5(e)). We assume that there are $L = 5$ reflected signals coming from five layers. Using WRELAX we get the estimates of the delays (0.035 ns, 0.109 ns, 0.464 ns, 0.716 ns, 4.5 ns) and the gains (1.1644, -0.2883, 0.4171, 0.2711, -0.0460). The reconstructed signal returned by each layer is shown in Figures 11.5(f) through (j). The reconstructed superimposed signals obtained by using the estimates of WRELAX and those of the matched filter approach are compared with the observed signal in Figures 11.5(k) and (l), respectively. Note that WRELAX significantly outperforms the conventional matched filter approach.

11.6 Conclusions

In this chapter, we have proposed a weighted Fourier transform based WRELAX algorithm for the well-known time delay estimation problem. By avoiding the computationally demanding multidimensional search over the parameter space, WRELAX minimizes the NLS

criterion at a much lower implementation cost. It is more efficient and systematic than the somewhat similarly structured EM algorithm. The WRELAX algorithm is also successfully applied to the detection and classification of roadway subsurface anomalies and extended to two cases of multiple looks.

Appendix: Derivation of CRBs

We sketch below the derivation of the CRBs for the parameter estimates of the following data model

$$\mathbf{Y}^{(m)} = \mathbf{\Omega}\boldsymbol{\alpha}^{(m)} + \mathbf{E}^{(m)}, \quad m = 1, 2, ..., M, \tag{11.58}$$

where

$$\mathbf{\Omega} = \mathbf{SA} \tag{11.59}$$

$$\mathbf{A} = \left[\begin{array}{cccc} \mathbf{a}(\tau_1) & \mathbf{a}(\tau_2) & \cdots & \mathbf{a}(\tau_L) \end{array} \right], \tag{11.60}$$

$$\mathbf{a}(\tau_l) = \left[\begin{array}{cccc} \exp\left[-j\frac{2\pi\tau_l}{NT_s}(-\frac{N}{2})\right] & \exp\left[-j\frac{2\pi\tau_l}{NT_s}(-\frac{N}{2}+1)\right] & \cdots & \exp\left[-j\frac{2\pi\tau_l}{NT_s}(\frac{N}{2}-1)\right] \end{array} \right]^T, \tag{11.61}$$

and

$$\boldsymbol{\alpha}^{(m)} = \left[\begin{array}{cccc} \alpha_1^{(m)} & \alpha_2^{(m)} & \cdots & \alpha_L^{(m)} \end{array} \right]^T. \tag{11.62}$$

In (11.58) the additive noise vectors $\mathbf{E}^{(m)}$, $m = 1, 2, \cdots, M$, are assumed to be zero-mean Gaussian random vectors with an unknown covariance matrix $\mathbf{Q}$ that are independent of each other. For convenience, we denote the three data models (11.4), (11.21), and (11.29) by Cases A, B, and C, respectively. Cases A and C can be viewed as special cases of the above data model, which corresponds to Case B. For Case A, we have $M = 1$, $\mathbf{Y}^{(1)} = \mathbf{Y}$, $\mathbf{E}^{(1)} = \mathbf{E}$, and $\boldsymbol{\alpha}^{(1)} = \boldsymbol{\alpha} = \left[\begin{array}{cccc} \alpha_1 & \alpha_2 & \cdots & \alpha_L \end{array} \right]^T$. For Case C, we have $\boldsymbol{\alpha}^{(m)} = \boldsymbol{\alpha}^{(1)} = \boldsymbol{\alpha}$, $m = 1, 2, \cdots, M$.

Let

$$\tilde{\mathbf{Y}} = \left[\begin{array}{cccc} (\mathbf{Y}^{(1)})^T & (\mathbf{Y}^{(2)})^T & \cdots & (\mathbf{Y}^{(M)})^T \end{array} \right]^T, \tag{11.63}$$

and

$$\tilde{\mathbf{E}} = \left[\begin{array}{cccc} (\mathbf{E}^{(1)})^T & (\mathbf{E}^{(2)})^T & \cdots & (\mathbf{E}^{(M)})^T \end{array} \right]^T. \tag{11.64}$$

Then

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{\Omega}}\tilde{\boldsymbol{\alpha}} + \tilde{\mathbf{E}}, \tag{11.65}$$

where

$$\tilde{\mathbf{\Omega}} = \begin{cases} \mathbf{1}_M \otimes \mathbf{\Omega}, & \text{for Cases A and C}, \\ \mathbf{I}_M \otimes \mathbf{\Omega}, & \text{for Case B}, \end{cases} \tag{11.66}$$

and

$$\tilde{\boldsymbol{\alpha}} = \begin{cases} \boldsymbol{\alpha}, & \text{for Cases A and C}, \\ \begin{bmatrix} (\boldsymbol{\alpha}^1)^T & (\boldsymbol{\alpha}^2)^T & \cdots & (\boldsymbol{\alpha}^M)^T \end{bmatrix}^T, & \text{for Case B}, \end{cases} \tag{11.67}$$

with $\mathbf{1}_M = [1\ 1\ \cdots\ 1]^T$, $\mathbf{I}_M$ denoting the $M \times M$ identity matrix, and $\otimes$ denoting the Kronecker product [17]. Let $\tilde{\mathbf{Q}} = E\{\tilde{\mathbf{E}}\tilde{\mathbf{E}}^H\}$ be the covariance matrix of $\tilde{\mathbf{E}}$. It follows that

$$\tilde{\mathbf{Q}} = \mathbf{I}_M \otimes \mathbf{Q}. \tag{11.68}$$

The unknown variables in the likelihood function of $\tilde{\mathbf{Y}}$ are the unknown elements of $\tilde{\mathbf{Q}}$, the real and imaginary parts of the gains (for complex valued gains) or simply the gains (for real valued gains), and the delays. The extended Slepian-Bangs' formula for the ijth element of the Fisher information matrix has the form [23, 24]:

$$\{\text{FIM}\}_{ij} = \text{tr}\left(\tilde{\mathbf{Q}}^{-1}\tilde{\mathbf{Q}}_i'\tilde{\mathbf{Q}}^{-1}\tilde{\mathbf{Q}}_j'\right) + 2\text{Re}\left[\left(\tilde{\boldsymbol{\alpha}}^H\tilde{\mathbf{\Omega}}^H\right)_i'\tilde{\mathbf{Q}}^{-1}\left(\tilde{\mathbf{\Omega}}\tilde{\boldsymbol{\alpha}}\right)_j'\right], \tag{11.69}$$

where $\mathbf{X}_i'$ denotes the derivative of $\mathbf{X}$ with respect to the ith unknown parameter. Note that FIM is a block diagonal matrix since $\tilde{\mathbf{Q}}$ does not depend on the parameters in $(\tilde{\mathbf{\Omega}}\tilde{\boldsymbol{\alpha}})$, and $(\tilde{\mathbf{\Omega}}\tilde{\boldsymbol{\alpha}})$ does not depend on the elements of $\tilde{\mathbf{Q}}$. Hence the CRBs of the estimates of the delays and gains can be determined from the second term of the right side of (11.69).

Next we derive the CRBs for the case of complex valued gains (which correspond to our simulation examples), and the CRBs for the case of real valued gains can be derived similarly. Let

$$\boldsymbol{\eta} = \begin{bmatrix} \text{Re}^T(\tilde{\boldsymbol{\alpha}}) & \text{Im}^T(\tilde{\boldsymbol{\alpha}}) & \boldsymbol{\tau}^T \end{bmatrix}^T, \tag{11.70}$$

where $\text{Im}(\mathbf{X})$ denotes the imaginary part of $\mathbf{X}$ and

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 & \tau_2 & \cdots & \tau_L \end{bmatrix}^T. \tag{11.71}$$

Let

$$\tilde{\mathbf{F}} = \left[\begin{array}{ccc} \tilde{\mathbf{\Omega}} & j\tilde{\mathbf{\Omega}} & \tilde{\mathbf{D}}_\tau \tilde{\mathbf{\Phi}} \end{array} \right]. \tag{11.72}$$

where

$$\tilde{\mathbf{D}}_\tau = \begin{cases} \mathbf{I}_M \otimes \mathbf{D}_\tau, & \text{for Cases A and C}, \\ \mathbf{I}_M \otimes \mathbf{D}_\tau, & \text{for Case B.} \end{cases} \tag{11.73}$$

with

$$\mathbf{D}_\tau = \mathbf{S} \left[\begin{array}{cccc} \frac{d\mathbf{a}(\tau_1)}{d\tau_1} & \frac{d\mathbf{a}(\tau_2)}{d\tau_2} & \cdots & \frac{d\mathbf{a}(\tau_L)}{d\tau_L} \end{array} \right], \tag{11.74}$$

and

$$\tilde{\mathbf{\Phi}} = \begin{cases} \mathbf{\Phi}, & \text{for Cases A and C}, \\ \left[\begin{array}{cccc} (\mathbf{\Phi}^{(1)})^T & (\mathbf{\Phi}^{(2)})^T & \cdots & (\mathbf{\Phi}^{(M)})^T \end{array} \right]^T, & \text{for Case B}, \end{cases} \tag{11.75}$$

with

$$\mathbf{\Phi} = \text{diag} \left[\begin{array}{cccc} \alpha_1 & \alpha_2 & \cdots & \alpha_L \end{array} \right], \tag{11.76}$$

and

$$\mathbf{\Phi}^{(m)} = \text{diag} \left[\begin{array}{cccc} \alpha_1^{(m)} & \alpha_2^{(m)} & \cdots & \alpha_L^{(m)} \end{array} \right], \quad m = 1, 2, \cdots, M. \tag{11.77}$$

Then

$$\text{CRB}(\boldsymbol{\eta}) = \left[2\text{Re}(\tilde{\mathbf{F}}^H \tilde{\mathbf{Q}}^{-1} \tilde{\mathbf{F}}) \right]^{-1}. \tag{11.78}$$

After some simple manipulations, we obtain the following more compact forms of the CRBs for Cases A, B, and C.

$$\text{CRB}(\boldsymbol{\eta}) = \left[\begin{array}{ccc} \text{Re}(\mathbf{\Delta}_1) & -\text{Im}(\mathbf{\Delta}_1) & \text{Re}(\mathbf{\Delta}_2) \\ \text{Im}(\mathbf{\Delta}_1) & \text{Re}(\mathbf{\Delta}_1) & \text{Im}(\mathbf{\Delta}_2) \\ \text{Re}^T(\mathbf{\Delta}_2) & -\text{Im}^T(\mathbf{\Delta}_2) & \Delta b f_3 \end{array} \right]^{-1}, \tag{11.79}$$

where

$$\mathbf{\Delta}_1 = \begin{cases} 2M\mathbf{\Omega}^H \mathbf{Q}^{-1} \mathbf{\Omega}, & \text{for Cases A and C}, \\ 2\, \mathbf{I}_M \otimes (\mathbf{\Omega}^H \mathbf{Q}^{-1} \mathbf{\Omega}), & \text{for Case B}, \end{cases} \tag{11.80}$$

$$\mathbf{\Delta}_2 = \begin{cases} 2M\mathbf{\Omega}^H \mathbf{Q}^{-1} \mathbf{D}_\tau \mathbf{\Phi}, & \text{for Cases A and C}, \\ 2\left[\mathbf{I}_M \otimes (\mathbf{\Omega}^H \mathbf{Q}^{-1} \mathbf{D}_\tau) \right] \tilde{\mathbf{\Phi}}, & \text{for Case B}, \end{cases} \tag{11.81}$$

and

$$\boldsymbol{\Delta}_3 = \begin{cases} 2M\mathrm{Re}\left(\boldsymbol{\Phi}^H \mathbf{D}_\tau^H \mathbf{Q}^{-1} \mathbf{D}_\tau \boldsymbol{\Phi}\right), & \text{for Cases A and C.} \\ 2\sum_{m=1}^{M} \mathrm{Re}\left[(\boldsymbol{\Phi}^{(m)})^H \mathbf{D}_\tau^H \mathbf{Q}^{-1} \mathbf{D}_\tau \boldsymbol{\Phi}^{(m)}\right], & \text{for Case B .} \end{cases} \tag{11.82}$$

Reference

[1] J. E. Ehrenberg, T. E. Ewatt, and R. D. Morris, "Signal processing techniques for resolving individual pulses in multipath signal," *Journal of the Acoustics Society of America*, vol. 63, pp. 1861–1865, January 1978.

[2] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. AP-34, pp. 276–280, March 1986.

[3] A. M. Bruckstein, T. J. Shan, and T. Kailath, "The resolution of overlapping echoes," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-33, pp. 1357–1367, December 1985.

[4] D. W. Tufts, R. Kumaresan, and I. Kirsteins, "Data adaptive signal estimation by singular value decomposition of a data matrix," *Proceedings of the IEEE*, vol. 70, pp. 684–685, June 1982.

[5] P. Stoica and A. Nehorai, "Performance study of conditional and unconditional direction-of-arrival estimation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-38, pp. 1783–1795, October 1990.

[6] Y. Bian and D. Last, "Eigen-decomposition techniques for Loran-C skywave estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, pp. 117–124, January 1997.

[7] I. P. Kirsteins, "High resolution time delay estimation," *Proceedings of ICASSP 87*, pp. 451–454, April 1987.

[8] I. P. Kirsteins and A. C. Kot, "Performance analysis of a high resolution time delay estimation algorithm," *Proceedings of ICASSP 90*, pp. 2767–2770, April 1990.

[9] B. M. Bell and T. E. Ewart, "Separating multipaths by global optimization of multidimensional matched filter," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. ASSP-34, pp. 1029–1037, October 1986.

[10] A. D. Blackowiak and S. D. Rajan, "Multipath arrival estimates using simulated annealing: Application to crosshole tomography experiment," *IEEE Journal of Oceanic Engineering*, vol. 20, pp. 157–165, July 1995.

[11] T. G. Manickam, R. J. Vaccaro, and D. W. Tufts, "A least-squares algorithm for multipath time-delay estimation," *IEEE Transactions on Signal Processing*, vol. 42, pp. 3229–3233, November 1994.

[12] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, pp. 47–60, November 1996.

[13] M. Feder and E. Weinstein, "Parameter estimation of superimposed signals using the EM algorithm," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 36, pp. 477–489, April 1988.

[14] R. Roy, A. Paulraj, and T. Kailath, "ESPRIT – A subspace rotation approach to estimation of parameters of cisoids in noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, pp. 1340–1342, October 1986.

[15] S. M. Kay, *Modern Spectral Estimation: Theory and Application.* Englewood Cliffs, N.J.: Prentice-Hall, 1988.

[16] J. Li and P. Stoica, "Efficient mixed-spectrum estimation with applications to target feature extraction," *IEEE Transactions on Signal Processing*, vol. 44, pp. 281–295, February 1996.

[17] G. W. Stewart, *Introduction to Matrix Computations.* New York, NY: Academic Press, Inc., 1973.

[18] V. G. Karmanov, *Programmation Mathematique.* Editions Mir, Moscow, 1977.

[19] *TMC2310*, TRW LSI Products, Inc.

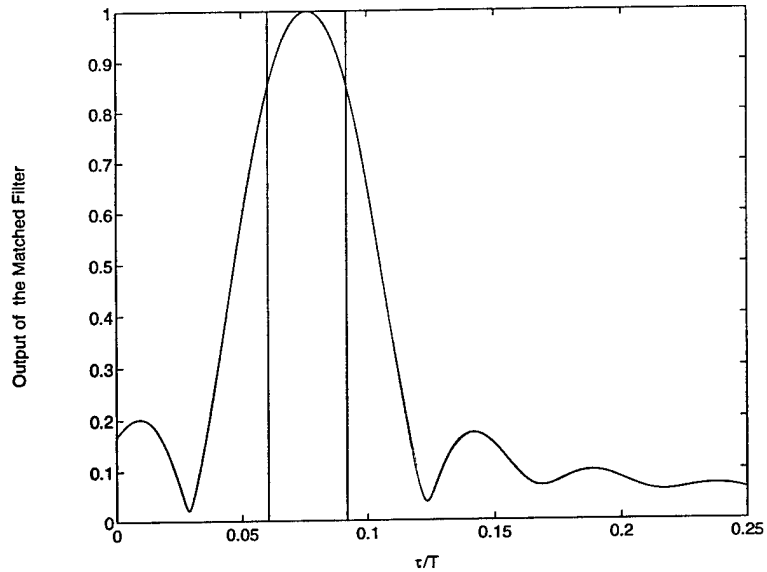
[20] *A Users' Guide for the A41102*, Austeck Microsystems, 1988.

[21] D. K. Barton, *Modern Radar System Analysis.* Norward, MA: Artech House Inc., 1988.

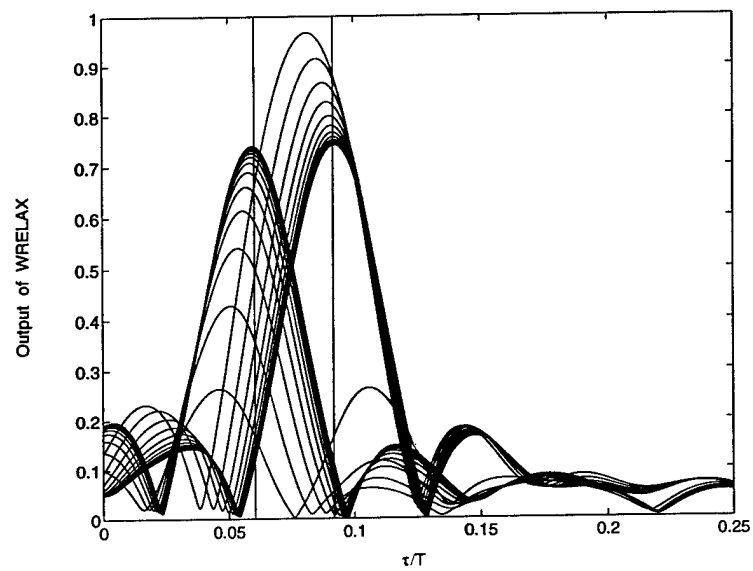
[22] J. Li, B. Halder, P. Stoica, and M. Viberg, "Computationally efficient angle estimation for signals with known waveforms," *IEEE Transactions on Signal Processing*, vol. 43, pp. 2154–2163, September 1995.

[23] W. Bangs, *Array processing with generalized beamformers.* Ph.D. dissertation, Yale University, New Haven, CT, 1971.

[24] P. Stoica and R. L. Moses, *Introduction to Spectral Analysis.* Englewood Cliffs, NJ: Prentice-Hall, 1997.



(a)



(b)

Figure 11.1: Illustrative comparison of WRELAX with the matched filter method. The vertical lines denote the true time delays. (a) The output of the matched filter. (b) The outputs of the two decoupled matched filters associated with WRELAX for all iterations.

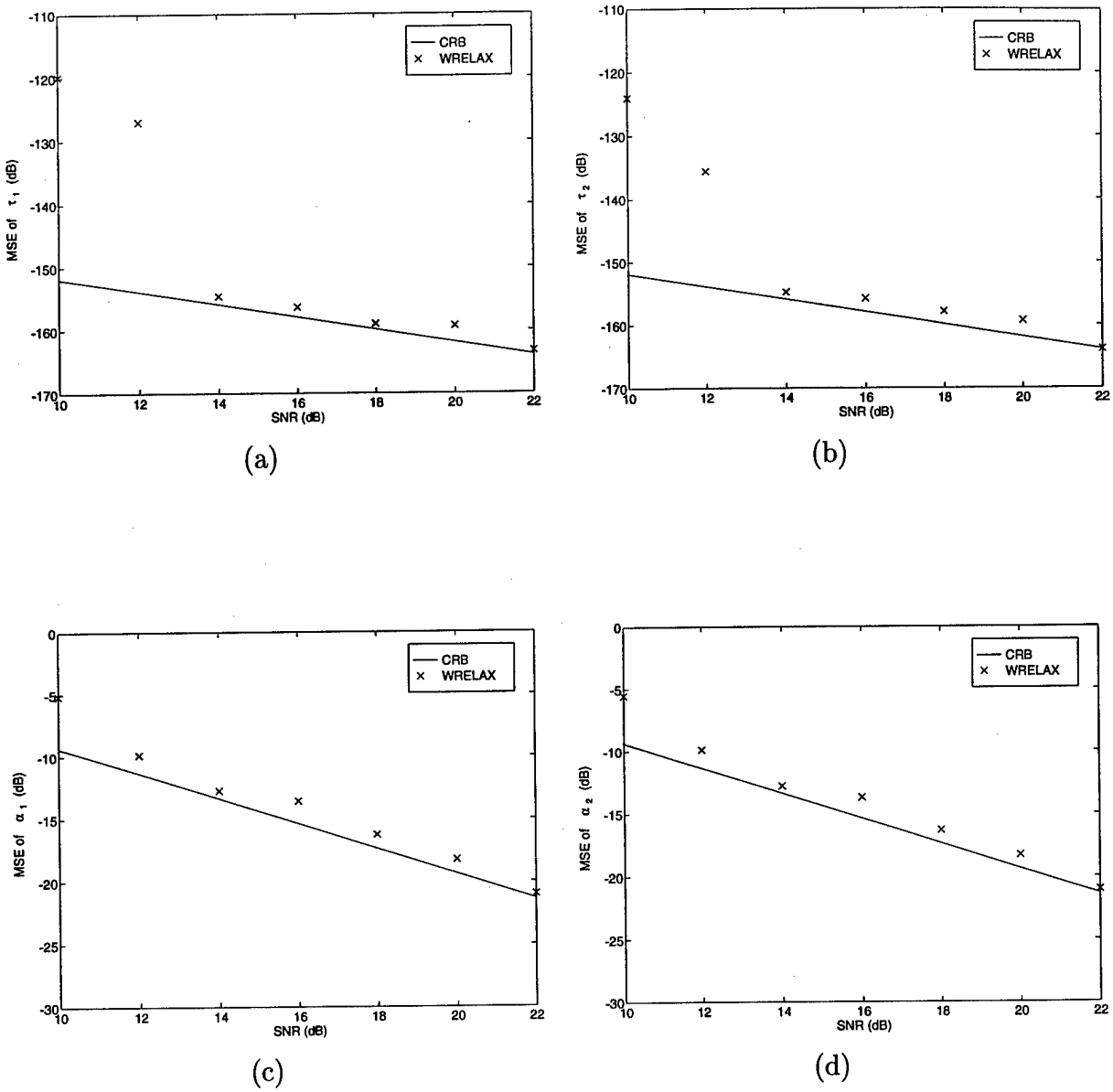
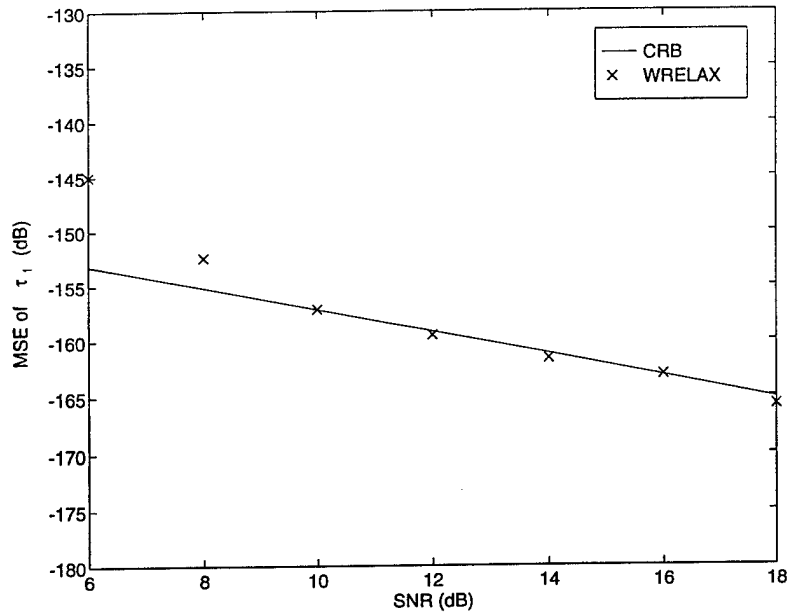
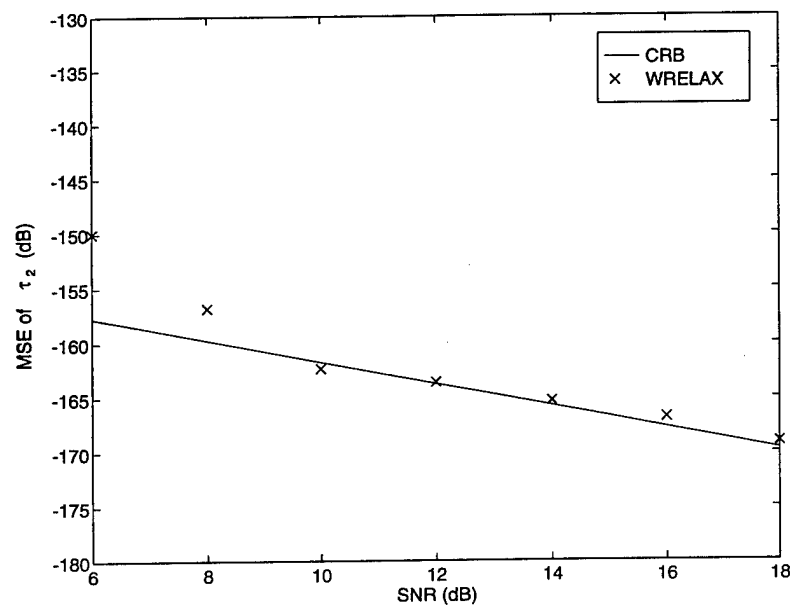


Figure 11.2: MSEs ("×") of WRELAX and CRBs (solid line) for (a) τ_1 , (b) τ_2, (c) α_1, and (d) α_2 for the single look case.



(a)



(b)

Figure 11.3: MSEs ("×") of WRELAX and CRBs (solid line) for (a) τ_1 and (b) τ_2 for multiple looks with fixed delays but arbitrary gains.

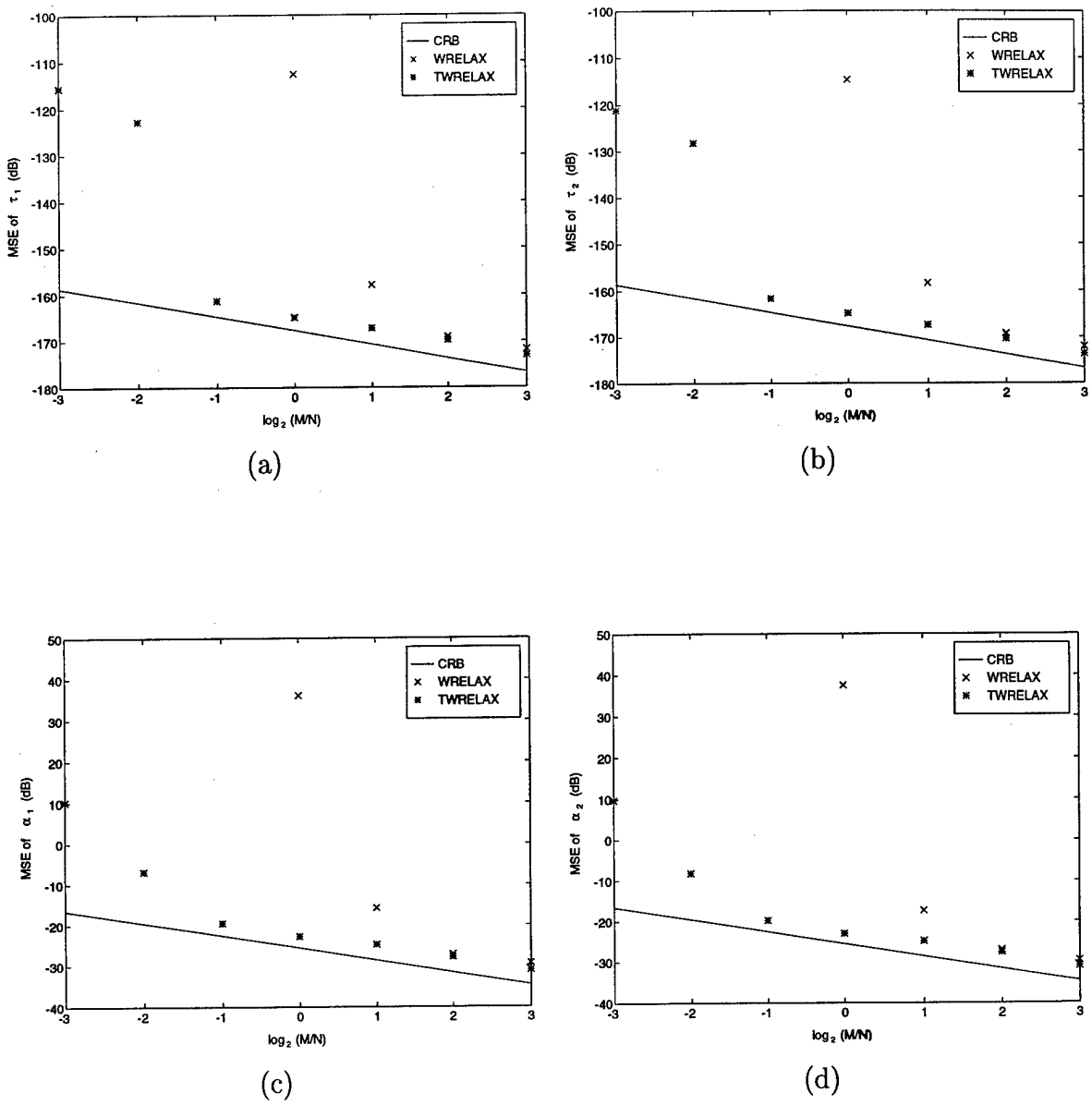
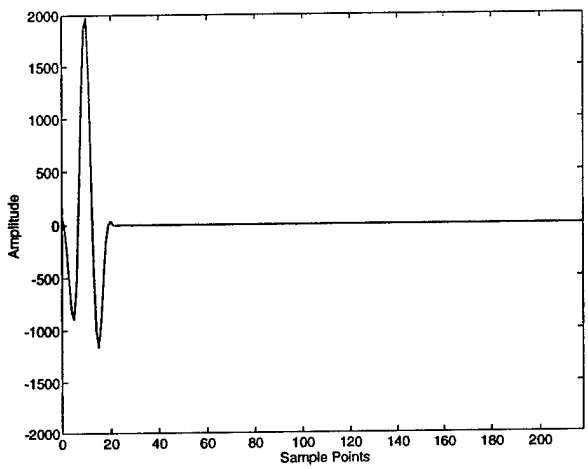
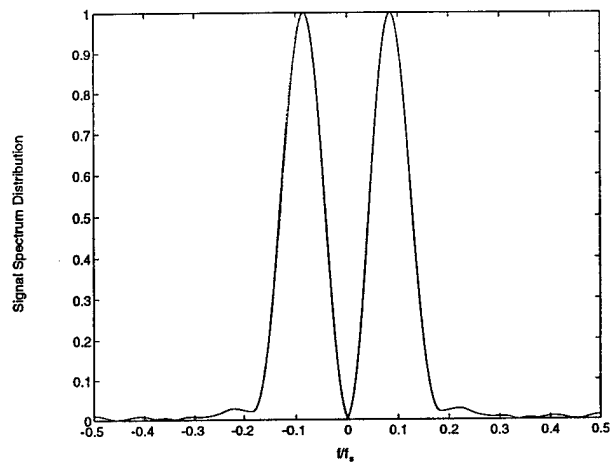


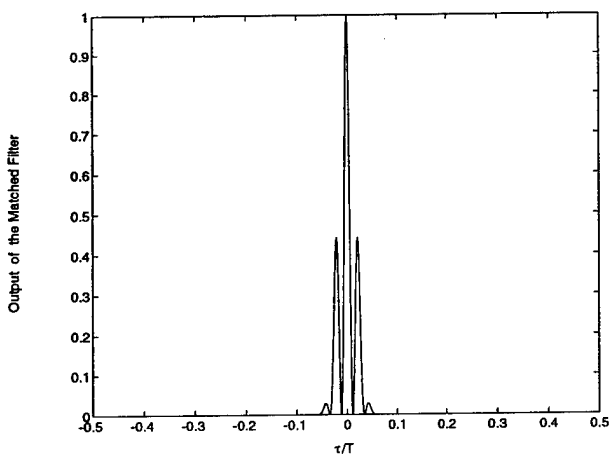
Figure 11.4: MSEs of WRELAX ("×"), TWRELAX ("*"), and CRBs (solid line) for (a) τ_1 , (b) τ_2, (c) α_1, and (d) α_2 for multiple looks with fixed delays and gains.


Amplitude
Sample Points

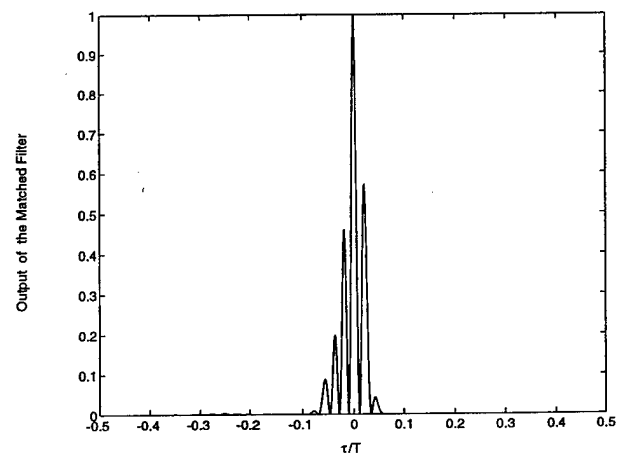
(a)


Signal Spectrum Distribution
f/f_s

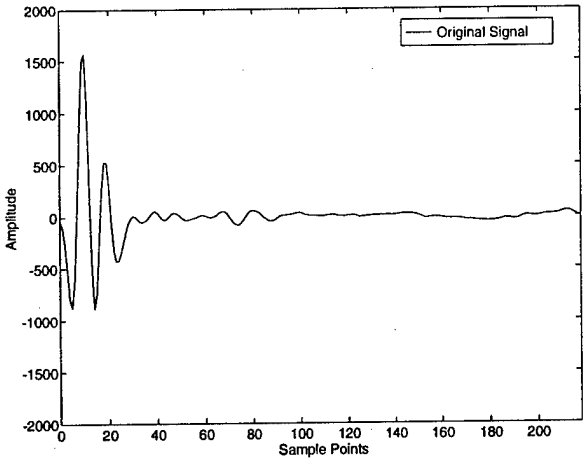
(b)


Output of the Matched Filter
τ/T

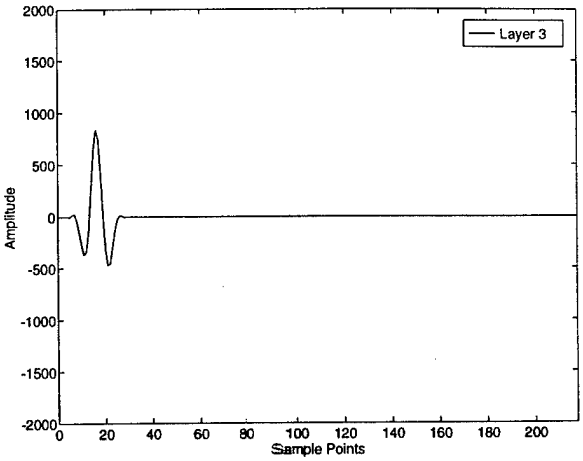
(c)


Output of the Matched Filter
τ/T

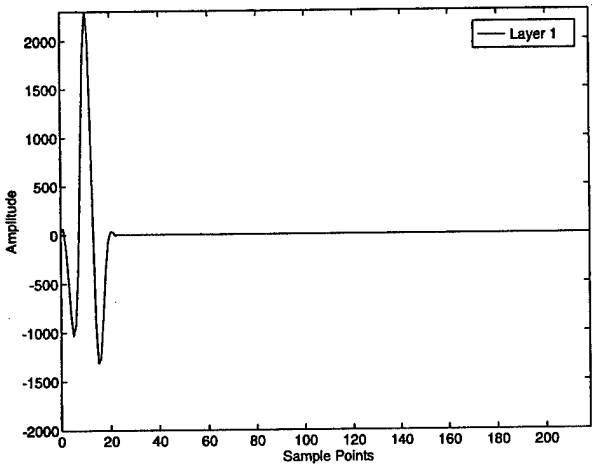
(d)


Original Signal
Amplitude
Sample Points

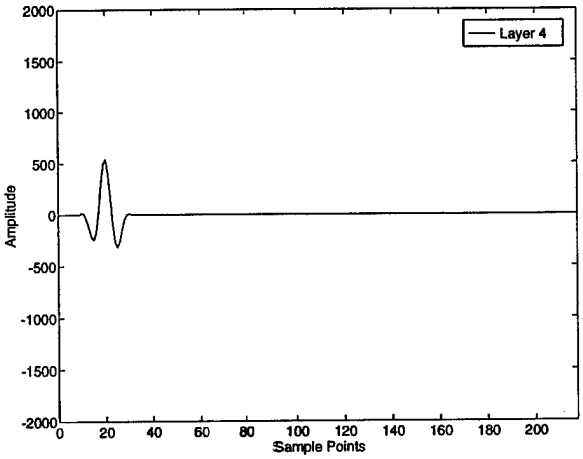
(e)


Layer 3
Amplitude
Sample Points

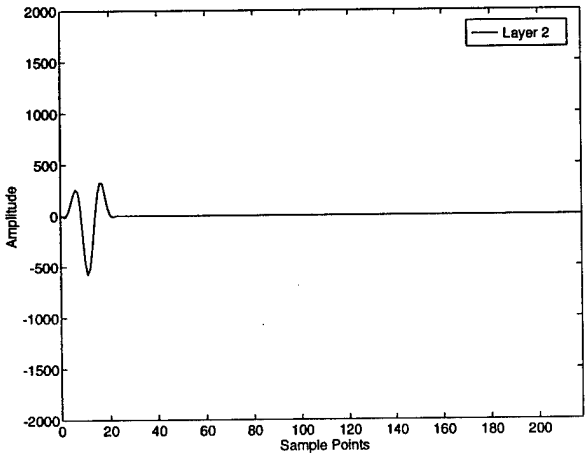
(h)


Layer 1
Amplitude
Sample Points

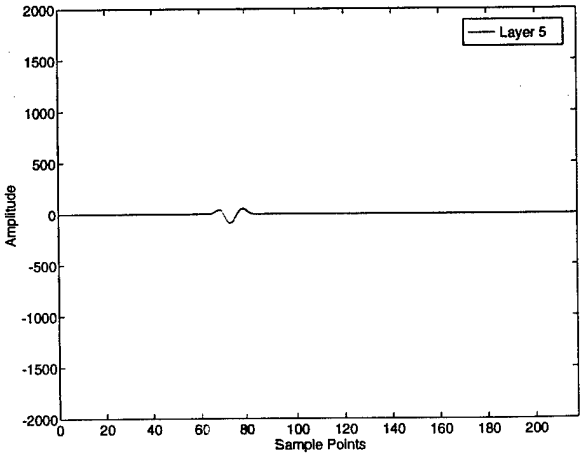
(f)


Layer 4
Amplitude
Sample Points

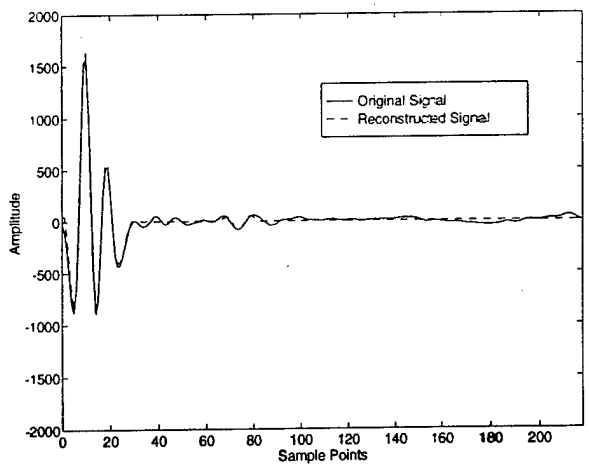
(i)


Layer 2
Amplitude
Sample Points

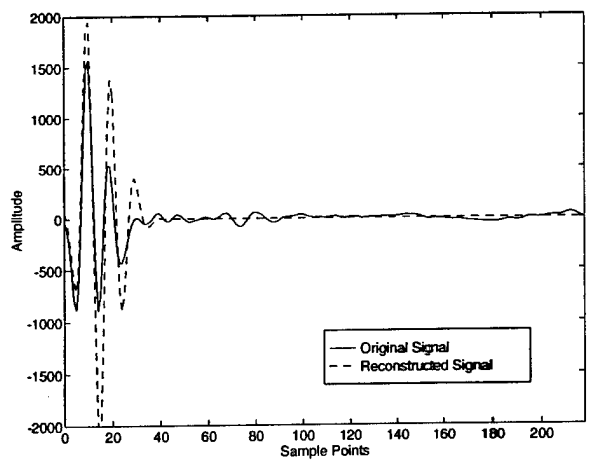
(g)


Layer 5
Amplitude
Sample Points

(j)



(k)



(l)

Figure 11.5: Application of WRELAX to the experimental data acquired by an ultra-wideband ground penetrating radar to analyze roadway subsurface structure . The data is assumed to be a superimposition of five reflection signals coming from five layers. (a) Waveform of the known signal $s(nT_s)$. (b) Discrete Fourier spectrum (magnitude) of $s(nT_s)$. (c) Matched filter output of the known signal $s(nT_s)$. (d) Matched filter output of the observed signal $y(nT_s)$. (e) Waveform of the observed signal $y(nT_s)$. (f)-(j) Reconstructed signals reflected by Layers 1 through 5, respectively, by using WRELAX. (k) Comparison of observed signal (solid line) with reconstructed signal using WRELAX (dashed line). (l) Comparison of the observed signal (solid line) with reconstructed signal using the conventional matched filter method (dashed line).

12. Time Delay Estimation via Optimizing Highly Oscillatory Cost Functions

12.1 Introduction

Time delay estimation is a well known problem in the field of underwater acoustic signal processing. For the purpose of passive localization, two or more spatially separated sensors are often used to measure the time difference of arrivals of a radiating source whose transmitted signal waveform is usually unknown [1, 2]. Another problem is to estimate time delays and amplitudes from the superposition of multiple signals with known waveforms plus noise received at a single sensor (or a beamformed array of sensors). This problem occurs in many applications, including multipath separation [3, 4, 5, 6, 7] and target feature extraction and classification [8] and is considered herein.

The most well-known time delay estimation technique is the matched filter approach. In the case of white Gaussian noise, it performs like the Maximum Likelihood (ML) method when the overlapping signals are well separated in the arrival times and is identical to the ML approach when there is only one signal present. The matched filter approach can be easily implemented using dedicated fast Fourier transform (FFT) chips or correlator chips. However, its resolution is limited to the reciprocal of the signal bandwidth. High resolution sinusoidal frequency estimation techniques such as MUSIC [9], Linear Prediction [10], and Maximum Likelihood (ML) approach are applied to the time delay estimation problems in [11, 12, 13], respectively. However, due to the spectral-division operation involved, these approaches should be applied to signals with flat (rectangular) or almost flat band-limited spectra. Further, they assume that the signals have complex-valued amplitudes. Hence they are usually not optimal when the signals have real-valued amplitudes. A computationally efficient approach based on the Expectation Maximization (EM) algorithm [14] is proposed in [15] that decouples the complicated multidimensional optimization problem into a sequence of multiple separate one-dimensional optimization problems. However, the EM method is very sensitive to initial conditions and no systematic initialization method is given in [15].

For active sonar systems, the most commonly used signal waveforms are bandpass signals. For some propagation environments [3, 4, 16], the received signal can be modeled as the sum of amplitude scaled and time shifted replicas of the known transmitted bandpass signal. Since the transmitted signal is bandpass, the correlation function between the received signal and the known transmitted signal will oscillate near the carrier frequency of the transmitted signal. For this case, many existing time delay estimation algorithms perform poorly due to converging to local optimum points [3, 5, 6, 7]. One way that can be used to avoid this oscillation problem is to first model the received signal as the sum of time shifted replicas of a known transmitted signal with different complex-valued amplitudes to obtain good initial estimates of the unknown parameters [5, 6]. This idea is based on the observation that the nonlinear least-squares (NLS) cost function for signals with complex-valued amplitudes is much smoother than the one for signals with real-valued amplitudes. Hence good initial estimates can be obtained by assuming the signal amplitudes to be complex-valued and minimizing a much smoother cost function. However, due to the parsimony principle [17], the initial estimates are not as accurate as those obtained by using the original real-valued data model. Hence the initial estimates can be refined next by optimizing the original highly oscillatory true cost function corresponding to real-valued amplitudes. The algorithms we propose in this chapter are inspired by this idea first proposed in [5, 6].

In our previous paper [18], a weighted Fourier transform based relaxation method (referred to as WRELAX) is presented for finding the global minimum of the frequency domain NLS cost function. The most striking feature of the WRELAX algorithm is that it decouples the multidimensional optimization problem into a series of one-dimensional optimization problems in a conceptually and computationally simple way. Compared with other existing algorithms, WRELAX is more systematic and efficient and has less limitations on the signal shapes. WRELAX can be used for signals with either real- or complex-valued amplitudes by minimizing slightly different cost functions. Unfortunately, just like many other algorithms, WRELAX is likely to converge to the local minimum instead of the global one when the NLS cost function is highly oscillatory.

In this chapter, enlightened by the idea used in [5, 6], we propose two approaches based on

WRELAX to deal with the problem of optimizing highly oscillatory cost functions. First we assume that the signal amplitudes are complex-valued and use WRELAX to obtain the initial estimates of the delays and the amplitudes of the superimposed signals by minimizing a much smoother NLS cost function. Then the initial estimates are refined with two approaches. One approach (referred to as Hybrid-WRELAX) uses the last step of the WRELAX algorithm to minimize the true NLS cost function corresponding to the real-valued signal amplitudes. The other approach (referred to as EXIP-WRELAX) uses the extended invariance principle (EXIP) [17, 19]. For Hybrid-WRELAX, the refinement step is iterative, while it is not for EXIP-WRELAX. Both of the algorithms are shown to approach the Cramér-Rao bound (CRB) as the signal-to-noise ratio (SNR) increases.

The remainder of this chapter is organized as follows. In Section 12.2, we describe the data model and formulate the problem of interest. The Hybrid-WRELAX and EXIP-WRELAX algorithms are presented in Sections 12.3 and 12.4, respectively. Numerical examples are provided in Section 12.5 to illustrate the performances of the new algorithms. Section 12.6 concludes the chapter. The CRB analysis is included in Appendix A.

12.2 Problem Formulation

The data model used in this chapter has the form [3, 4, 5, 6, 7, 16]

$$y(t) = \sum_{l=1}^{L} \alpha_l s(t - \tau_l) + e(t) \quad 0 \le t \le T, \tag{12.1}$$

where $s(t)$, $0 \le t \le T_0$, represents the known real-valued transmitted signal (usually bandpass), $y(t)$ denotes the real-valued received signal, which is composed of L replicas of $s(t)$ with different real-valued amplitudes $\{\alpha_l\}_{l=1}^{L}$ and real-valued delays $\{\tau_l\}_{l=1}^{L}$, and $e(t)$ is the real-valued receiver noise, which is modeled as a zero-mean Gaussian random process.

The above data model applies to some multipath propagation environments [3, 4, 16]. In practice, many problems can be described by this model. As pointed out in [4], this model applies to any problems where a known probe signal $s(t)$ excites a linear time-invariant filter whose output is observed in the presence of noise and where the impulse response of the

filter is a finite sum of the scaled and time-shifted Dirac delta functions.

The sampled received signal can be written as

$$y(nT_s) = \sum_{l=1}^{L} \alpha_l s(nT_s - \tau_l) + e(nT_s), \quad n = 0, 1, \cdots, N-1, \tag{12.2}$$

where T_s is the sampling period and is equal to the reciprocal of the sampling frequency f_s.

Our problem of interest herein is to estimate $\{\alpha_l, \tau_l\}_{l=1}^{L}$ from $\{y(nT_s)\}_{n=0}^{N-1}$ with known $s(t)$, $0 \leq t \leq T_0$, or $\{s(nT_s)\}_{n=0}^{N-1}$ when the signals are closely spaced. More specifically, we are interested in the high resolution time delay estimation problem in which the probe signal $s(t)$ has highly oscillatory correlation function (such as bandpass signals).

Although we could solve the estimation problem in the time-domain [3, 7, 15, 20], we prefer to do it in the frequency domain. This is because for the time domain processing methods, we could be restricted to using the discrete values of $\{\tau_l\}_{l=1}^{L}$ if we only know the sampled version of $s(t)$. For this case, if a more accurate delay estimate is required, then one has to resort to interpolation [3]. This inconvenience can be avoided by transforming the problem to the frequency domain, where $\{\tau_l\}_{l=1}^{L}$ can take on a continuum of values. Another advantage comes from the simple implementation structure. As will be seen in the next two sections, our frequency domain algorithms are based on a sequence of Fourier transforms, which can be easily implemented by using the currently available dedicated high performance FFT chips, such as TMC2310 [21] and A41102 [22].

Let $Y(k)$, $S(k)$, and $E(k)$, $k = -N/2, -N/2+1, ..., N/2-1$, denote the discrete Fourier transforms (DFT's) of $y(nT_s)$, $s(nT_s)$, and $e(nT_s)$, respectively. Provided that aliasing is negligible, then $Y(k)$ can be written as:

$$Y(k) = S(k) \sum_{l=1}^{L} \alpha_l e^{j\omega_l k} + E(k), \tag{12.3}$$

where

$$\omega_l = -\frac{2\pi\tau_l}{NT_s}. \tag{12.4}$$

Note that the time delay estimation problem is similar to the sinusoidal parameter estimation problem except that the exponential signals are weighted by the known signal

spectrum. If we divided both sides of (12.3) by $S(k)$, the problem would become identical to the sinusoidal parameter estimation problem. Yet we should not do so for the following reasons: first, $S(k)$ could be zero for some k; second, the noise $E(k)/S(k)$ will no longer be a white noise even when $E(k)$ is white; third, when $E(k)$ is a white noise, the larger the $S(k)$ at sample k, the higher the SNR of the corresponding $Y(k)$ and hence dividing $Y(k)$ by $S(k)$ will de-emphasize those $Y(k)'s$ that have high SNRs. Because of this, many well-known sinusoidal parameter estimation algorithms, such as MUSIC [9], ESPRIT [23], and PRONY [24], are not directly applicable to our problem of interest. Further, since these algorithms are designed for complex-valued amplitudes, they do not take advantage of the real-valued amplitudes and hence cannot provide the best possible performance.

12.3 The Hybrid-WRELAX Algorithm

We consider below estimating the unknown parameters by minimizing the following NLS criterion:

$$C_1(\{\alpha_l, \omega_l\}_{l=1}^L) = \sum_{k=-N/2}^{N/2-1} \left| Y(k) - S(k) \sum_{l=1}^{L} \alpha_l e^{j\omega_l k} \right|^2 . \tag{12.5}$$

Since both the transmitted signal $s(t)$ and the received signal $y(t)$ are real-valued, their Fourier transforms are conjugate symmetric, i.e., $Y(-k) = Y^*(k)$ and $S(-k) = S^*(k), \quad k = 1, 2, \cdots, N/2-1$, where $(\cdot)^*$ denotes the complex conjugate, and $Y(-N/2)$, $Y(0)$, $S(-N/2)$, and $S(0)$ are real-valued. It can be readily shown that the above cost function is equivalent to

$$C_2(\{\alpha_l, \omega_l\}_{l=1}^L) = \sum_{k=-N/2}^{0} W^2(k) \left| Y(k) - S(k) \sum_{l=1}^{L} \alpha_l e^{j\omega_l k} \right|^2 , \tag{12.6}$$

where $\{W(k) = 1\}_{k=-N/2+1}^{-1}$ and $W(-N/2) = W(0) = 1/\sqrt{2}$. We assume that $e(nT_s)$ is a real-valued zero-mean white Gaussian random process with variance σ^2. Yet $E(k)$ will not be a circularly symmetric complex-valued zero-mean white Gaussian random process since $E(-k) = E^*(k), \quad k = 1, 2, \cdots, N/2-1$. (The circularly symmetric assumption on the noise

is widely used in the literature [25].) Nevertheless, it is shown in Appendix A that for this white noise case, the above NLS approach is the same as the ML method.

The cost function $C_2(\{\alpha_l, \omega_l\}_{l=1}^L)$ in (12.6) with $\{\alpha_l\}_{l=1}^L$ being real-valued is referred to as the true cost function. Minimizing $C_2(\{\alpha_l, \omega_l\}_{l=1}^L)$ with respect to the unknown parameters is a highly nonlinear optimization problem. For narrowband transmitted signals, the cost function is highly oscillatory and have numerous closely spaced local minima, which makes it very difficult to find the global minimum. By assuming the real-valued amplitudes $\{\alpha_l\}_{l=1}^L$ to be complex-valued, a much smoother cost function can be obtained. This is equivalent to formulate the original time delay estimation problem in its complex analytic signal form. Since the analytic signal of the transmitted signal is lowpass, its autocorrelation function is no longer oscillatory. This is the conventional complex demodulation process and is widely used in practice. Although it is much easier to find the global minimum of the cost function corresponding to complex-valued amplitudes, the so-obtained estimates can be much less accurate than those obtained by minimizing the true cost function. The two cost functions share the same global minimum only when there is no noise. However, as suggested in [5, 6], we can minimize the cost function associated with complex-valued amplitudes to obtain the initial conditions needed to minimize the true cost function. Below, we present a relaxation based global minimizer of the NLS criterion based on this idea. The algorithm is referred to as the Hybrid-WRELAX algorithm. It simply requires a sequence of weighted Fourier transforms.

Before we present our approach, let us consider the following preparations. Let

$$\begin{aligned} \mathbf{W} &= \operatorname{diag}\left\{ W(-N/2),\ W(-N/2+1),\ \cdots,\ W(-1),\ W(0) \right\} \\ &= \operatorname{diag}\left\{ \tfrac{1}{\sqrt{2}},\ 1,\ \cdots,\ 1,\ \tfrac{1}{\sqrt{2}} \right\}, \end{aligned} \tag{12.7}$$

$$\mathbf{Y} = \mathbf{W} \begin{bmatrix} Y(-N/2) & Y(-N/2+1) & \cdots & Y(0) \end{bmatrix}^T, \tag{12.8}$$

$$\mathbf{S} = \mathbf{W}\ \operatorname{diag}\left\{ S(-N/2),\ S(-N/2+1),\ \cdots,\ S(0) \right\}, \tag{12.9}$$

and

$$\mathbf{a}(\omega_l) = \begin{bmatrix} e^{j\omega_l(-N/2)} & e^{j\omega_l(-N/2+1)} & \cdots & 1 \end{bmatrix}^T \tag{12.10}$$

where $(\cdot)^T$ denotes the transpose. Denote

$$\mathbf{Y}_l = \mathbf{Y} - \sum_{i=1, i\neq l}^{L} \hat{\alpha}_i[\mathbf{S}\mathbf{a}(\hat{\omega}_i)] \tag{12.11}$$

where $\{\hat{\alpha}_i, \hat{\omega}_i\}_{i=1,i\neq l}$ are assumed to be given. Let

$$\mathbf{b}(\omega_l) = \mathbf{S}\mathbf{a}(\omega_l), \quad l = 1, 2, \cdots, L. \tag{12.12}$$

Then (12.6) becomes

$$C_3(\alpha_l, \omega_l) = \| \mathbf{Y}_l - \alpha_l \mathbf{b}(\omega_l) \|^2, \tag{12.13}$$

where $\| \cdot \|$ denotes the Euclidean norm. Minimizing $C_3(\alpha_l, \omega_l)$ with respect to the real-valued α_l yields the estimate $\hat{\alpha}_l$ of α_l

$$\begin{aligned} \hat{\alpha}_l &= \frac{\mathrm{Re}\left[\mathbf{b}^H(\omega_l)\mathbf{Y}_l\right]}{\mathbf{b}^H(\omega_l)\mathbf{b}(\omega_l)} \\ &= \frac{\mathrm{Re}\left[\mathbf{a}^H(\omega_l)(\mathbf{S}^*\mathbf{Y}_l)\right]}{\| \mathbf{S} \|_F^2}. \end{aligned} \tag{12.14}$$

where $(\cdot)^H$ denotes the conjugate transpose, $\mathrm{Re}(\mathbf{Z})$ represents the real part of $\mathbf{Z}$, and $\| \cdot \|_F$ denotes the *Frobenius* norm [26]. (More specifically, $\| \mathbf{S} \|_F = \sqrt{\sum_{n=-N/2}^{0} |W(n)S(n)|^2}$.) Then the estimate $\hat{\omega}_l$ of ω_l is obtained as follows:

$$\begin{aligned} \hat{\omega}_l &= \arg\min_{\omega_l} \left\| \mathbf{Y}_l - \frac{\mathrm{Re}\left[\mathbf{b}^H(\omega_l)\mathbf{Y}_l\right]}{\mathbf{b}^H(\omega_l)\mathbf{b}(\omega_l)} \mathbf{b}(\omega_l) \right\|^2 \\ &= \arg\max_{\omega_l} \mathrm{Re}^2\left[\mathbf{a}^H(\omega_l)(\mathbf{S}^*\mathbf{Y}_l)\right]. \end{aligned} \tag{12.15}$$

where we have used the fact that $\mathbf{b}^H(\omega_l)\mathbf{b}(\omega_l) = \sum_{n=-N/2}^{0} |W(n)S(n)|^2$ and hence is independent of ω_l. Hence $\hat{\omega}_l$ is obtained as the location of the dominant peak of $\mathrm{Re}^2\left[\mathbf{a}^H(\omega_l)(\mathbf{S}^*\mathbf{Y}_l)\right]$, which can be efficiently computed by using FFT with the weighted data vector $\mathbf{S}^*\mathbf{Y}_l$ padded

with zeros. An alternative scheme to zero-padding FFT is to find an approximate peak location first by using FFT without much zero-padding and then perform a fine search nearby the approximate peak location by, for example, the *fmin* function in MATLAB, which uses the Golden section search algorithm. With the estimate of ω_l at hand, $\hat{\alpha}_l$ is easily computed from the corresponding complex height by using $\hat{\omega}_l$ to replace ω_l in (12.14).

Similarly, minimizing $C_3(\alpha_l, \omega_l)$ with respect to ω_l and the complex-valued α_l, respectively, yields the estimates $\hat{\omega}_l$ of ω_l and $\hat{\alpha}_l$ of α_l,

$$\hat{\omega}_l = \arg\max_{\omega_l} \left| \mathbf{a}^H(\omega_l)(\mathbf{S}^*\mathbf{Y}_l) \right|^2, \tag{12.16}$$

and

$$\hat{\alpha}_l = \left. \frac{\mathbf{a}^H(\omega_l)(\mathbf{S}^*\mathbf{Y}_l)}{\| \mathbf{S} \|_F^2} \right|_{\omega_l = \hat{\omega}_l}, \tag{12.17}$$

where $\hat{\omega}_l$ can also be found via FFT with the weighted data vector.

With the above simple preparations, we now present the Hybrid-WRELAX algorithm.

Step 1: Obtain the initial conditions for Step 2 by assuming that $\{\alpha_l\}_{l=1}^L$ are complex-valued and using the WRELAX algorithm as follows:

Substep (1): Assume $L = 1$. Obtain $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1}$ from $\mathbf{Y}$ by using (12.16) and (12.17).

Substep (2): Assume $L = 2$. Compute $\mathbf{Y}_2$ with (12.11) by using $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1}$ obtained in Substep (1). Obtain $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=2}$ from $\mathbf{Y}_2$. Next, compute $\mathbf{Y}_1$ by using $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=2}$ and then redetermine $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1}$ from $\mathbf{Y}_1$.

Iterate the update of $\{\hat{\omega}_2, \hat{\alpha}_2\}$ and $\{\hat{\omega}_1, \hat{\alpha}_1\}$ until "practical convergence" is achieved (to be discussed later on).

Substep (3): Assume $L = 3$. Compute $\mathbf{Y}_3$ by using $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1}^2$ obtained in Substep (2). Obtain $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=3}$ from $\mathbf{Y}_3$. Next, compute $\mathbf{Y}_1$ by using $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=2}^3$ and redetermine $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1}$ from $\mathbf{Y}_1$. Then compute $\mathbf{Y}_2$ by using $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1,3}$ and redetermine $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=2}$ from $\mathbf{Y}_2$.

Iterate the update of $\{\hat{\omega}_3, \hat{\alpha}_3\}$,$\{\hat{\omega}_1, \hat{\alpha}_1\}$, and $\{\hat{\omega}_2, \hat{\alpha}_2\}$ until "practical convergence".

Remaining Substeps: Continue similarly until L is equal to the desired or estimated number of signals.

Step 2: Refine the estimates obtained in Step 1 with the last step of the WRELAX algorithm (i.e., the last substep of Step 1 above) by using Equations (12.14) and (12.15) derived for the real-valued $\{\alpha_l\}_{l=1}^L$ and using $\{\hat{\omega}_l\}_{l=1}^L$ and the real parts of $\{\hat{\alpha}_l\}_{l=1}^L$ obtained in Step 1 as initial conditions. Iteratively update $\{\hat{\omega}_l, \hat{\alpha}_l\}$, $l = 1, 2, \cdots, L$, until "practical convergence".

The "practical convergence" in the iterations of the above method may be determined by checking the relative change of the cost function $C_2(\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1}^L)$ in (12.6) between two consecutive iterations. The algorithm is bound to converge to at least a local minimum point [27]. The convergence speed depends on the time delay spacing of the signals. If the spacing between any two signals is not too much smaller than the reciprocal of the signal bandwidth, the algorithm converges in a few steps. As the spacing of the signals becomes closer, the convergence speed becomes slower.

Note that WRELAX can be used directly for signals with real-valued amplitudes. For this case, the approach would consist of the substeps of Step 1 above except that (12.14) and (12.15) will be used instead of (12.16) and (12.17), respectively. We will use a numerical example in Section 5 to show the problem encountered by the direct use of WRELAX when the cost function is highly oscillatory.

Once we have obtained the estimates $\{\hat{\omega}_l\}_{l=1}^L$, the estimates $\{\hat{\tau}_l\}_{l=1}^L$ of $\{\tau_l\}_{l=1}^L$ can be readily calculated by using (12.4).

12.4 The EXIP-WRELAX Algorithm

The Invariance Principle (IP) of ML estimators is well known in the estimation theory [28]. The invariance principle gives a simple answer to the relationship between the minimizers of a given cost function parameterized in two different ways in some special cases. By appropriately reparameterizing the original cost function and enlarging the supporting domain of the parameter space, less accurate estimates can be obtained from this simple data

model. These estimates may be refined to asymptotically achieve the performance available using the original data model. This is the basic idea behind the Extended Invariance Principle (EXIP) proposed in [17, 19] for the purpose of achieving some computational advantages. In this section, we present an EXIP based algorithm, referred to as the EXIP-WRELAX algorithm, that avoids dealing with the highly oscillatory true cost function entirely.

By using (12.8) and (12.12), the cost function (12.6) with $\{\alpha_l\}_{l=1}^{L}$ being real-valued can be written in the following vector form

$$C_{\boldsymbol{\eta}}(\boldsymbol{\eta}) = \left\| \mathbf{Y} - \sum_{l=1}^{L} \alpha_l \mathbf{b}(\omega_l) \right\|_2^2, \tag{12.18}$$

where

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\alpha}^T & \boldsymbol{\omega}^T \end{bmatrix}^T, \tag{12.19}$$

with

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_L \end{bmatrix}^T, \tag{12.20}$$

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_1 & \omega_2 & \cdots & \omega_L \end{bmatrix}^T. \tag{12.21}$$

By replacing the real-valued amplitudes $\{\alpha_l\}_{l=1}^{L}$ with the complex-valued amplitudes $\{\tilde{\alpha}_l\}_{l=1}^{L}$ (notations introduced for the sake of clarity) in (12.18), we obtain the following cost function:

$$C_{\tilde{\boldsymbol{\eta}}}(\tilde{\boldsymbol{\eta}}) = \left\| \mathbf{Y} - \sum_{l=1}^{L} \tilde{\alpha}_l \mathbf{b}(\omega_l) \right\|_2^2, \tag{12.22}$$

where

$$\tilde{\boldsymbol{\eta}} = \begin{bmatrix} \mathrm{Re}^T(\tilde{\boldsymbol{\alpha}}) & \mathrm{Im}^T(\tilde{\boldsymbol{\alpha}}) & \boldsymbol{\omega}^T \end{bmatrix}^T, \tag{12.23}$$

with $\mathrm{Im}(\mathbf{Z})$ denotes the imaginary part of $\mathbf{Z}$, and

$$\tilde{\boldsymbol{\alpha}} = \begin{bmatrix} \tilde{\alpha}_1 & \tilde{\alpha}_2 & \cdots & \tilde{\alpha}_L \end{bmatrix}^T. \tag{12.24}$$

Denote

$$\hat{\boldsymbol{\eta}} = \arg\min_{\boldsymbol{\eta}} \; C_{\boldsymbol{\eta}}(\boldsymbol{\eta}), \tag{12.25}$$

and

$$\hat{\tilde{\boldsymbol{\eta}}} = \arg\min_{\tilde{\boldsymbol{\eta}}} \; C_{\tilde{\boldsymbol{\eta}}}(\tilde{\boldsymbol{\eta}}). \tag{12.26}$$

Let

$$f(\boldsymbol{\eta}) = \mathbf{F}\boldsymbol{\eta}, \tag{12.27}$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{12.28}$$

with $\mathbf{I}$ and $\mathbf{0}$ denote the $L \times L$ identity matrix and the $L \times L$ matrix with zero elements, respectively. Using the EXIP principle [19, 17], we can obtain a new estimate $\hat{\hat{\boldsymbol{\eta}}}$ from $\hat{\tilde{\boldsymbol{\eta}}}$ by solving the following weighted least squares problem

$$\hat{\hat{\boldsymbol{\eta}}} = \arg\min_{\boldsymbol{\eta}} \left[\hat{\tilde{\boldsymbol{\eta}}} - f(\boldsymbol{\eta})\right]^T \mathbf{W}_{\text{EXIP}} \left[\hat{\tilde{\boldsymbol{\eta}}} - f(\boldsymbol{\eta})\right], \tag{12.29}$$

where

$$\mathbf{W}_{\text{EXIP}} = E\left\{\frac{\partial^2 \left[C_{\tilde{\boldsymbol{\eta}}}(\tilde{\boldsymbol{\eta}})\right]}{\partial\tilde{\boldsymbol{\eta}}\partial\tilde{\boldsymbol{\eta}}^T}\right\}\Bigg|_{\tilde{\boldsymbol{\eta}}=\hat{\tilde{\boldsymbol{\eta}}}}. \tag{12.30}$$

It has been shown in [19, 17] that $\hat{\hat{\boldsymbol{\eta}}}$ is asymptotically (for large N or high SNR) statistically equivalent to $\hat{\boldsymbol{\eta}}$. The weighting matrix $\mathbf{W}_{\text{EXIP}}$ is simply the *Fisher Information Matrix* (possibly scaled by a constant) for the complex-valued $\{\tilde{\alpha}_l\}_{l=1}^{L}$ with $\tilde{\boldsymbol{\eta}}$ replaced by its estimate $\hat{\tilde{\boldsymbol{\eta}}}$ (see Appendix B for more details). It can be easily shown that

$$\hat{\hat{\boldsymbol{\eta}}} = \left(\mathbf{F}^T\mathbf{W}_{\text{EXIP}}\mathbf{F}\right)^{-1}\left(\mathbf{F}^T\mathbf{W}_{\text{EXIP}}\right)\hat{\tilde{\boldsymbol{\eta}}}. \tag{12.31}$$

The EXIP-WRELAX algorithm is composed of two steps. The first step is the same as Step 1 of the Hybrid-WRELAX algorithm and the second step is to refine the initial conditions obtained in Step 1 by using (12.31). Compared to the Hybrid-WRELAX algorithm, the second step of the EXIP-WRELAX algorithm is non-iterative and avoids dealing with the highly oscillatory true NLS cost function entirely. Our numerical examples show that at low SNR, the former tends to outperform the latter.

12.5 Simulation Results and Discussions

In this section, we present several numerical examples illustrating the performances of the proposed algorithms. The performances of the new algorithms are compared with the CRB, which gives the minimum attainable variances for any unbiased estimators.

In the numerical examples below, we use a windowed chirp signal,

$$s(t) = w(t)\cos\left[2\pi f_0 t + \beta\left(t - \frac{T_0}{2}\right)^2\right], \quad 0 \le t \le T_0, \tag{12.32}$$

where f_0 denotes the carrier frequency, β represents the chirp rate, and

$$w(t) = \begin{cases} 0.5 - 0.5\cos(\pi t/T_w), & 0 < t < T_w, \\ 1, & T_w \le t \le T_0 - T_w, \\ 0.5 - 0.5\cos[\pi(t - T_0)/T_w], & T_0 - T_w < t \le T_0, \end{cases} \tag{12.33}$$

with $T_w = T_0/10$.

In the following simulations, we use $N = 256$, $\beta = \pi \times 10^5$, the signal bandwidth $B_s = \beta T_0/\pi$, and the sampling frequency $f_s = 8B_s$. T_0 is chosen in such a way that $T_0 = (N/2-1)T_s$. In this case, it can be shown that $T_0 = \sqrt{\frac{(N/2-1)\pi}{8\beta}} = 12.6$ ms, $T = 25.3$ ms, $T_s = 99.209$ μs, $B_s = 1.26$ KHz, $f_s = 10.08$ KHz, and the resolution limit of the conventional matched filter method is generally considered to be around $\tau_e = 1/B_s = 0.79368$ ms.

In all of the examples below, we have used $\epsilon = 0.001$ (the relative change in the cost function) to test the practical convergence of the WRELAX algorithm. The one-dimensional search is performed in two steps, with a coarse search using FFT followed by a fine search using the $fmin$ function of MATLAB. Since the cost function for real-valued amplitudes is more oscillatory than the one for complex-valued amplitudes, we use more zero paddings with FFT for the former case. For the former case, the data length after zero padding is $4N$, while for the latter case, it is N. The sampled noise $\{e(nT_s)\}$ is assumed to be a real-valued zero-mean white Gaussian random process with variance σ^2. The SNR for each signal is defined as $10\log_{10}(\alpha_i^2/2\sigma^2)$. The MSE is obtained through 100 independent Monte-Carlo trials.

To see the oscillatory nature of the cost function we are dealing with, consider, for

simplicity, the case where there is only one direct path with delay $\tau_1 = 0$ and no noise is present. The normalized cost functions used to obtained $\hat{\omega}_1$ for real-valued α_1 (solid line, corresponding to (12.15) and complex-valued α_1 (dashed line, corresponding to (12.16) are compared with each other in Figure 1, where the horizontal axis denotes the normalized time delay τ/T and the carrier frequency of the transmitted signal is $f_0 = 2B_s$. From Figure 1, it can be seen that the cost function obtained by assuming the real-valued α_1 to be complex-valued is approximately the envelope of the true cost function [3, 5, 6]. The former is very smooth and does not change with f_0, while the latter is highly oscillatory and oscillates more abruptly as f_0 increases. Maximizing the latter can yield much more accurate parameter estimates than maximizing the former due to the sharper dominant peak of the latter (see the following example).

The carrier frequency has a significant impact on the achievable estimation accuracy, as can be seen from Figure 2. Figure 2 compares the CRBs for the first signal when there are $L = 2$ signals with $\alpha_1 = 1$, $\alpha_2 = 1$, $\tau_1 = T_0/8$, and $\tau_2 = T_0/8 + 0.5\tau_e$, and different carrier frequencies $f_0 = rB_s$. (The CRB curves for the other signal are similar.) It appears that the CRBs for both the delays and amplitudes are usually sensitive to f_0, especially for the delay estimates, and the higher the carrier frequency, the lower their CRBs. This result can be intuitively explained with Figure 1. As the carrier frequency becomes larger, the mainlobe of the true cost function becomes narrower, and thus a better accuracy can be obtained. However, it generally requires more sophisticated and computationally more expensive implementation algorithms to achieve the higher accuracy potential provided by a larger carrier frequency.

To illustrate the problem of the direct application of WRELAX to the case of real-valued $\{\alpha_l\}_{l=1}^{L}$, consider the example where the signals are the same as used in Figure 2 except that f_0 is fixed to $2B_s$. Waveforms of the transmitted signal and the noise-free received signal are compared in Figures 3(a) and (b), respectively. The output of the matched filter is shown in Figure 3(c). From Figure 3(c), it is obvious that the matched filter method cannot resolve the two signals. We compare the convergence properties of WRELAX for assuming $\{\alpha_l\}_{l=1}^{L}$ being real-valued ("o") and complex-valued ("*") in the absence of noise in Figure

4, where the horizontal axis denotes the iteration number and the vertical axis denotes the NLS cost function in (12.6). It can be seen that even for the noise free case, WRELAX gets trapped to some local minimum very quickly for assuming $\{\alpha_l\}_{l=1}^L$ being real-valued, while converges to the global minimum for assuming $\{\alpha_l\}_{l=1}^L$ being complex-valued even though $\{\alpha_l\}_{l=1}^L$ are in fact real-valued. This example demonstrates the importance of Step 1 of the Hybrid-WRELAX algorithm.

Finally, we add noise to the above example and compare the performances of the two new algorithms. The MSEs of the WRELAX ("+") for assuming $\{\alpha_l\}_{l=1}^L$ being complex-valued, Hybrid-WRELAX ("o"), and EXIP-WRELAX ("×") are compared with the CRBs obtained by assuming $\{\alpha_l\}_{l=1}^L$ being complex-valued (dashed line) and real-valued (solid line) in Figure 5. Note that both Hybrid-WRELAX and EXIP-WRELAX achieve the corresponding CRB. Note also that the threshold effect is obvious in Figure 5, where the MSEs deviate away from the CRBs at low SNR. Although the WRELAX for assuming $\{\alpha_l\}_{l=1}^L$ being complex-valued also attains its corresponding CRB (dashed line) at high SNR, this wrong CRB can be larger than the true CRB by approximately 30 dB. (Note that the former CRB is expected to be worse than the latter CRB due to the parsimony principle [17].) In this example, Hybrid-WRELAX outperforms EXIP-WRELAX at low SNR.

12.6 Conclusions

In this chapter, we have proposed two relaxation based algorithms (Hybrid-WRELAX and EXIP-WRELAX) to deal with the difficult problem of resolving closely spaced multipaths by minimizing the highly oscillatory nonlinear least squares (NLS) cost functions. The basic idea of the two algorithms is to first find reliable initial estimates of the unknown parameters by minimizing a much less oscillatory cost function and then refine the initial estimates by either minimizing the true cost function or applying the extended invariance principle. Both approaches rely on the WRELAX algorithm, which is a relaxation-based global minimizer of the NLS criterion requiring only a sequence of weighted Fourier transforms. Both of the two proposed algorithms are shown to approach the Cramér-Rao bound as the signal-to-noise ratio increases. At low SNR, Hybrid-WRELAX performs better than

EXIP-WRELAX and hence is preferred. At high SNR, the latter is preferred due to its simplicity and similar performance as the former.

Appendix A: Derivation of CRBs

We sketch below the derivation of the CRBs for the real-valued parameters of the data model in (12.3). Due to the conjugate symmetry property of DFT, $\{Y(k)\}_{k=-N/2}^{N/2-1}$ can be expressed in terms of $\{Y(k)\}_{k=-N/2}^{0}$ with $\{Y(k)\}_{k=-N/2+1}^{-1}$ being complex-valued and $\{Y(-N/2),\ Y(0)\}$ being real-valued. Let

$$\mathbf{Y}_c = \begin{bmatrix} Y(-N/2+1) & Y(-N/2+2) & \cdots & Y(-1) \end{bmatrix}^T, \tag{12.34}$$

$$\mathbf{Y}_r = \begin{bmatrix} Y(-N/2) & Y(0) \end{bmatrix}^T, \tag{12.35}$$

$$\mathbf{E}_c = \begin{bmatrix} E(-N/2+1) & E(-N/2+2) & \cdots & E(-1) \end{bmatrix}^T, \tag{12.36}$$

$$\mathbf{E}_r = \begin{bmatrix} E(-N/2) & E(0) \end{bmatrix}^T, \tag{12.37}$$

$$\mathbf{S}_c = \mathrm{diag}\left\{ S(-N/2+1),\ S(-N/2+2),\ \cdots,\ S(-1) \right\}, \tag{12.38}$$

and

$$\mathbf{S}_r = \mathrm{diag}\left\{ S(-N/2),\ S(0) \right\}. \tag{12.39}$$

From (12.3) and (12.20), we have

$$\mathbf{Y}_c = \boldsymbol{\Omega}_c \boldsymbol{\alpha} + \mathbf{E}_c, \tag{12.40}$$

$$\mathbf{Y}_r = \boldsymbol{\Omega}_r \boldsymbol{\alpha} + \mathbf{E}_r, \tag{12.41}$$

where $\boldsymbol{\alpha}$ is defined in (12.20),

$$\boldsymbol{\Omega}_c = \mathbf{S}_c \mathbf{A}_c, \tag{12.42}$$

with

$$\mathbf{A}_c = \begin{bmatrix} \mathbf{a}_c(\tau_1) & \mathbf{a}_c(\tau_2) & \cdots & \mathbf{a}_c(\tau_L) \end{bmatrix}, \tag{12.43}$$

$$\mathbf{a}_c(\tau_l) = \left[\exp\left[-j\frac{2\pi\tau_l}{NT_s}(-\frac{N}{2}+1)\right] \ \exp\left[-j\frac{2\pi\tau_l}{NT_s}(-\frac{N}{2}+2)\right] \cdots \ \exp\left[-j\frac{2\pi\tau_l}{NT_s}(-1)\right] \right]^T, \tag{12.44}$$

and

$$\mathbf{\Omega}_r = \mathbf{S}_r\mathbf{A}_r, \tag{12.45}$$

with

$$\mathbf{A}_r = \left[\mathbf{a}_r(\tau_1) \ \ \mathbf{a}_r(\tau_2) \ \ \cdots \ \ \mathbf{a}_r(\tau_L) \right], \tag{12.46}$$

and

$$\mathbf{a}_r(\tau_l) = \left[\exp\left[-j\frac{2\pi\tau_l}{NT_s}(-\frac{N}{2})\right] \ \ 1 \right]^T. \tag{12.47}$$

Assume that the additive noise $\{e(nT_s)\}_{n=0}^{N-1}$ is a real-valued zero-mean white Gaussian random process with variance σ^2. Denote

$$\boldsymbol{\theta} = \left[\boldsymbol{\alpha}^T \ \ \boldsymbol{\tau}^T \ \ \sigma^2 \right]^T, \tag{12.48}$$

where

$$\boldsymbol{\tau} = \left[\tau_1 \ \ \tau_2 \ \ \cdots \ \ \tau_L \right]^T. \tag{12.49}$$

Since DFT is a unitary operator, the likelihood function for $\mathbf{Y}_c$ and $\mathbf{Y}_r$ has the form

$$\begin{aligned} p(\mathbf{Y}_c, \mathbf{Y}_r|\boldsymbol{\theta}) \ &= \ \frac{1}{2\pi^{N/2}\sigma^N}\exp\left\{-\frac{1}{\sigma^2}\left(\mathbf{Y}_c - \mathbf{\Omega}_c\boldsymbol{\alpha}\right)^H\left(\mathbf{Y}_c - \mathbf{\Omega}_c\boldsymbol{\alpha}\right) - \right. \\ &\quad \left. \frac{1}{2\sigma^2}\left(\mathbf{Y}_r - \mathbf{\Omega}_r\boldsymbol{\alpha}\right)^T\left(\mathbf{Y}_r - \mathbf{\Omega}_r\boldsymbol{\alpha}\right)\right\}. \end{aligned} \tag{12.50}$$

Using the $\mathbf{Y}$ and the $\mathbf{S}$ defined in (12.8) and (12.9), respectively, we can rewrite (12.50) in the following compact form

$$p(\mathbf{Y}_c, \mathbf{Y}_r|\boldsymbol{\theta}) = \frac{1}{2\pi^{N/2}\sigma^N}\exp\left\{-\frac{1}{\sigma^2}\left(\mathbf{Y} - \mathbf{SA}\boldsymbol{\alpha}\right)^H\left(\mathbf{Y} - \mathbf{SA}\boldsymbol{\alpha}\right)\right\}. \tag{12.51}$$

where

$$\mathbf{A} = \left[\mathbf{a}(\tau_1) \ \ \mathbf{a}(\tau_2) \ \ \cdots \ \ \mathbf{a}(\tau_L) \right], \tag{12.52}$$

with

$$\mathbf{a}(\tau_l) = \left[\exp\left[-j\frac{2\pi\tau_l}{NT_s}(-\frac{N}{2})\right] \ \ \exp\left[-j\frac{2\pi\tau_l}{NT_s}(-\frac{N}{2}+1)\right] \ \ \cdots \ \ 1 \right]^T. \tag{12.53}$$

Hence the ML estimates of $\boldsymbol{\alpha}$ and $\boldsymbol{\tau}$ are obtained by minimizing $(\mathbf{Y} - \mathbf{SA}\boldsymbol{\alpha})^H (\mathbf{Y} - \mathbf{SA}\boldsymbol{\alpha})$, which is equivalent to the NLS cost function in (12.6).

The CRB matrix $\mathrm{CRB}(\boldsymbol{\theta})$ corresponding to the unknown parameter vector $\boldsymbol{\theta}$ has the form [25]:

$$\mathrm{CRB}^{-1}(\boldsymbol{\theta}) = E\left\{\left[\frac{\partial \ln p(\mathbf{Y}_c,\ \mathbf{Y}_r|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right]\left[\frac{\partial \ln p(\mathbf{Y}_c, \mathbf{Y}_r|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right]^T\right\}. \tag{12.54}$$

It can be shown that the matrix $\mathrm{CRB}(\boldsymbol{\theta})$ is block diagonal with its last row and column being zero except for the last diagonal element. Let the signal parameter vector $\boldsymbol{\eta}$ be denoted as

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\alpha}^T & \boldsymbol{\tau}^T \end{bmatrix}^T. \tag{12.55}$$

Then it is readily shown that the ijth element for $\mathrm{CRB}^{-1}(\boldsymbol{\eta})$ has the form:

$$\left[\mathrm{CRB}^{-1}(\boldsymbol{\eta})\right]_{ij} = \frac{2}{\sigma^2}\mathrm{Re}\left\{\left[(\mathbf{SA}\boldsymbol{\alpha})_i'\right]^H \left[(\mathbf{SA}\boldsymbol{\alpha})_j'\right]\right\}, \tag{12.56}$$

where $(\mathbf{Z})_i'$ denotes $\partial \mathbf{Z}/\partial \eta_i$ with η_i being the ith element of $\boldsymbol{\eta}$.

Appendix B: Derivation of $\mathbf{W}_{\mathbf{EXIP}}$

For complex-valued $\tilde{\boldsymbol{\alpha}}$, where $\tilde{\boldsymbol{\alpha}}$ is defined in (12.24), let

$$\tilde{\boldsymbol{\eta}} = \begin{bmatrix} \mathrm{Re}^T(\tilde{\boldsymbol{\alpha}}) & \mathrm{Im}^T(\tilde{\boldsymbol{\alpha}}) & \boldsymbol{\tau}^T \end{bmatrix}^T. \tag{12.57}$$

Then similar to the derivations in Appendix A, the ijth element of $\mathbf{W}_{\mathrm{EXIP}}$ has the form:

$$\left[\mathbf{W}_{\mathrm{EXIP}}\right]_{ij} = 2\mathrm{Re}\left\{\left[(\mathbf{SA}\tilde{\boldsymbol{\alpha}})_i'\right]^H \left[(\mathbf{SA}\tilde{\boldsymbol{\alpha}})_j'\right]\right\}\Big|_{\tilde{\boldsymbol{\eta}}=\hat{\tilde{\boldsymbol{\eta}}}}, \tag{12.58}$$

where $(\mathbf{Z})_i'$ denotes $\partial \mathbf{Z}/\partial \tilde{\eta}_i$ with $\tilde{\eta}_i$ being the ith element of $\tilde{\boldsymbol{\eta}}$, and $\hat{\tilde{\boldsymbol{\eta}}}$ is the estimate of $\tilde{\boldsymbol{\eta}}$ obtained by using WRELAX when assuming complex-valued signal amplitudes.

Reference

[1] G. C. Carter, "Coherence and time delay estimation," *Proceedings of IEEE*, vol. 75, no. 2, pp. 236–255, February 1987.

[2] Y. T. Chan, J. M. Riley, and J. B. Plant, "A parameter estimation approach to time-delay estimation and signal detection," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-28, no. 1, pp. 8–16, February 1980.

[3] B. M. Bell and T. E. Ewart, "Separating multipaths by global optimization of multidimensional matched filter," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. ASSP-34, pp. 1029–1037, October 1986.

[4] R. J. Tremblay, G. C. Carter, and D. W. Lytle, "A practical approach to the estimation of amplitude and time-delay parameters of a composite signal," *IEEE Journal of Oceanic Engineering*, vol. OE-12, no. 1, pp. 273–278, January 1987.

[5] R. J. Vaccaro, C. S. Ramalingam, and D. W. Tufts, "Least-squares time-delay estimation for transient signals in a multipath environment," *Journal of Acoustical Society of America*, vol. 92, no. 1, pp. 210–218, July 1992.

[6] T. G. Manickam, R. J. Vaccaro, and D. W. Tufts, "A least-squares algorithm for multipath time-delay estimation," *IEEE Transactions on Signal Processing*, vol. 42, pp. 3229–3233, November 1994.

[7] A. D. Blackowiak and S. D. Rajan, "Multipath arrival estimates using simulated annealing: Application to crosshole tomography experiment," *IEEE Journal of Oceanic Engineering*, vol. 20, pp. 157–165, July 1995.

[8] R. J. Tremblay, *Development and analysis of echo classification using time delays.* PhD thesis, University of Washington, 1995.

[9] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. AP-34, pp. 276–280, March 1986.

[10] D. W. Tufts, R. Kumaresan, and I. Kirsteins, "Data adaptive signal estimation by singular value decomposition of a data matrix," *Proceedings of the IEEE*, vol. 70, pp. 684–685, June 1982.

[11] Y. Bian and D. Last, "Eigen-decomposition techniques for Loran-C skywave estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, pp. 117–124, January 1997.

[12] I. P. Kirsteins, "High resolution time delay estimation," *Proceedings of ICASSP 87*, pp. 451–454, April 1987.

[13] I. P. Kirsteins and A. C. Kot, "Performance analysis of a high resolution time delay estimation algorithm," *Proceedings of ICASSP 90*, pp. 2767–2770, April 1990.

[14] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, pp. 47–60, November 1996.

[15] M. Feder and E. Weinstein, "Parameter estimation of superimposed signals using the EM algorithm," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 36, pp. 477–489, April 1988.

[16] C. B. Officer, *Introduction to the Theory of Sound Transmission.* New York,NY: McMgraw-Hill, 1958.

[17] T. Söderström and P. Stoica, *System Identification.* London, U.K.: Prentice-Hall International, 1989.

[18] J. Li and R. Wu, "An efficient algorithm for time delay estimation," *IEEE Transactions on Signal Processing*, vol. 46, pp. 2231–2235, August 1998.

[19] P. Stoica and T. Söderström, "On reparametrization of loss functions used in estimation and the invariance principle," *Signal Processing*, vol. 17, pp. 383–387, August 1989.

[20] A. M. Bruckstein, T. J. Shan, and T. Kailath, "The resolution of overlapping echoes," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-33, pp. 1357–1367, December 1985.

[21] *TMC2310*, TRW LSI Products, Inc.

[22] *A Users' Guide for the A41102*, Austeck Microsystems, 1988.

[23] R. Roy, A. Paulraj, and T. Kailath, "ESPRIT – A subspace rotation approach to estimation of parameters of cisoids in noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, pp. 1340–1342, October 1986.

[24] S. M. Kay, *Modern Spectral Estimation: Theory and Application.* Englewood Cliffs, NJ: Prentice-Hall, 1988.

[25] P. Stoica and R. L. Moses, *Introduction to Spectral Analysis.* Englewood Cliffs, NJ: Prentice-Hall, 1997.

[26] G. W. Stewart, *Introduction to Matrix Computations.* New York, NY: Academic Press, Inc., 1973.

[27] W. I. Zangwill, *Nonlinear Programming: A Unified Approach.* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1967.

[28] P. W. Zehna, "Invariance of maximum likelihood estimation," *Annual Mathematic Statistics*, vol. 37, p. 755, 1966.

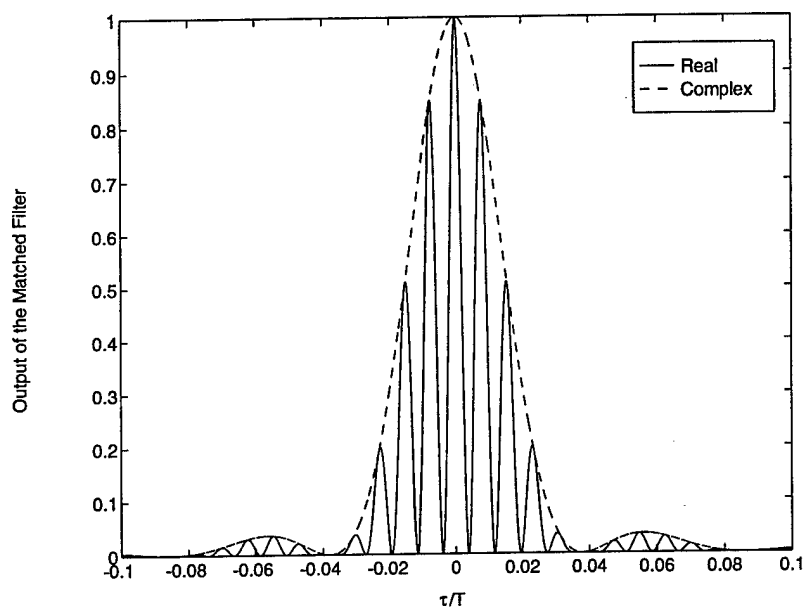
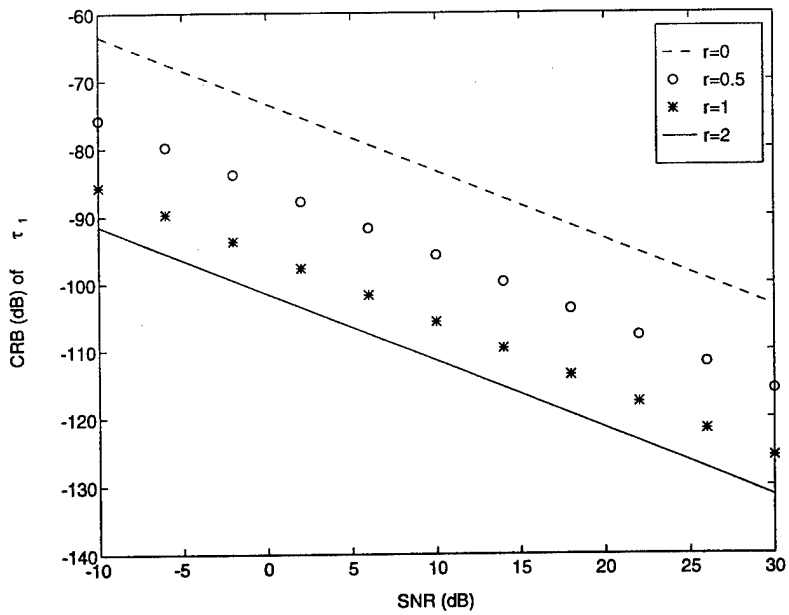
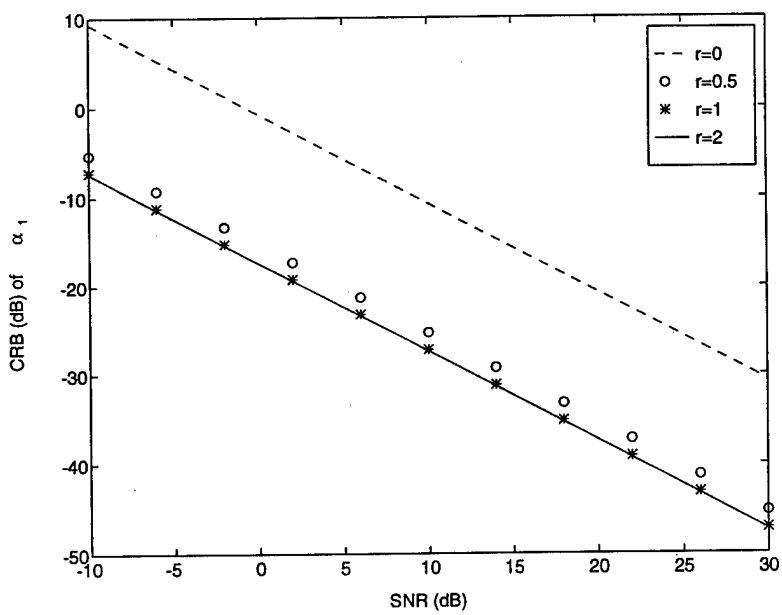


Figure 12.1: Comparison of the cost function obtained by assuming the real-valued signal amplitudes to be complex-valued (dashed line) and the true cost function (solid line).

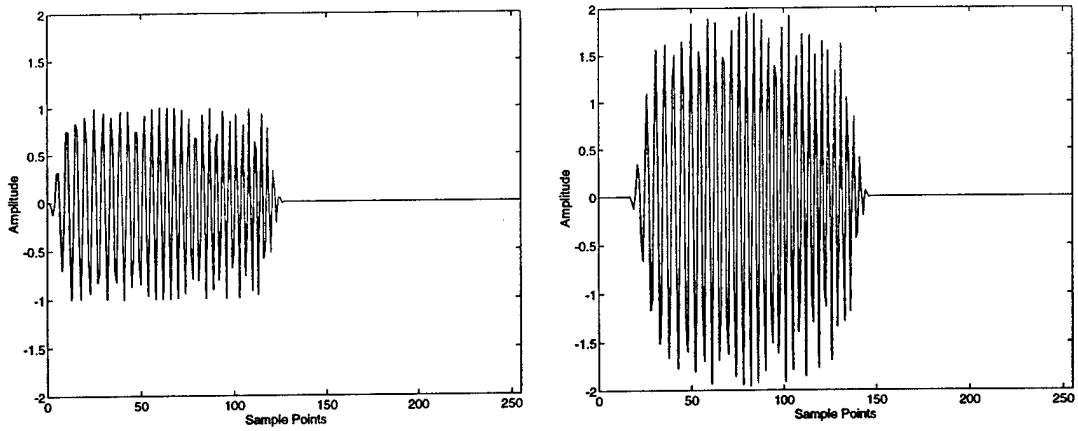


(a)

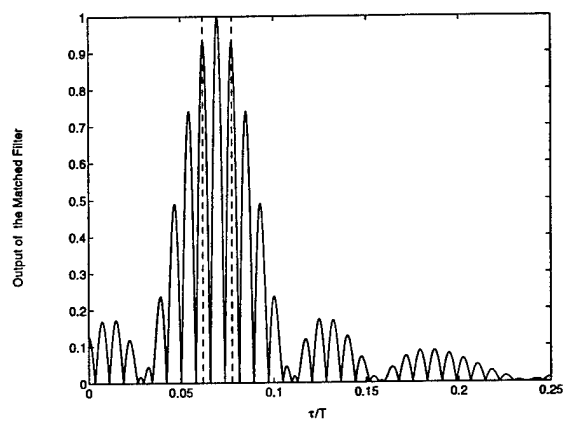


(b)

Figure 12.2: Effects of the carrier frequency ($f_0 = rB_s$) of the transmitted signal on the CRBs for (a) τ_1 and (b) α_1.



(a) (b)



(c)

Figure 12.3: (a) Transmitted signal waveform $s(nT_s)$. (b) Noise-free observed signal waveform $y(nT_s)$. (c) Matched filter output of $y(nT_s)$ (solid line), where the two vertical dashed lines indicate the true normalized arrival times of the two signals.

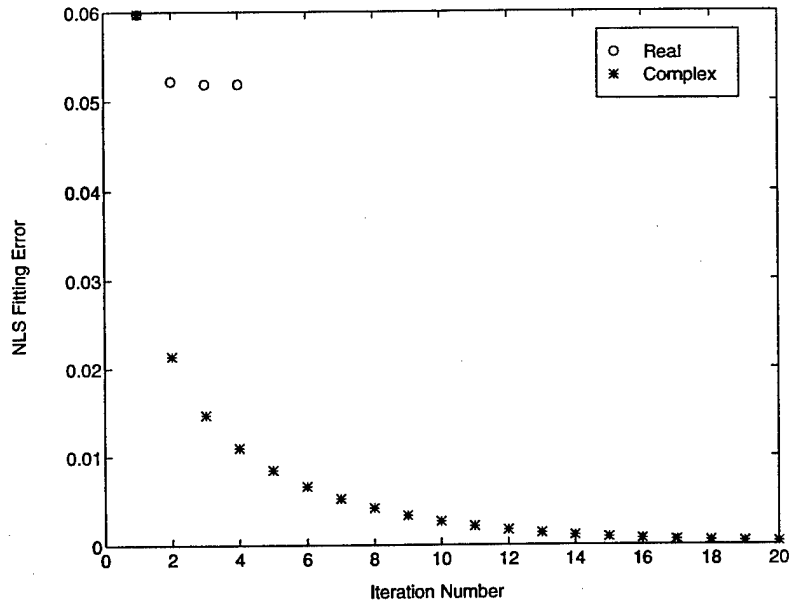


Figure 12.4: Convergence properties of WRELAX for assuming signal amplitudes being complex-valued ("$*$") and real-valued ("o").

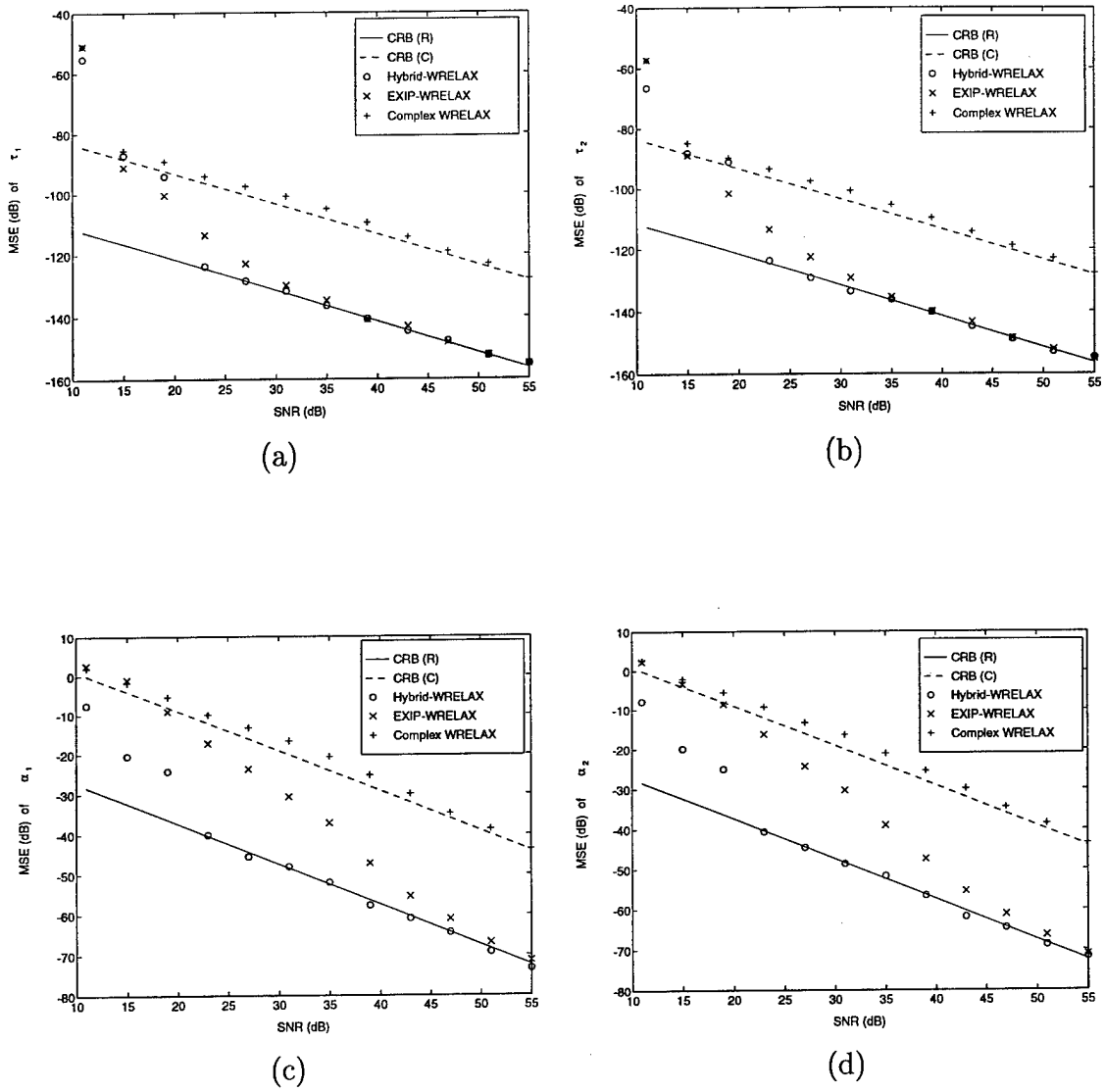


Figure 12.5: Comparison of the MSEs of the WRELAX for assuming complex-valued signal amplitudes ("+"), Hybrid-WRELAX ("o"), and EXIP-WRELAX ("×") with the CRBs corresponding to complex-valued (dashed line) and real-valued (solid line) signal amplitudes for (a) τ_1, (b) τ_2, (c) α_1, and (d) α_2.

13. Super Resolution Time Delay Estimation via MODE-WRELAX

13.1 Introduction

Time delay estimation is a well-known problem that arises frequently in radar, sonar, radio navigation, geophysical/seismic exploration, wireless communication, and medical imaging. It falls mainly into two categories: one is the Time of Arrival (TOA) estimation based on one sensor, such as radar and active sonar; the other one is the Time Difference of Arrival (TDOA) estimation based on multiple sensors, such as passive sonar, radio positioning and navigation systems. Practical signal models and configurations of sensors depend on specific applications. For example, the transmitted signal waveform is known for the former case while it is usually not for the latter situation. In this chapter, we only consider the former time delay estimation problem based on one sensor with known transmitted signal shapes.

Matched filter approach is the simplest one for this problem. By correlating the received signal with the known transmitted signal (complex conjugated) and searching for the peaks of the cross correlation outputs, we can obtain the time delay and amplitude estimates from the peak positions and heights correspondingly. The major drawback of the matched filter approach lies in the fact that it cannot resolve two signals with a time spacing less than the reciprocal of the signal bandwidth. How to resolve very closely spaced signals is the focus of this chapter and this issue has received a significant amount of attentions in the past two decades. Efficient solution to this problem has many potential applications including feature extraction via high range resolution radar and synthetic aperture radar, detection and classification of roadway subsurface anomalies by using ultra wideband ground penetrating radar, and multipath separation in sonar and wireless communications.

Many super resolution time delay estimation techniques have been devised recently. In the time domain, the received signal can be modeled as the sum of multiple scaled and delayed replicas of the transmitted signal plus noise. In the Fourier frequency domain, this data model becomes the sum of multiple weighted complex exponentials plus noise. The frequency domain data model is similar to those used for the sinusoidal parameter and

angle estimation problems except that the complex exponentials are weighted by the known signal spectrum. Based on this observation, many existing sinusoidal frequency and angle estimation algorithms, such as MUSIC [1], linear prediction [2], and maximum likelihood, are suggested to solve this time delay estimation problem [3, 4, 5, 6]. However, they are best suited for complex-valued signals with special shapes (such as flat band-limited spectrum). A computationally efficient approach based on the Expectation Maximization (EM) algorithm [7] is proposed in [8] that decouples a complicated multidimensional optimization problem into a sequence of multiple separate one-dimensional optimization problems. However, the EM method is very sensitive to initial conditions and no systematic initialization method is given in [8]. The separation of multipaths from real-valued bandpass underwater acoustic signals with highly oscillating correlation functions is a very challenging issue and is addressed in [9, 10, 11]. The algorithms proposed in [9, 10, 11] are all based on a nonlinear least squares (NLS) fitting criterion and differ with each other in the way how the NLS cost function is optimized. Except for the EM algorithm presented in [8], all other aforementioned algorithms estimate the time delays and the amplitudes separately, i.e., the delays are estimated first and then they are used with a linear least-squares approach to estimate the amplitudes. When the time delay estimates are very close to each other, the amplitude estimates can be very poor due to the ill-conditioning problem.

In our previous paper [12], a Weighted Fourier transform and RELAXation based method (referred to as WRELAX) is presented to minimize a frequency domain NLS cost function. The most striking feature of the WRELAX algorithm is that it decouples the multidimensional optimization problem into a series of one-dimensional optimization problems in a conceptually and computationally simple way. Compared with other existing algorithms, WRELAX is more systematic and efficient and has less limitations on the signal shapes. High estimation accuracy can be attained for both the delays and the amplitudes since they are estimated jointly and no matrix inversion is involved. WRELAX can be applied to either complex- or real-valued signals by minimizing slightly different cost functions. WRELAX was extended in [13] to deal with the real-valued signals with highly oscillatory correlation functions. The resolution of WRELAX are much higher than that of the conventional

matched filter approach. However, when the signals are very closely spaced in arrival times, the convergence speed of WRELAX decreases rapidly.

In this chapter, we study how MODE [14, 15] can be used with our efficient WRELAX algorithm for super resolution time delay estimation. The new algorithm is referred to as MODE-WRELAX. Although MODE can provide very poor amplitude estimates and WRELAX has the slow convergence problem, MODE-WRELAX outperforms both MODE and WRELAX. MODE-WRELAX can be used for both complex- and real-valued signals (including those with highly oscillatory correlation functions).

The remainder of this chapter is organized as follows. Section 13.2 establishes the data model and states the problem of interest. The MODE-WRELAX algorithm is presented in Section 13.3. Efficient implementation of the algorithm is given in Section 13.4. Numerical examples are provided in Section 13.5 to illustrate the performance of MODE-WRELAX. Finally, Section 13.6 contains our conclusions.

13.2 Data Model and Problem Statement

For conventional radar and active sonar, the most commonly used probing signal has the following form:

$$\tilde{s}(t) = m(t)\cos[2\pi f_0 t + \theta(t)], \tag{13.1}$$

where $m(t)$ and $\theta(t)$ denote the amplitude and phase modulations, respectively, and f_0 represents the carrier frequency. Usually, the above real-valued bandpass signal is converted into inphase (I) and quadrature (Q) components. The classical approach for this conversion is the analog quadrature demodulation by utilizing sine and cosine mixers and low-pass filters. However, this approach is highly sensitive to analog component mismatch between the I and Q channels. The state-of-the-art technology for such a conversion is the digital quadrature sampling [16], whose performance is mainly limited by the A/D quantization. The I and Q components thus obtained can be used to form the following complex-valued analytic signal

$$s(t) = m(t)e^{j\theta(t)}. \tag{13.2}$$

The received signal from a scatterer can be described as

$$\tilde{r}(t) = Am(t-\tau)\cos[2\pi f_0(t-\tau) + \theta(t-\tau) + \phi], \tag{13.3}$$

where A, ϕ, and τ denote the amplitude, phase, and time delay determined by the scatterer and the propagation medium. After quadrature demodulation, the received signal can be expressed as a scaled and delayed version of the complex analytic signal $s(t)$

$$r(t) = \alpha s(t-\tau), \tag{13.4}$$

where α represents the complex-valued amplitude and has the form

$$\alpha = Ae^{j\phi}e^{-j2\pi f_0\tau}. \tag{13.5}$$

For certain underwater acoustic signal propagation environment [9, 10, 11, 17], the received signal $\tilde{r}(t)$ itself due to a scatterer is just a scaled and delayed version of the probing signal $\tilde{s}(t)$, which corresponds to the case where no phase is induced by the multipath channel. No quadrature demodulation is needed in this case and (13.4) still holds with $r(t) = \tilde{r}(t)$, $s(t) = \tilde{s}(t)$, and $\alpha = A$. This corresponds to the real-valued signal model, which is also valid in other applications, such as the ultra wideband ground penetrating radar where the probing signal is carrier-free and not sinusoidal.

The time delay estimation data model considered in this chapter has the following general form:

$$y(t) = \sum_{l=1}^{L} \alpha_l s(t-\tau_l) + e(t) \quad 0 \le t \le T, \tag{13.6}$$

where $s(t)$, $0 \le t \le T_0$, represents an *arbitrary* known transmitted signal , $y(t)$ denotes the received signal, which is composed of L replicas of $s(t)$ with different amplitudes $\{\alpha_l\}_{l=1}^{L}$ and delays $\{\tau_l\}_{l=1}^{L}$, and $e(t)$ is the additive noise, which is modeled as a zero-mean white Gaussian random process. Without loss of generality, we assume that $s(t)$, $y(t)$, $e(t)$, and $\{\alpha_l\}_{l=1}^{L}$ are either all complex-valued or all real-valued.

The sampled received signal can be written as

$$y(nT_s) = \sum_{l=1}^{L} \alpha_l s(nT_s - \tau_l) + e(nT_s), \quad n = 0, 1, \cdots, N-1, \tag{13.7}$$

where T_s is the sampling period and is equal to the reciprocal of the sampling frequency f_s.

Our problem of interest herein is to estimate $\{\alpha_l, \tau_l\}_{l=1}^{L}$ from $\{y(nT_s)\}_{n=0}^{N-1}$ with known $s(t)$, $0 \leq t \leq T_0$, or $\{s(nT_s)\}_{n=0}^{N-1}$ when the signals are very closely spaced.

Although we could solve the estimation problem in the time domain [3, 8, 9, 11], we prefer to do it in the frequency domain. This is because for the time domain processing methods, we could be restricted to using the discrete values of $\{\tau_l\}_{l=1}^{L}$ if we only know the sampled version of $s(t)$. For this case, if a more accurate delay estimate is required, then one has to resort to interpolation [9]. This inconvinence can be avoided by transforming the problem into the frequency domain, where $\{\tau_l\}_{l=1}^{L}$ can take on a continuum of values. Let $Y(k)$, $S(k)$, and $E(k)$, $k = -N/2, -N/2+1, ..., N/2-1$, denote the discrete Fourier transforms (DFT's) of $y(nT_s)$, $s(nT_s)$, and $e(nT_s)$, respectively. Provided that aliasing is negligible, then $Y(k)$ can be written as:

$$Y(k) = S(k) \sum_{l=1}^{L} \alpha_l e^{j\omega_l k} + E(k), \tag{13.8}$$

where

$$\omega_l = -\frac{2\pi\tau_l}{NT_s}. \tag{13.9}$$

Note that the time delay estimation problem is similar to the sinusoidal parameter estimation problem except that the exponential signals are weighted by the known signal spectrum. If we divided both sides of (13.8) by $S(k)$, the problem would become identical to the sinusoidal parameter estimation problem. Yet we should not do so for the following reasons: first, $S(k)$ could be zero for some k; second, the noise $E(k)/S(k)$ will no longer be a white noise even when $E(k)$ is white; third, when $E(k)$ is a white noise, the larger the $S(k)$ at sample k, the higher the SNR of the corresponding $Y(k)$ and hence dividing $Y(k)$ by $S(k)$ will de-emphasize those $Y(k)'s$ that have high SNRs. Because of this, many well-known sinusoidal parameter estimation algorithms, such as MUSIC [1], ESPRIT [18],

PRONY [19], MODE [14, 15], are not best suited to our problem of interest. Further, since these algorithms are designed for complex-valued amplitudes, they cannot provide the best possible performance for real-valued signals.

13.3 The MODE-WRELAX Algorithm

In this section, we will first present the MODE-WRELAX algorithm for complex-valued signals, and then extend it to real-valued signals (especially those with highly oscillatory correlation functions) and also multiple look cases.

13.3.1 MODE-WRELAX for Complex-Valued Signals

Assume that $(\cdot)^T$ denote the transpose and let

$$\mathbf{Y} = \begin{bmatrix} Y(-N/2) & Y(-N/2+1) & \cdots & Y(N/2-1) \end{bmatrix}^T, \tag{13.10}$$

$$\mathbf{S} = \text{diag}\left\{ S(-N/2), \quad S(-N/2+1), \quad \cdots, \quad S(N/2-1) \right\}, \tag{13.11}$$

$$\mathbf{E} = \begin{bmatrix} E(-N/2) & E(-N/2+1) & \cdots & E(N/2-1) \end{bmatrix}^T, \tag{13.12}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_L \end{bmatrix}^T, \tag{13.13}$$

and

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}(\omega_1) & \mathbf{a}(\omega_2) & \cdots & \mathbf{a}(\omega_L) \end{bmatrix}^T, \tag{13.14}$$

with

$$\mathbf{a}(\omega_l) = \begin{bmatrix} e^{j\omega_l(-N/2)} & e^{j\omega_l(-N/2+1)} & \ldots & e^{j\omega_l(N/2-1)} \end{bmatrix}^T. \tag{13.15}$$

Then the data model (13.8) can be written in the following vector form:

$$\mathbf{Y} = \mathbf{S}\mathbf{A}\boldsymbol{\alpha} + \mathbf{E}. \tag{13.16}$$

When $\mathbf{S}$ is an identity matrix, then the above time delay estimation issue becomes a sinusoidal parameter estimation problem and MODE is an asymptotically statistically efficient estimator of $\{\omega_l\}_{l=1}^{L}$ for complex-valued signals [14, 15]. The MODE algorithm [14, 15] can be easily extended to the data model in (13.16) where $\mathbf{S}$ is an arbitrary diagonal matrix as follows. The MODE estimates $\{\hat{\omega}_l\}_{l=1}^{L}$ of $\{\omega_l\}_{l=1}^{L}$ can be obtained by minimizing the following cost function

$$C_1(\{\omega_l\}_{l=1}^{L}) = \mathbf{Y}^H \mathbf{P}_{\tilde{\mathbf{A}}}^{\perp} \mathbf{Y}, \tag{13.17}$$

where $(\cdot)^H$ denotes the conjugate transpose and

$$\mathbf{P}_{\tilde{\mathbf{A}}}^{\perp} = \mathbf{I} - \tilde{\mathbf{A}} \left(\tilde{\mathbf{A}}^H \tilde{\mathbf{A}} \right)^{-1} \tilde{\mathbf{A}}^H, \tag{13.18}$$

with $\mathbf{I}$ denoting the identity matrix and

$$\tilde{\mathbf{A}} = \mathbf{S}\mathbf{A}. \tag{13.19}$$

To avoid the search over the parameter space, $C_1(\{\omega_l\}_{l=1}^{L})$ can also be reparametrized in terms of another parameter vector $\mathbf{b} = \begin{bmatrix} b_0 & b_1 & \cdots & b_L \end{bmatrix}^T$, where $\{b_l\}_{l=0}^{L}$ are the coefficients of the following polynomial:

$$b(z) \triangleq \sum_{l=0}^{L} b_l z^{L-l} \triangleq b_0 \prod_{l=1}^{L} (z - e^{j\omega_l}); \; b_0 \neq 0. \tag{13.20}$$

Since the polynomial $b(z)$ in (13.20) has all of its zeros on the unit circle, its coefficients $\{b_l\}$ satisfy the conjugate symmetry constraint [14]:

$$b_l = b_{L-l}^*, \quad l = 0, 1, \cdots, L, \tag{13.21}$$

where $(\cdot)^*$ denotes the complex conjugate. Let

$$\mathbf{B} = \begin{bmatrix} b_0 & & 0 \\ \vdots & \ddots & \\ b_L & & b_0 \\ & \ddots & \vdots \\ 0 & & b_L \end{bmatrix} \in \mathcal{C}^{N \times (N-L)}. \tag{13.22}$$

Assume that the diagonal elements of $\mathbf{S}$ are nonzero (see **Remark 1** for more discussions). Let

$$\tilde{\mathbf{B}} = \mathbf{S}^{-H}\mathbf{B}. \tag{13.23}$$

It can be readily verified that $\mathbf{B}^H\mathbf{A} = \mathbf{0}$ and hence $\tilde{\mathbf{B}}^H\tilde{\mathbf{A}} = \mathbf{0}$. Then $\mathbf{P}_{\tilde{\mathbf{A}}}^{\perp} = \tilde{\mathbf{B}}\left(\tilde{\mathbf{B}}^H\tilde{\mathbf{B}}\right)^{-1}\tilde{\mathbf{B}}^H$ and minimizing $C_1(\{\omega_l\}_{l=1}^L)$ in (13.17) is equivalent to minimizing

$$C_2(\{b_l\}_{l=0}^L) = \mathbf{Y}^H\tilde{\mathbf{B}}\left(\tilde{\mathbf{B}}^H\tilde{\mathbf{B}}\right)^{-1}\tilde{\mathbf{B}}^H\mathbf{Y}. \tag{13.24}$$

Note that $\tilde{\mathbf{B}}^H\tilde{\mathbf{B}}$ in (13.24) can be replaced by a consistent estimate without affecting the asymptotically statistical efficiency of the minimizer of (13.24). Hence $\hat{\mathbf{b}}$ can be obtained computationally efficiently as follows:

$$\hat{\mathbf{b}} = \arg\min_{\mathbf{b}}\left[\mathbf{Y}^H\mathbf{S}^{-H}\mathbf{B}\left(\hat{\mathbf{B}}_0^H\mathbf{S}^{-1}\mathbf{S}^{-H}\hat{\mathbf{B}}_0\right)^{-1}\mathbf{B}^H\mathbf{S}^{-1}\mathbf{Y}\right], \tag{13.25}$$

where $\hat{\mathbf{B}}_0$ is the initial estimate of $\mathbf{B}$ obtained by replacing $\mathbf{b}$ with $\hat{\mathbf{b}}^{(0)}$ in (13.22). The initial value $\hat{\mathbf{b}}^{(0)}$ is obtained by setting $\tilde{\mathbf{B}}^H\tilde{\mathbf{B}}$ in (13.24) to $\mathbf{I}$:

$$\hat{\mathbf{b}}^{(0)} = \arg\min_{\mathbf{b}}\left[\mathbf{Y}^H\mathbf{S}^{-H}\mathbf{B}\mathbf{B}^H\mathbf{S}^{-1}\mathbf{Y}\right]. \tag{13.26}$$

To avoid the trivial solution $\mathbf{b} = \mathbf{0}$, we should impose $\| \mathbf{b} \| = 1$ (where $\| \cdot \|$ denotes the Euclidean norm) in (13.25) and (13.26) or some other similar constraints. (For detailed implementation steps, see Section 4.) The estimates $\{\hat{\omega}_l\}_{l=1}^L$ of $\{\omega_l\}_{l=1}^L$ are the phases of the roots of the polynomial $\sum_{l=0}^L \hat{b}_l z^{L-l}$. Once $\{\hat{\omega}_l\}_{l=1}^L$ are obtained, the amplitudes $\boldsymbol{\alpha}$ are estimated by applying the linear least-squares approach to

$$\mathbf{Y} \approx \mathbf{S}\hat{\mathbf{A}}\boldsymbol{\alpha}, \tag{13.27}$$

where $\hat{\mathbf{A}}$ is formed by replacing $\{\omega_l\}_{l=1}^L$ with $\{\hat{\omega}_l\}_{l=1}^L$ in (13.14).

Remark 1: MODE cannot be implemented efficiently to avoid the search over the parameter space when $S(k) = 0$ for some k. The most commonly used complex analytic signal $s(t)$ is low-pass. For this case, we can select a contiguous segment of $\mathbf{Y}$ satisfying $|S(k)| > 0, \quad K_1 \le k \le K_2$, and preferrably with $|S(k)|$ above a certain threshold to avoid numerical problems. We can then apply MODE to the segment $\{Y(k)\}_{k=K_1}^{K_2}$ to estimate $\{\omega_l\}_{l=1}^L$.

Remark 2: The amplitude estimates given above can be very poor when the SNR is not sufficiently high. This is because some of the MODE estimates $\{\hat{\omega}_l\}_{l=1}^{L}$ can be so closely spaced that $\hat{\mathbf{A}}$ in (13.27) is seriously ill-conditioned. We use a simple spacing adjustment scheme to avoid this problem. After obtaining the MODE estimates $\{\hat{\omega}_l\}_{l=1}^{L}$ of $\{\omega_l\}_{l=1}^{L}$, we first sort them in the ascending order and then check the spacing between two adjacent estimates. If the distance between any two estimates, say $\hat{\omega}_1$ and $\hat{\omega}_2$ ($\hat{\omega}_1 \leq \hat{\omega}_2$), is smaller than a predefined threshold, say $\triangle\omega_t$, we adjust the estimates by replacing $\hat{\omega}_1$ with $\hat{\omega}_1 - 0.5\triangle\omega_t$ and $\hat{\omega}_2$ with $\hat{\omega}_2 + 0.5\triangle\omega_t$. The amplitudes are then estimated using the adjusted estimates of $\{\omega_l\}_{l=1}^{L}$. This spacing adjustment step is *ad hoc* but can be used to provide good initial delay and amplitude estimates to replace the first $L-1$ steps of WRELAX.

The MODE estimates $\{\hat{\omega}_l\}_{l=1}^{L}$ of $\{\omega_l\}_{l=1}^{L}$ and $\{\hat{\alpha}_l\}_{l=1}^{L}$ of $\{\alpha_l\}_{l=1}^{L}$, which may not be optimal, especially for real-valued signals, can be refined by using the last step of the WRELAX algorithm.

WRELAX is a relaxation-based minimizer of the following nonlinear least-squares (NLS) criterion:

$$C_3(\{\alpha_l, \omega_l\}_{l=1}^{L}) = \| \mathbf{Y} - \sum_{l=1}^{L} \alpha_l \mathbf{Sa}(\omega_l) \|^2 . \tag{13.28}$$

When $e(nT_s)$ is a zero-mean white Gaussian random process, $E(k)$ is also white since DFT is a unitary transformation. For this white noise case, the NLS approach is the same as the maximum likelihood (ML) method. When $e(nT_s)$ is not white, NLS approach can still provide estimates with good statistical accuracy [20].

Minimizing $C_3(\{\alpha_l, \omega_l\}_{l=1}^{L})$ with respect to the unknown parameters is a highly nonlinear optimization problem and it is very difficult to find the global minimum. WRELAX decouples the multi-dimensional optimization problem into a sequence of one-dimensional optimization problems in a conceptually and computationally simple way. WRELAX estimates the delays and amplitudes jointly and requires only a sequence of weighted Fourier transforms. When the signals are not spaced very closely, WRELAX usually converges in a few steps. However, when the signals are very closely spaced, the convergence speed of WRELAX is very slow. Yet by using the above MODE algorithm to obtain the initial con-

ditions and then using the last step of WRELAX to refine them, super resolution time delay estimation can be achieved with a fast convergence speed.

Before we present the MODE-WRELAX algorithm, let us consider the following preparations. Let

$$\mathbf{Y}_l = \mathbf{Y} - \sum_{i=1, i\neq l}^{L} \hat{\alpha}_i[\mathbf{Sa}(\hat{\omega}_i)], \tag{13.29}$$

where $\{\hat{\alpha}_i, \hat{\omega}_i\}_{i=1,i\neq l}$ are assumed to be given . Then (13.28) becomes

$$C_4(\alpha_l, \omega_l) = \| \mathbf{Y}_l - \alpha_l \mathbf{Sa}(\omega_l) \|^2 . \tag{13.30}$$

Minimizing $C_4(\alpha_l, \omega_l)$ with respect to ω_l and the complex-valued α_l yields

$$\hat{\omega}_l = \arg\max_{\omega_l} \left| \mathbf{a}^H(\omega_l)(\mathbf{S}^*\mathbf{Y}_l) \right|^2 , \tag{13.31}$$

and

$$\hat{\alpha}_l = \left. \frac{\mathbf{a}^H(\omega_l)(\mathbf{S}^*\mathbf{Y}_l)}{\| \mathbf{S} \|_F^2} \right|_{\omega_l = \hat{\omega}_l} , \tag{13.32}$$

where $\| \cdot \|_F$ denotes the *Frobenius* norm [21].

With the above preparations, we now present the steps of the MODE-WRELAX algorithm for complex-valued signals.

Step (1): Select a contiguous segment of data vector $\mathbf{Y}$ (for MODE use only) so that $|S(k)| > 0, \quad K_1 \leq k \leq K_2$. Apply MODE to the segment to obtain $\{\hat{\omega}_l\}_{l=1}^L$. Adjust $\{\hat{\omega}_l\}_{l=1}^L$ so that the minimum spacing of $\{\hat{\omega}_l\}_{l=1}^L$ is at least $\triangle\omega_t$. Obtain the estimates $\{\hat{\alpha}_l\}_{l=1}^L$ of $\{\alpha_l\}_{l=1}^L$ by using (13.27).

Step (2): Refine the estimates obtained in Step (1) by using the last step of WRELAX. That is, compute $\mathbf{Y}_1$ by using $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=2}^L$ obtained in Step (1). Obtain $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1}$ from $\mathbf{Y}_1$ by using (13.31) and (13.32). Next, compute $\mathbf{Y}_2$ by using the updated $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1,3,\cdots,L}$ and determine $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=2}$ from $\mathbf{Y}_2$. Then compute $\mathbf{Y}_3$ by using the updated $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1,2,4,\cdots,L}$ and determine $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=3}$ from $\mathbf{Y}_3$. Continue this procedure and similarly determine $\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=L}$ from $\mathbf{Y}_L$. Repeat the above process until "practical convergence" (to be discussed later on).

The "practical convergence" in the iterations of the above WRELAX algorithm may be determined by checking the relative change of the cost function $C_3(\{\hat{\omega}_l, \hat{\alpha}_l\}_{l=1}^L)$ in (13.28)

between two consecutive iterations. The algorithm is bound to converge to at least a local minimum point under mild conditions [22].

Once $\{\hat{\omega}_l\}_{l=1}^{L}$ are determined, the delay estimates $\{\hat{\tau}_l\}_{l=1}^{L}$ of $\{\tau_l\}_{l=1}^{L}$ can be computed by using (13.9) with $\{\omega_l\}_{l=1}^{L}$ replaced by $\{\hat{\omega}_l\}_{l=1}^{L}$.

Similarly, we can use MODE as an initialization method for the EM time delay estimation algorithm [7], which is referred to as MODE-EM. However, we have found through numerical simulations that the convergence speed of MODE-EM is slower than that of MODE-WRELAX.

13.3.2 MODE-WRELAX for Real-Valued Signals

Real-valued signals are often bandpass signals that occur, for example, in underwater sonar and ultra wideband ground penetrating radar applications. Bandpass signals have highly oscillatory correlation functions, which makes the super resolution time delay estimation problem more difficult. The larger the center frequency of the pass band, the sharper the oscillation of the correlation function.

Consider the data model expressed by (13.8). When the signals $s(t)$, $y(t)$, and $e(t)$ are all real-valued, their Fourier transforms are conjugate symmetric, i.e., $Y(-k) = Y^*(k)$, $S(-k) = S^*(k)$, and $E(-k) = E^*(k)$, $\quad k = 1, 2, \cdots, N/2-1$, and $Y(-N/2)$, $Y(0)$, $S(-N/2)$, $S(0)$, $E(-N/2)$, and $E(0)$ are real-valued. Define

$$\begin{aligned} \mathbf{W} &= \text{diag}\left\{ W(-N/2),\ W(-N/2+1),\ \cdots,\ W(-1),\ W(0) \right\} \\ &= \text{diag}\left\{ \tfrac{1}{\sqrt{2}},\ 1,\ \cdots,\ 1,\ \tfrac{1}{\sqrt{2}} \right\}, \end{aligned} \tag{13.33}$$

$$\bar{\mathbf{Y}} = \mathbf{W} \left[Y(-N/2) \quad Y(-N/2+1) \quad \cdots \quad Y(0) \right]^T, \tag{13.34}$$

$$\bar{\mathbf{S}} = \mathbf{W} \ \text{diag}\left\{ S(-N/2),\ S(-N/2+1),\ \cdots,\ S(0) \right\}, \tag{13.35}$$

$$\bar{\mathbf{E}} = \mathbf{W} \left[E(-N/2) \quad E(-N/2+1) \quad \cdots \quad E(0) \right]^T, \tag{13.36}$$

and

$$\bar{\mathbf{A}} = \begin{bmatrix} \bar{\mathbf{a}}(\omega_1) & \bar{\mathbf{a}}(\omega_2) & \cdots & \bar{\mathbf{a}}(\omega_L) \end{bmatrix}^T, \tag{13.37}$$

where

$$\bar{\mathbf{a}}(\omega_l) = \begin{bmatrix} e^{j\omega_l(-N/2)} & e^{j\omega_l(-N/2+1)} & \cdots & 1 \end{bmatrix}^T. \tag{13.38}$$

Then it follows that

$$\bar{\mathbf{Y}} = \bar{\mathbf{S}}\bar{\mathbf{A}}\boldsymbol{\alpha} + \bar{\mathbf{E}}. \tag{13.39}$$

Since the amplitudes $\{\alpha_l\}_{l=1}^L$ are real-valued and due to the conjugate symmetry of $Y(k)$, $S(k)$, and $E(k)$, it can be proven that minimizing $C_3(\{\alpha_l, \omega_l\}_{l=1}^L)$ is equivalent to minimizing

$$C_5(\{\alpha_l, \omega_l\}_{l=1}^L) = \| \bar{\mathbf{Y}} - \sum_{l=1}^{L} \alpha_l \bar{\mathbf{S}}\bar{\mathbf{a}}(\omega_l) \|^2 . \tag{13.40}$$

For the case of white Gaussian noise, the above NLS approach is the same as the ML method. For bandpass real-valued signals, $C_5(\{\alpha_l, \omega_l\}_{l=1}^L)$ is a highly oscillatory cost function and is very difficult to find its global minimum. Although MODE is derived for complex-valued signals, we can apply it to $\bar{\mathbf{Y}}$ in (13.39) by assuming the real-valued amplitudes $\{\alpha_l\}_{l=1}^L$ to be complex-valued. These initial estimates are then refined by the WRELAX algorithm. Since the attraction domain of the cost function $C_5(\{\alpha_l, \omega_l\}_{l=1}^L)$ is extremely small, a very good initial condition is required to achieve the global convergence of any minimizer of $C_5(\{\alpha_l, \omega_l\}_{l=1}^L)$. The MODE estimates are first refined by WRELAX by assuming $\{\alpha_l\}_{l=1}^L$ to be complex-valued since the attraction domain of $C_5(\{\alpha_l, \omega_l\}_{l=1}^L)$ becomes much larger when assuming the real-valued $\{\alpha_l\}_{l=1}^L$ to be complex-valued [10]. The so-obtained estimates are refined again by WRELAX by using the fact that $\{\alpha_l\}_{l=1}^L$ are real-valued. The cost functions of WRELAX are changed slightly when the signals are real-valued. Let

$$\bar{\mathbf{Y}}_l = \bar{\mathbf{Y}} - \sum_{i=1, i\neq l}^{L} \hat{\alpha}_i [\bar{\mathbf{S}}\bar{\mathbf{a}}(\hat{\omega}_i)], \tag{13.41}$$

where $\{\hat{\alpha}_i, \hat{\omega}_i\}_{i=1, i\neq l}$ are assumed to be given . Then (13.40) becomes

$$C_6(\alpha_l, \omega_l) = \| \bar{\mathbf{Y}} - \alpha_l \bar{\mathbf{S}}\bar{\mathbf{a}}(\omega_l) \|^2 . \tag{13.42}$$

Minimizing $C_6(\alpha_l, \omega_l)$ with respect to ω_l and real-valued α_l yields

$$\hat{\omega}_l = \arg\max_{\omega_l} \mathrm{Re}^2 \left[\bar{\mathbf{a}}^H(\omega_l)(\bar{\mathbf{S}}^*\bar{\mathbf{Y}}_l)\right], \tag{13.43}$$

and

$$\hat{\alpha}_l = \left.\frac{\mathrm{Re}\left[\bar{\mathbf{a}}^H(\omega_l)(\bar{\mathbf{S}}^*\bar{\mathbf{Y}}_l)\right]}{\| \bar{\mathbf{S}} \|_F^2}\right|_{\omega_l=\hat{\omega}_l}. \tag{13.44}$$

With the above preparations, we now present the steps of the MODE-WRELAX algorithm for real-valued signals.

Step (1): Select a contiguous segment of data vector $\bar{\mathbf{Y}}$ so that $|\bar{S}(k)| > 0, \quad \bar{K}_1 \leq k \leq \bar{K}_2$. By assuming the real-valued $\{\alpha_l\}_{l=1}^L$ to be complex-valued, obtain the estimates $\{\hat{\omega}_l\}_{l=1}^L$ and $\{\hat{\alpha}_l\}_{l=1}^L$ in the same way as Step (1) of the MODE-WRELAX algorithm for complex-valued signals.

Step (2): Refine the estimates obtained in Step (1) above by using the last step of WRELAX by assuming complex-valued signals. Take the real parts of the so-obtained amplitude estimates as the amplitude estimates $\{\hat{\alpha}_l\}_{l=1}^L$ of $\{\alpha_l\}_{l=1}^L$.

Step (3): Refine the estimates obtained in Step (2) above by using the last step of WRELAX and the fact that the signals are real-valued.

13.3.3 Extensions to Multiple Looks

The above algorithms are designed for the single look case. However, they are readily extended to the multiple look case where multiple independent measurements are available, which occurs, for example, when a radar emits a sequence of pulses for target detection. For the multiple look data model, it is assumed that the delays are fixed while the unknown amplitudes vary from scan to scan. Under this assumption, the extensions of both MODE and WRELAX are possible and the detailed derivations are omitted here due to the limited space.

13.4 Efficient Implementation of MODE-WRELAX

Since MODE-WRELAX is mainly composed of two blocks, MODE and WRELAX, below we consider the efficient implementation of each block.

13.4.1 MODE

As stated before, it is necessary to constrain $\mathbf{b}$ to avoid the trivial solution $\mathbf{b} = 0$ when minimizing $C_2(\{b_l\}_{l=0}^{L})$ in (13.24). Furthermore, the conjugate symmetric constraint in (13.21) can also be easily included to improve the performance. With conjugate symmetry, the number of unknowns is about halved. This does not guarantee that the zeros are on the unit circle. It is a necessary but not sufficient condition for the zeros to be on the unit circle. This constraint can be eliminated by reparameterizing $C_2(\{b_l\}_{l=0}^{L})$ in (13.24) with a real-valued vector $\boldsymbol{\beta} \in \mathcal{R}^{(L+1)\times 1}$ which satisfies

$$\mathbf{b} = \boldsymbol{\Gamma}\boldsymbol{\beta}, \tag{13.45}$$

where $\boldsymbol{\Gamma} \in \mathcal{C}^{(L+1)\times(L+1)}$ denotes a matrix made from $0, 1, \pm j$. Let

$$\mathbf{V} = \mathbf{S}^{-1}\mathbf{Y}, \tag{13.46}$$

and

$$\tilde{\mathbf{V}} = \begin{bmatrix} V(-N/2+L) & V(-N/2+L-1) & \cdots & V(-N/2) \\ V(-N/2+L+1) & V(-N/2+L) & \cdots & V(-N/2+1) \\ \vdots & \vdots & \ddots & \vdots \\ V(N/2-1) & V(N/2-2) & \cdots & V(N/2-L-1) \end{bmatrix}. \tag{13.47}$$

Then the optimization problem in (13.25) becomes

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^H \mathrm{Re}\left[\boldsymbol{\Gamma}^H \tilde{\mathbf{V}}^H \left(\hat{\mathbf{B}}_0^H \mathbf{S}^{-1}\mathbf{S}^{-H}\hat{\mathbf{B}}_0\right)^{-1} \tilde{\mathbf{V}}\boldsymbol{\Gamma}\right]\boldsymbol{\beta}, \tag{13.48}$$

with

$$\hat{\boldsymbol{\beta}}^{(0)} = \arg\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^H \mathrm{Re}\left(\boldsymbol{\Gamma}^H \tilde{\mathbf{V}}^H \tilde{\mathbf{V}}\boldsymbol{\Gamma}\right)\boldsymbol{\beta}. \tag{13.49}$$

To avoid the trivial minimizer $\boldsymbol{\beta} = \mathbf{0}$, we impose $\| \boldsymbol{\beta} \| = 1$.

To implement (13.48) and (13.49) efficiently, we note that $\left(\mathbf{S}^{-H}\hat{\mathbf{B}}_0\right)$ is a banded but not a Toeplitz matrix. Also, $\left(\hat{\mathbf{B}}_0^H\mathbf{S}^{-1}\mathbf{S}^{-H}\hat{\mathbf{B}}_0\right)$ is a banded Hermitian matrix with band width L, which is usually far less than the matrix dimension $N - L$.

The steps of minimizing (13.48) and the amount of computations (complex operations) required in each step are summarized as follows:

Step 1: Compute $\mathbf{C} = \left(\mathbf{S}^{-H}\hat{\mathbf{B}}_0\right)^H \mathbf{S}^{-H}\hat{\mathbf{B}}_0$.

It is easy to verify that the (i, j)th element of matrix $\mathbf{C}$ is given by:

$$C_{i,j} = \begin{cases} 0, & \text{for } |i-j| > L, \\ \sum_{l=0}^{L-i+j} \frac{\hat{b}_l^* \hat{b}_{l+i-j}}{|S_{i+l}|^2}, & \text{for } |i-j| \leq L \text{ and } i \geq j, \\ C_{ij}^*, & \text{for } |i-j| \leq L \text{ and } i < j. \end{cases} \tag{13.50}$$

This step requires $O\left(NL^2\right)$ flops.

Step 2: Compute the Cholesky decomposition $\mathbf{G}\mathbf{G}^H$ of $\mathbf{C}$.

Since $\mathbf{C}$ is a positive-definite banded Hermitian matrix with the band width L, the Cholesky factor $\mathbf{G}$ is a banded lower triangular matrix with band width L, which is calculated by the following iterative procedure [23]:

$$\begin{aligned} &\mathbf{G} = \mathbf{C}; \\ &\left\{ \begin{array}{l} \text{for} \quad j = 1 : N - L \\ \qquad \text{for} \quad l = \max\{1, j-L\} : j-1, \quad g_{j,j} = g_{j,j} - |g_{j,l}|^2; \quad \text{end} \\ \qquad g_{j,j} = \sqrt{g_{j,j}}; \\ \qquad \left\{ \begin{array}{l} \text{for} \quad i = j+1 : \min\{j+L, N-L\} \\ \qquad \text{for} \quad l = \max\{1, j-L\} : j-1, \quad g_{i,j} = g_{i,j} - g_{i,l}g_{j,l}^*; \quad \text{end}; \\ \qquad g_{i,j} = g_{i,j}/g_{j,j}; \\ \text{end} \end{array} \right. \\ \text{end} \end{array} \right. \end{aligned} \tag{13.51}$$

This step requires $O\left(NL^2\right)$ flops.

Step 3: Compute $\mathbf{Z} = \mathbf{G}^{-1}\tilde{\mathbf{V}}$.

Since $\mathbf{G}$ is a banded lower triangular matrix with band width L, the lth column of $\mathbf{Z}$ can be obtained by back substitution from matrix $\mathbf{G}$ and the lth column of $\tilde{\mathbf{V}}$, which is computed by the following procedure:

$$\mathbf{z}_l = l\text{th column of matrix } \tilde{\mathbf{V}};$$

$$\begin{cases} \text{for} \quad j = 1 : N - L \\ \qquad z_l(j) = z_l(j)/g_{j,j}; \\ \qquad\quad \text{for} \quad i = j+1 : \min\{j+L, N-L\}, \quad z_l(i) = z_l(i) - g_{i,j} z_l(j); \quad \text{end} \\ \text{end} \end{cases} \tag{13.52}$$

This step requires $O(NL^2)$ flops.

Step 4: Compute $\mathbf{\Psi} = \mathbf{Z}^H \mathbf{Z}$.

Since $\mathbf{Z}$ is an $(N-L) \times (L+1)$ Hermitian matrix, this step requires $O(NL^2)$ flops.

Step 5: Compute $\mathbf{\Omega} = \mathrm{Re}\left(\mathbf{\Gamma}^H \mathbf{\Psi} \mathbf{\Gamma}\right)$.

This step requires $O(L^3)$ flops.

Step 6: Compute $\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^T \mathbf{\Omega} \boldsymbol{\beta}$ subject to $\| \boldsymbol{\beta} \| = 1$.

Note that $\hat{\boldsymbol{\beta}}$ is the eigenvector of $\mathbf{\Omega}$ corresponding to its smallest eigenvalue. This step requires $O(L^3)$ flops.

In practice, L, the number of signals, is usually much smaller than N, the number of data points, hence Steps 1 through 4 constitute the major computational load of MODE, which is around $O(NL^2)$ flops.

13.4.2 WRELAX

Compared with MODE, WRELAX is computationally much simpler. From (13.31) and (13.43), we note that WRELAX involves a sequence of one-dimensional search over the parameter space. This search can be implemented using the weighted FFT and dedicated high speed FFT chips, such as TMC2310 [24], A41102 [25], and TM-66 *swiFFT* [26]. With brute force search, sufficient zero paddings are needed to guarantee the high accuracy of the estimates. An alternative scheme to zero-padding FFT is to find an approximate peak location first by using FFT without much zero-padding and then perform a fine search nearby

the approximate peak location by, for example, the $fmin$ function in MATLAB, which uses the Golden section search algorithm.

Golden section search is an efficient one-dimensional iterative optimization method exhibiting local convergence property [22]. Each iteration requires the evaluation of the the cost function once. After each iteration, the search interval is shortened by a factor of 0.618. To estimate the amount of computations required by WRELAX, let us assume that the iteration number needed by the Golden section search is N_g. N_g depends on the accuracy desired for the estimates of $\{\omega_L\}_{l=1}^{L}$. Provided that the desired accuracy for $\{\hat{\omega}_L\}_{l=1}^{L}$ is $\Delta\omega_{min}$, then the iteration number N_g needed by the Gloden section search is the minimum integer satisfying the following inequality

$$\frac{2\pi}{\tilde{N}} 0.618^{N_g} \leq \Delta\omega_{min}, \tag{13.53}$$

where $\tilde{N}$ denotes the the number of data points after zero-padding used by the coarse-gridded FFT search and is power of 2. From (13.53), it follows that

$$N_g = \left\lceil \log_{0.618}\left(\frac{\tilde{N}\Delta\omega_{min}}{2\pi}\right)\right\rceil, \tag{13.54}$$

where $\lceil X \rceil$ rounds X to the nearest integer $\geq X$. The amount of computations required by WRELAX for each iteration is $O\left[\left(\tilde{N}\log_2\tilde{N} + 4NN_g\right)L\right]$ flops.

13.5 Numerical Examples

In this section, we present several numerical examples illustrating the performance of MODE-WRELAX. In all of the examples below, we have used $\epsilon = 0.001$ to test the convergence of WRELAX. All data sequences are zero-padded to the nearest power of 2. MODE is applied to a data segment (see **Remark 1** in Section 3.1) satisfying $|S(k)| > \max\{|S(k)|\}/10$. The spacing threshold value $\Delta\omega_t$ (see **Remark 2** in Section 3.1) is chosen as $0.15\tau_e$, where τ_e is the equivalent pulse width and is equal to the reciprocal of the signal bandwidth. (τ_e is usually considered to be the resolution limit of the matched filter approach.) The one-dimensional search is performed in two steps, a coarse search using FFT followed by a fine search using the $fmin$ function of MATLAB. The mean-squared error (MSE) is obtained

through 100 Monte-Carlo trials. For all of the examples below, the MODE amplitude estimates are obtained without the spacing adjustment.

Case A: Application to Ultra Wideband Ground Penetrating Radar

The detection and classification of roadway subsurface anomalies are very important for the design and quality evaluation of highways. Ultra wideband ground penetrating radar emits nonsinusoidal impulses with extremely large bandwidth (several GHz) and is very suitable for this application because of its high range resolution (on the order of several centimeters). The returned echoes of the ultra wideband ground penetrating radar are superimposed real-valued signals reflected from the boundaries of different media (layers, voids, etc.), which can be described by (13.6). Both the delays and gains are very useful for the detection and classification of roadway subsurface anomalies. The delays can be used to determine the layer thickness or anomaly location and the gains can be used to classify the type of media because the gains are related to the reflection coefficient at the boundary between two media with different dielectric constants. Once we get the estimates of the media dielectric constants, we can judge the type of the media.

Although the range resolution of the ultra wideband ground penetrating radar is very high, it is still very difficult, if not impossible, to identify closely spaced echoes from different layers by visual examination or using the matched filter method. Yet the closely spaced echoes may be more important for the detection and classification of the anomalies.

Ultra wideband signals have many unique features. The sampled version of such a signal $s(t)$ is depicted in Figure 13.1(a) as a function of the sample points, where the sampling interval is T_s=0.07 ns. The discrete time Fourier transform (magnitude) of the signal in Figure 13.1(a) is shown in Figure 13.1(b), where $f_s = 1/T_s$=14.28 GHz. From Figure 13.1(b), it can be seen that the signal spectrum covers a wide range (from 0 to 2.5 GHz). Figure 13.1(c) shows the autocorrelation function (magnitude) of the signal $s(t)$ in Figure 13.1(a), from which high sidelobes can be observed. These high sidelobes will greatly degrade the performance of the conventional matched filter approach. The observed signal $y(t)$ is assumed to be composed of signals coming from three layers with delays $\tau_1 = 40T_s = 2.835$ ns, $\tau_2 = 70T_s = 4.961$ ns, and $\tau_3 = 72T_s = 5.102$ ns, and amplitudes $\alpha_1 = 1.0$, $\alpha_2 = 0.4$, and

$\alpha_3 = 0.3$. Noise is also added to the observed signal and the noise variance is 324, as determined by the data collection system. The sampled waveform of $y(t)$ is as shown in Figure 13.1(d). Note that since the signals coming from Layers 2 and 3 are so closely spaced, only two signals can be observed by visual examination.

The performances of MODE (a, b, and c), WRELAX (d, e, and f), and MODE-WRELAX (g, h, and i) are compared in Figure 13.2. In this example, the data length N is 200. In Figure 13.2, the solid lines and the symbols "o" denote the true and estimated echoes of each layer, respectively. From Figure 13.2, we note that the echo due to Layer 1 is well estimated and the estimates are almost the same for all methods since this layer is well separated from the other two layers. However, the estimated echoes due to Layers 2 and 3 differ greatly. MODE can resolve the two closely spaced signals but the estimates are biased, especially for the amplitude estimates. In some other trials, the MODE amplitude estimates are even poorer and are on the order of 10^{13}. WRELAX cannot resolve the two closed spaced signals due to its slow convergence. MODE-WRELAX not only successfully resolves the two closely spaced signals but also provides very accurate estimates for both the delays and the amplitudes. (In this example, the number of iterations required by Steps (2) and (3) of the MODE-WRELAX algorithm for real-valued signals is 24 and 10, respectively.)

Case B: Application to Multipath Underwater Acoustic Signals

We now show the performance of MODE-WRELAX for bandpass real signals with highly oscillatory correlation functions, which may occur in underwater sonar applications. The performances of MODE, WRELAX, and MODE-WRELAX are compared with the Cramér-Rao bound (CRB), which gives the minimum attainable variances for any unbiased estimators. (Detailed derivations of the CRB can be found in [13].)

In this example, we use a windowed chirp signal,

$$s(t) = w(t)\cos\left[2\pi f_0 t + \beta\left(t - \frac{T_0}{2}\right)^2\right], \quad 0 \le t \le T_0, \tag{13.55}$$

where f_0 denotes the carrier frequency, β represents the chirp rate, and

$$w(t) = \begin{cases} 0.5 - 0.5\cos(\pi t/T_w), & 0 < t < T_w, \\ 1, & T_w \leq t \leq T_0 - T_w, \\ 0.5 - 0.5\cos[\pi(t - T_0)/T_w], & T_0 - T_w < t \leq T_0, \end{cases} \tag{13.56}$$

with $T_w = T_0/10$.

Other signal parameters are chosen as $N = 256$, $\beta = \pi \times 10^5$, the signal bandwidth $B_s = \beta T_0/\pi$, and the sampling frequency $f_s = 8B_s$. T_0 is chosen in such a way that $T_0 = (N/2-1)T_s$. In this case, it can be shown that $T_0 = \sqrt{\frac{(N/2-1)\pi}{8\beta}} = 12.6$ ms, $T = 25.3$ ms, $T_s = 99.209$ μs, $B_s = 1.26$ KHz, $f_s = 10.08$ KHz, and the resolution limit of the conventional matched filter method is around $\tau_e = 1/B_s = 0.79368$ ms. The carrier frequency of the transmitted signal is $f_0 = 2B_s$.

Since the cost function for real-valued amplitudes is more oscillatory than the one for complex-valued amplitudes, we use FFT with more zero paddings for WRELAX and MODE-WRELAX. For the former case, the data length after zero padding is $4N$, while for the latter case, it is N. The sampled noise $\{e(nT_s)\}$ is assumed to be a real-valued zero-mean white Gaussian random process with variance σ^2. The SNR for each signal is defined to be $10\log_{10}\left(\frac{\alpha_l^2 \sum_{n=0}^{N-1} |s(nT_s)|^2/N}{\sigma^2}\right)$.

To see the oscillatory nature of the cost function we deal with, consider the case where there is only one direct path with delay $\tau_1 = 0$ and no noise is present. The cost functions used to obtain $\hat{\omega}_1$ for real-valued α_1 (solid line, corresponding to (13.43) and complex-valued α_1 (dashed line, corresponding to (13.31) are compared with each other in Figure 13.3, where the horizontal axis denotes the normalized time delay τ/T. From Figure 13.3, it can be seen that the cost function for complex-valued α_1 is approximately the envelope of that of the real-valued α_1 [10]. The former is very smooth and does not change with f_0, while the latter is highly oscillatory and oscillates more abruptly as f_0 increases. Maximizing the latter can yield much more accurate parameter estimates than maximizing the former due to the sharper dominant peak of the latter.

Now we consider an example where the echoes corresponds to $L = 2$ paths with $\alpha_1 = 1$, $\alpha_2 = 1$. $\tau_1 = T_0/8$, and $\tau_2 = T_0/8 + 0.2\tau_e$. The MSEs of MODE ("o"), WRELAX

("×"), and MODE-WRELAX ("*") are compared with the corresponding CRBs (solid line) in Figure 13.4. Note that due to the highly oscillatory cost functions and very closely spaced signals, WRELAX converges to some local minimum instead of the global one, which yields very poor estimates. Since the MODE amplitude estimates are obtained without spacing adjustment, they are so poor at low SNR that some of their MSEs are above the axis limit due to the inversion of ill-conditioned matrices corresponding to very closely spaced delay estimates. Although the MSEs of the MODE estimates are close to the CRBs corresponding to the complex-valued amplitudes when the SNR is high, the wrong CRBs (not shown to avoid too many lines in the figure) can be larger than the true CRBs, which correspond to the real-valued amplitudes, by approximately 30 dB. (Note that the former CRBs are expected to be worse than the latter CRBs due to the parsimony principle [27].) MODE-WRELAX significantly outperforms MODE and WRELAX and can approach the true CRBs. Note that for the real-valued signals that do not have highly oscillatory cost functions, for example, for the ground penetrating radar probing signal used in Case A (see Figure 13.1 (c)), Step (2) of the MODE-WRELAX approach given in Section 3.2 can be skipped. However, when the signals have highly oscillatory correlation functions, such as the one used in this example, Step (2) is needed to yield the best estimates since the initial estimates provided by MODE is not accurate enough to achieve the global convergence of the last step of WRELAX for such real-valued signals. For this example, the SNR threshold for MODE-WRELAX to approach the CRBs without Step (2) is about 10dB higher than that for MODE-WRELAX with Step (2). (Note that the average numbers of iterations required by Steps (2) and (3) of MODE-WRELAX for real-valued signals are $20 \sim 33$ and $7 \sim 13$, respectively, in this example.)

Case C: Application to High Range Resolution Radar

In the this example, we apply the time delay estimation technique to target feature extraction with high range resolution radar (HRRR). HRRR can be used to form one-dimensional target range signatures (radar cross section (RCS) versus range) with high resolution, which can be used for automatic target recognition. To achieve the high range resolution, the radar must transmit signals with very large bandwidth. For this purpose, pulsed linear frequency

modulation (FM) chirp waveform is commonly used. However, the requirement for the linearity of the wideband frequency modulation is very stringent if FFT is used to obtain target range signatures [28]. We can relax this requirement by applying our super resolution time delay estimation technique to this problem.

The transmitted windowed complex-valued signal has the form

$$s(t) = w(t)e^{j\beta(t-\frac{T_0}{2})^2}, \quad 0 \le t \le T_0, \tag{13.57}$$

where the window function $w(t)$ is defined in (13.56) and the definitions of all other parameters are the same as those used in the previous example.

In this example, we use $N = 128$, $\beta = \pi \times 10^{12}$, the signal bandwidth $B_s = \beta T_0/\pi$, and the sampling frequency $f_s = 4B_s$. T_0 is chosen in such a way that $T_0 = (N/2-1)T_s$. In this case, it can be shown that $T_0 = \sqrt{\frac{(N/2-1)\pi}{4\beta}} = 3.9686$ μs, $T = 8.0002$ μs, $T_s = 0.062994$ μs, $B_s =$ 3.9686 MHz, $f_s = 15.875$ MHz, and the resolution limit of the conventional matched filter method is around $\tau_e = 1/B_s = 0.25198$ μs. For the sake of simplicity, we assume that the target is composed of two scatterers with $\alpha_1 = e^{j\pi/8}$, $\alpha_2 = e^{j\pi/4}$, $\tau_1 = T_0/128$, and $\tau_2 = T_0/128 + 0.2\tau_e$. Zero-mean complex white Gaussian random noise is added and the definition of SNR is the same as used in Case B. We use FFT with no zero-padding in WRELAX and MODE-WRELAX.

The MSEs of MODE ("o"), WRELAX ("×"), and MODE-WRELAX ("*") are compared with the corresponding CRBs (solid line) in Figure 13.5. The performance of WRELAX is poor due to its slow convergence for very closely spaced signals. Both MODE and MODE-WRELAX can approach the CRBs when the SNR is high. However, MODE-WRELAX outperforms MODE significantly at low SNR. (Note that the average number of iterations required by Step (2) of MODE-WRELAX for complex-valued signals is 20 $\sim$ 33 in this example.)

13.6 Conclusions

We have studied time delay estimation of very closely spaced signals and presented a super resolution method (referred to as MODE-WRELAX). The popular direction estimation technique MODE is modified and used in combination with our efficient WRELAX algorithm for time delay estimation. MODE-WRELAX outperforms MODE in estimation accuracy and provides better resolution than WRELAX. MODE-WRELAX can be used with not only complex-valued signals but also real-valued signals, including those with highly oscillatory correlation functions. Numerical examples have shown that MODE-WRELAX can approach the corresponding CRBs as the SNR increases.

Reference

[1] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. AP-34, pp. 276–280, March 1986.

[2] D. W. Tufts, R. Kumaresan, and I. Kirsteins, "Data adaptive signal estimation by singular value decomposition of a data matrix," *Proceedings of the IEEE*, vol. 70, pp. 684–685, June 1982.

[3] A. M. Bruckstein, T. J. Shan, and T. Kailath, "The resolution of overlapping echoes," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-33, pp. 1357–1367, December 1985.

[4] Y. Bian and D. Last, "Eigen-decomposition techniques for Loran-C skywave estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, pp. 117–124, January 1997.

[5] I. P. Kirsteins, "High resolution time delay estimation," *Proceedings of ICASSP 87*, pp. 451–454, April 1987.

[6] I. P. Kirsteins and A. Quazi, "Exact maximum likelihood time delay estimation for deterministic signals," *Proceedings of EURASIP 1988*, pp. 531–534, 1988.

[7] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, pp. 47–60, November 1996.

[8] M. Feder and E. Weinstein, "Parameter estimation of superimposed signals using the EM algorithm," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 36, pp. 477–489, April 1988.

[9] B. M. Bell and T. E. Ewart, "Separating multipaths by global optimization of multidimensional matched filter," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. ASSP-34, pp. 1029–1037, October 1986.

[10] T. G. Manickam, R. J. Vaccaro, and D. W. Tufts, "A least-squares algorithm for multipath time-delay estimation," *IEEE Transactions on Signal Processing*, vol. 42, pp. 3229–3233, November 1994.

[11] A. D. Blackowiak and S. D. Rajan, "Multipath arrival estimates using simulated annealing: Application to crosshole tomography experiment," *IEEE Journal of Oceanic Engineering*, vol. 20, pp. 157–165, July 1995.

[12] J. Li and R. Wu, "An efficient algorithm for time delay estimation," *IEEE Transactions on Signal Processing*, vol. 46, no. 8, pp. 2231–2235, August 1998.

[13] R. Wu and J. Li, "Time delay estimation via optimizing highly oscillatory cost functions," *IEEE Journal of Oceanic Engineering*, vol. 23, no. 3, pp. 235–244, July 1998.

[14] P. Stoica and K. C. Sharman, "Novel eigenanalysis method for direction estimation," *IEE Proceedings, Pt. F*, vol. 137, pp. 19–26, February 1990.

[15] P. Stoica and K. C. Sharman, "Maximum likelihood methods for direction-of-arrival estimation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-38, pp. 1132–1143, July 1990.

[16] L. E. Pellon, "A double Nyquist digital product detector for quadrature sampling," *IEEE Transactions on Signal Processing*, vol. 40, no. 7, pp. 1670–1681, July 1992.

[17] R. B. MacLeod, "Modeling of active reverberation by time delay estimation," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2465–2468, Seatle, Washington, May 1998.

[18] R. Roy, A. Paulraj, and T. Kailath, "ESPRIT – A subspace rotation approach to estimation of parameters of cisoids in noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, pp. 1340–1342, October 1986.

[19] S. M. Kay, *Modern Spectral Estimation: Theory and Application.* Englewood Cliffs, N.J.: Prentice-Hall, 1988.

[20] J. Li and P. Stoica, "Efficient mixed-spectrum estimation with applications to target feature extraction," *IEEE Transactions on Signal Processing*, vol. 44, pp. 281–295, February 1996.

[21] G. W. Stewart, *Introduction to Matrix Computations.* New York, NY: Academic Press, Inc., 1973.

[22] W. I. Zangwill, *Nonlinear Programming: A Unified Approach.* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1967.

[23] G. H. Golub and C. F. V. Loan, *Matrix Computations.* Baltimore, MD: Johns Hopkins University Press, 1989.

[24] *TMC2310*, TRW LSI Products, Inc.

[25] *A Users' Guide for the A41102*, Austeck Microsystems, 1988.

[26] *TM-66 swiFFT chip*, Texas Memory Systems, Inc., 1998.

[27] T. Söderström and P. Stoica, *System Identification.* London, U.K.: Prentice-Hall International, 1989.

[28] C. V. Jakowatz, Jr., D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. A. Thompson, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach.* Norwell, MA: Kluwer Academic Publishers, 1996.

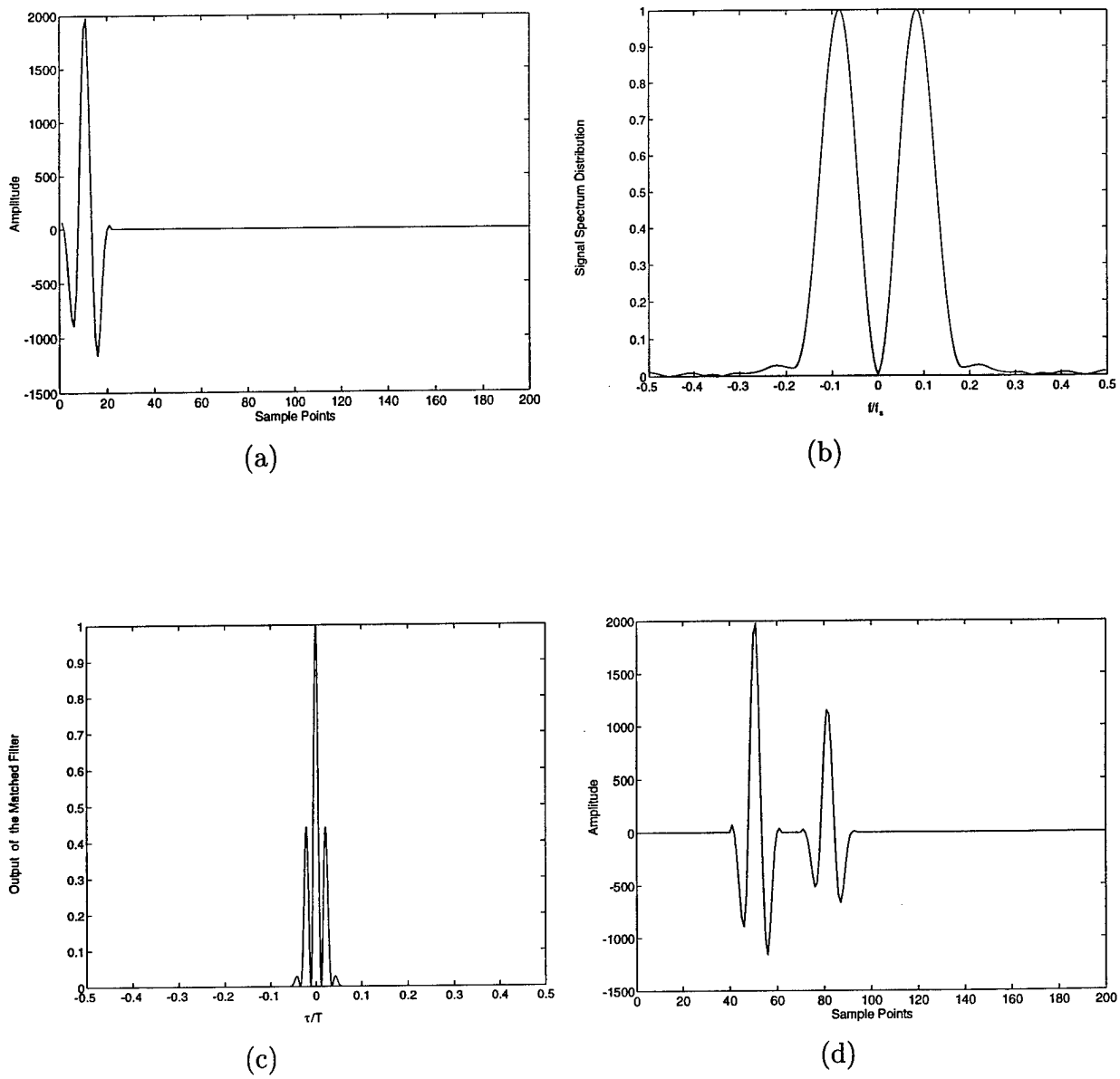


Figure 13.1: Signals used by an ultra-wideband ground penetrating radar. (a) Waveform of the known signal $s(nT_s)$. (b) Discrete Fourier spectrum (magnitude) of $s(nT_s)$. (c) Matched filter output of the known signal $s(nT_s)$. (d) Observed signal $y(nT_s)$ consisting of three reflections from three layers plus noise.

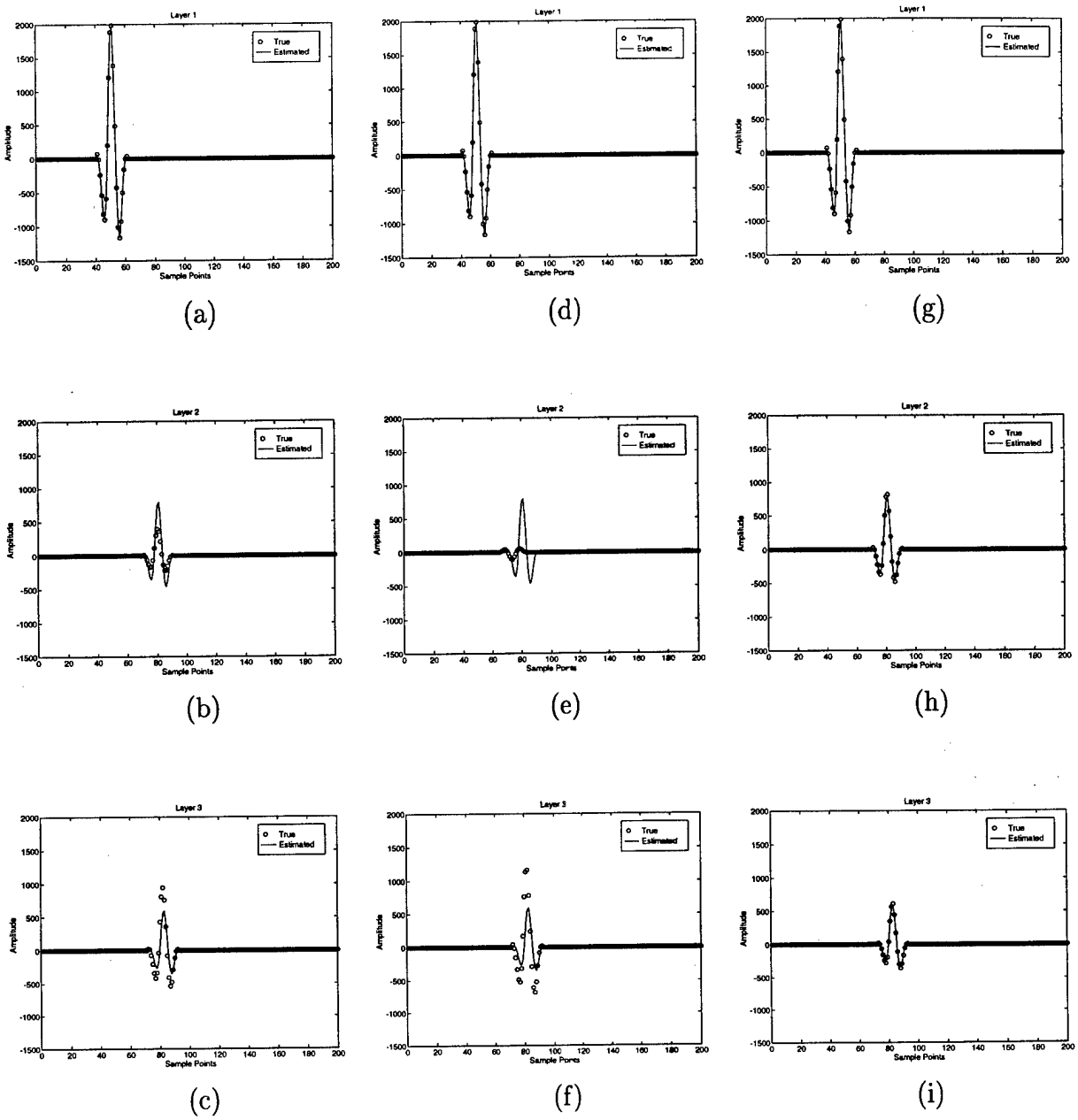


Figure 13.2: Comparison of the true (solid line) and the estimated ("o") echoes of each layer by using MODE (a, b, c), WRELAX (d, e, f) and MODE-WRELAX (g, h, i).

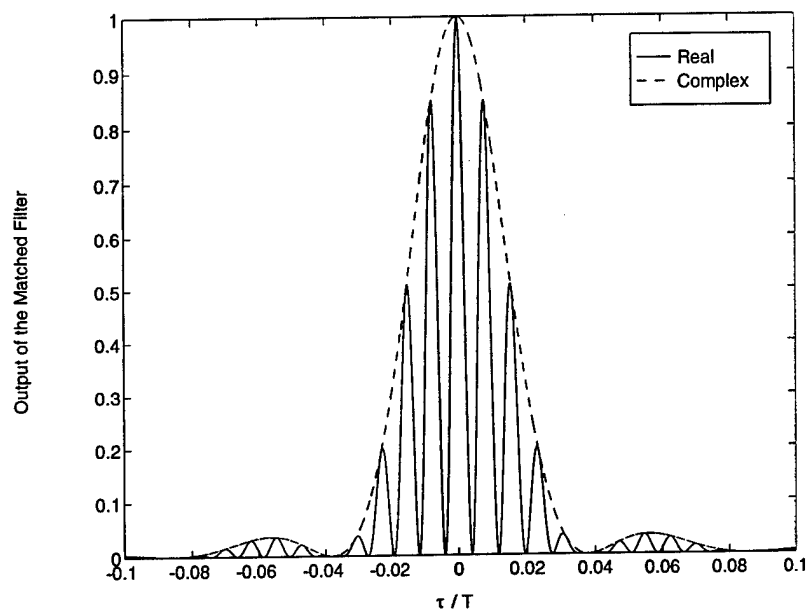


Figure 13.3: Comparison of the cost functions for complex-valued (dashed line) and real-valued signal amplitudes (solid line) .

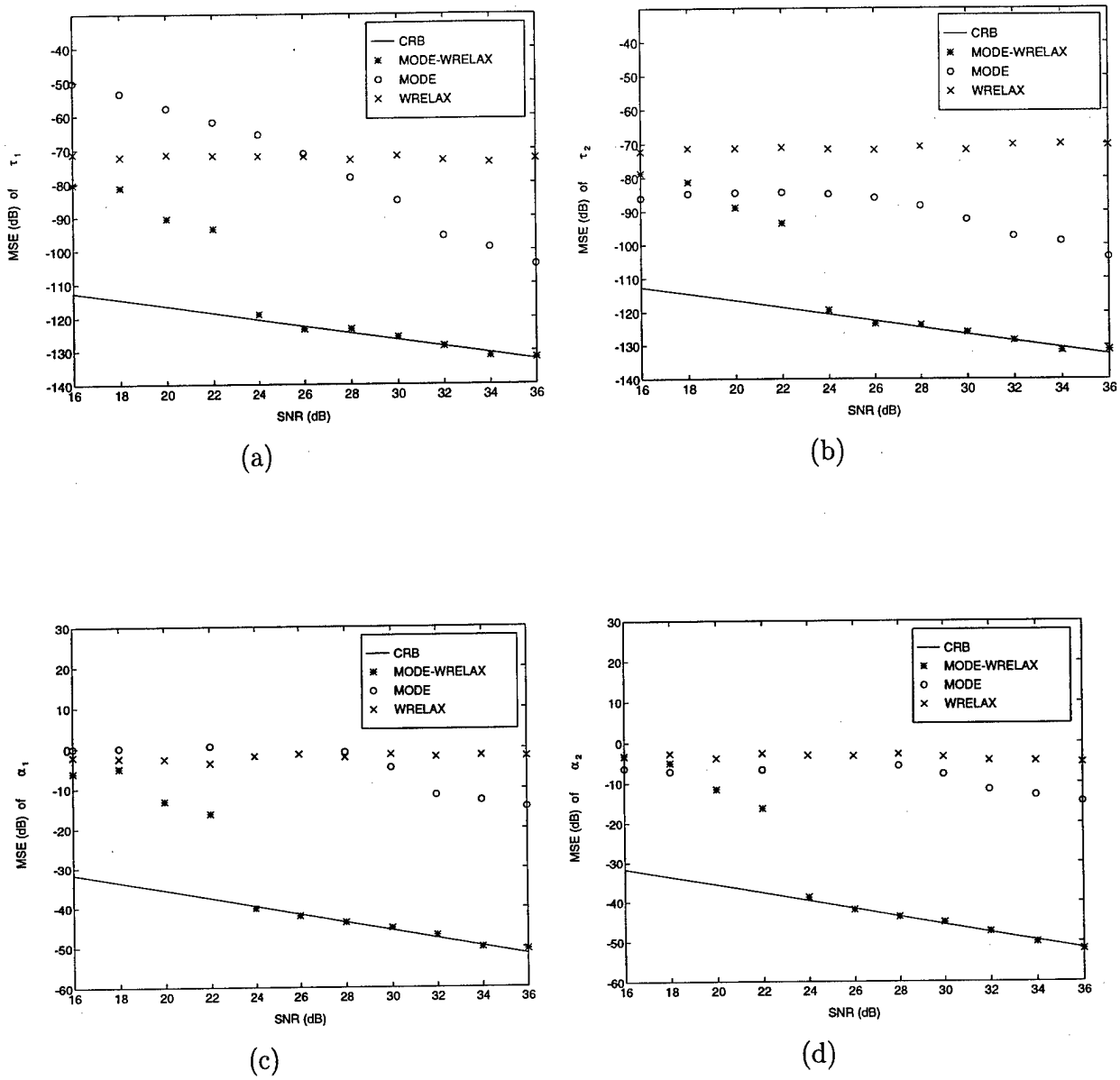


Figure 13.4: Comparison of the MSEs of WRELAX ("×"), MODE("o"), and MODE-WRELAX ("*") with the CRBs (solid line) corresponding to real-valued signals for (a) τ_1, (b) τ_2, (c) α_1, and (d) α_2.

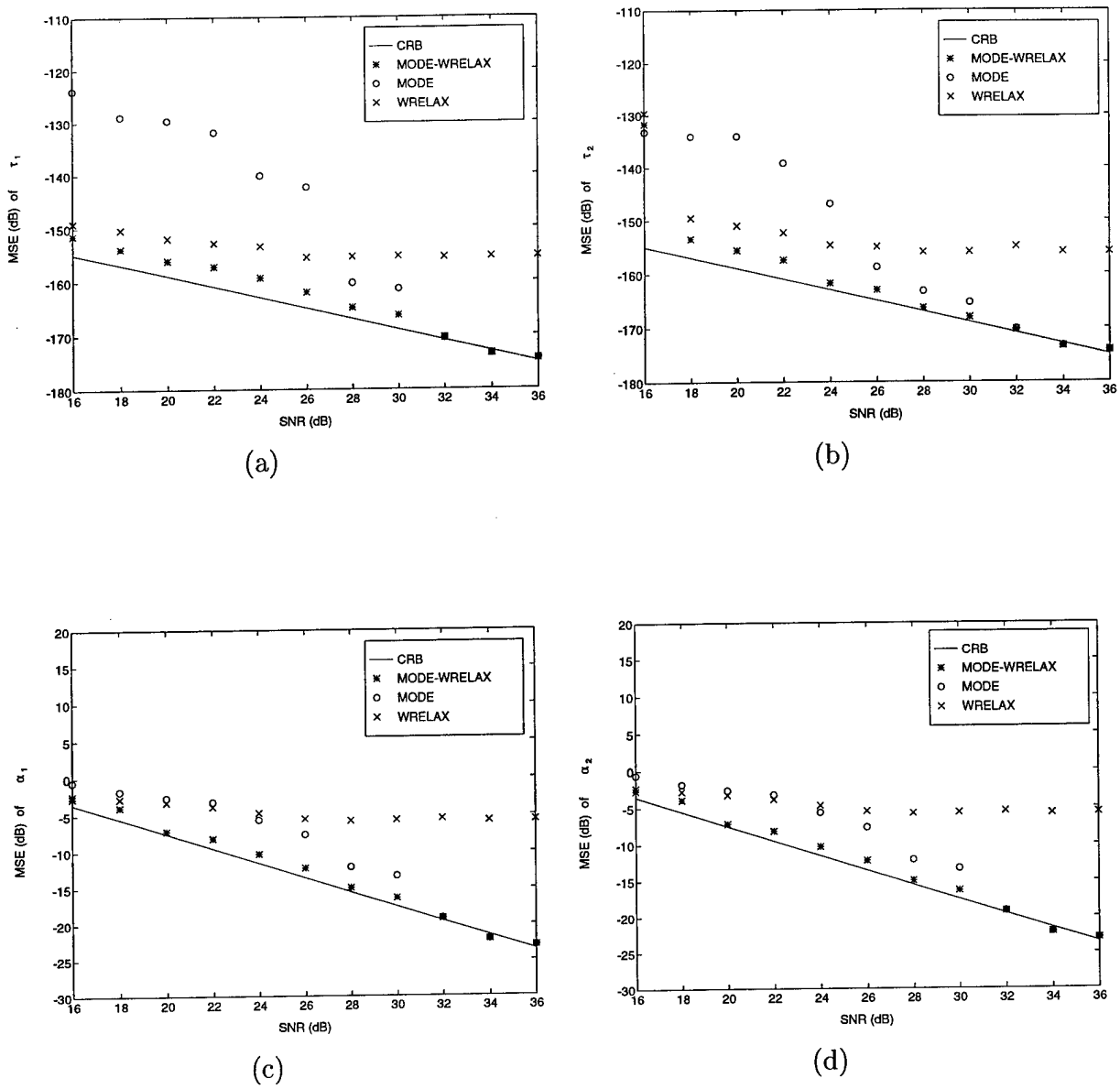


Figure 13.5: Comparison of the MSEs of WRELAX ("×"), MODE("o"), and MODE-WRELAX ("*") with the CRBs corresponding to complex-valued signals (solid line) for (a) τ_1, (b) τ_2, (c) α_1, and (d) α_2.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate of any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (*Leave Blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	6/15/99	Final, 1996-1999

4. TITLE AND SUBTITLE
Radar Imaging and Feature Extraction

5. FUNDING NUMBERS
G
N00014-96-1-0817

6. AUTHOR(S)
Dr. Jian Li

7. PERFORMING ORGANIZATION NAME(S) AND ADDESS(ES)
University of Florida
Dept. of Electrical and Computer Engineering
437 EB, P.O. Box 116130, Gainesville, FL 32611

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONORING / MORNITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
Ballston Centre Tower One
800 North Quincy Street
Arlington, VA 22217-5660

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTRAY NOTES
The work in this report has been written up for publication in several journals and conference proceedings.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

Advanced spectral estimation methods are presented for radar imaging and target feature extraction. We study problems involved in inverse synthetic aperture radar (ISAR) autofocus and imaging, synthetic aperture radar (SAR) autofocus and motion compensation, superresolution SAR image formation, three-dimensional (3-D) target feature extraction via curvilinear SAR (CLSAR), and time delay estimation. For the ISAR imaging problem, we present a parametric AUTOCLEAN (AUTOfocus via CLEAN) algorithm, and two non-parametric algorithms including an adaptive Capon and a recursive APES (Amplitude and Phase EStimation). For the problems related to SAR imaging, we propose a Semi-PARametric (SPAR) algorithm for target feature extraction and superresolution image formation, and two parametric methods, MCRELAX (Motion Compensation RELAX) and MCCLEAN (Motion Compensation CLEAN), for simultaneous target feature extraction and cross-range phase error compensation. For the 3-D target feature extraction problem, an AUTOfocus algorithm based on the RELAXation-based optimization approach (AUTORELAX) is proposed to compensate the aperture errors in CLSAR and to extract 3-D target features. For the time delay estimation problem, we first present a Weighted Fourier transform and RELAXation-based (WRELAX) approach. Then, a MODE-WRELAX (MODE (Method Of Direction Estimation) together with WRELAX) algorithm is proposed for the superresolution time delay estimation.

14. SUBJECT TERMS
SAR Autofocus; SAR Image Formation; ISAR Autofocus; Feature Extraction; Super Resolution

15. NUMBER OF PAGES
322

16. PRICE CODE

17.SECURITY CLASSIFICATION OF REPORT	18.SECURITY CLASSIFICATION OF THIS PAGE	19.SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unclassified	unclassified	unclassified	UL