State-Space Approaches to Ultra-Wideband Doppler Processing

by

David J. Holl, Jr.

A Dissertation
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Doctor of Philosophy
in
Electrical and Computer Engineering
by

____

May 2007

APPROVED:

____

Professor David Cyganski, Major Advisor

____

Professor Kevin Clements

____

Professor Homer Walker

Abstract

National security needs dictate the development of new radar systems capable of identifying and tracking exoatmospheric threats to aid our defense. These new radar systems feature reduced noise floors, electronic beam steering, and ultra-wide bandwidths, all of which facilitate threat discrimination. However, in order to identify missile attributes such as RF reflectivity, distance, and velocity, many existing processing algorithms rely upon narrow bandwidth assumptions that break down with increased signal bandwidth.

We present a fresh investigation into these algorithms for removing bandwidth limitations and propose novel state-space and direct-data factoring formulations such as

- the multidimensional extension to the Eigensystem Realization Algorithm,
- employing state-space models in place of interpolation to obtain a form which admits a separation and isolation of solution components,
- and side-stepping the joint diagonalization of state transition matrices, which commonly plagues methods like multidimensional ESPRIT.

We then benchmark our approaches and relate the outcomes to the Cramer-Rao bound for the case of one and two adjacent reflectors to validate their conceptual design and identify those techniques that compare favorably to or improve upon existing practices.

Acknowledgements

First I thank my advisor, Professor David Cyganski, for sharing his creative insights in physics and electrical engineering, how to identify opportunities in any problem, and for the six years I spent in his lab with access to his diverse array of research projects. I also thank my dissertation committee, Professors Kevin Clements and Homer Walker, for their exceptional instruction and expertise, and I thank MIT Lincoln Laboratory's Group 34 for the financial support and internship opportunities which they have invested in my research.

Most importantly, I am grateful for the patience, love and support of my wife, Nita Madhav; I wish her success in her upcoming novel. And to my mom and dad, I thank them for the world perspective they've imbued upon me, and to my older sister, Sally, and younger brother, Donald, for raising me.

Additionally, graduate school would not have been as much fun without the intellectual curiosity and boundless adventures of my roommates: Shashank Kulkarni, Slobodan Pajic and Charles Wilcox, and my labmates: Vincent Amendolare, Ryan Angilly, Christopher Boumenot, Daniel Breen, Jason Farmer, Nicholas Hatch, Jason Huang, Antti Koski, Pavan Reddy, Nicholas Sherwood, Vivek Varshney, and Benjamin Woodacre.

Contents

List of Figures **vi**

List of Tables **ix**

1 Introduction **1**

2 Radar Signal Processing Background **4**

2.1 Target Impulse Response 4

2.2 Fourier Processing 7

2.3 State-Space Theory 9

2.3.1 State-Space Modeling of Frequency Domain Impulse Response . . . 10

2.3.2 Direct State-Space System Identification 11

2.3.3 Example Output 20

3 Joint-Doppler Processing **26**

3.1 Separating Range and Range-Rate 27

3.2 Fourier Processing 29

3.3 Joint-Doppler Direct State-Space System Identification 33

3.3.1 Multidimensional DSS Observability (or Controllability) Processing 34

3.3.2 Multidimensional DSS Balanced Processing 39

3.3.3 Example Output 42

3.3.4 An Alternative to Interpolation 47

3.4 Linear Fast-Time Invariant (LFTI) DSS Systems 50

3.4.1 Algebraic Formulation 50

3.4.2 Example Output 54

4 Method Validation **56**

4.1 One Scatterer 58

4.2 Interfering Scatterers - Range Sweep 63

4.3 Interfering Scatterers - Range-Rate Sweep 64

4.4 Discussion 69

5 Conclusions **75**

A State-Space Similarity Transformations **78**

B Cramer-Rao Bound for Range and Range-Rate Estimation **80**

Bibliography **86**

List of Figures

2.1 A simplified radar scenario consisting of a target having 3 prominent scattering centers identified as $s \in \{1, 2, 3\}$. 5
2.2 An inverse DFT applied to a frequency domain impulse response yields a map of reflection strength versus time delay or range. 8
2.3 Using color to portray reflection coefficient and stacking multiple range-intensity plots collected over time produces a plot of range versus time versus intensity, which gives insight to target scene dynamics. 9
2.4 A systems-level block diagram expressing the target's impulse function as a linear operation on some input signal to produce an output. 10
2.5 Using color to portray reflection coefficient and stacking multiple range-intensity plots collected over time produces a plot of range versus time versus intensity, which gives insight to target scene motion dynamics. 21
2.6 LL-Cone. 22
2.7 Fourier vs. State-Space, 2.75-3.25 GHz, 25 dB SNR. This figure is reprinted with permission from [24]. 23
2.8 A plot of range versus aspect angle of LL-Cone's scattering features processed over 8-12 GHz. The individual scatterers are color coded by their frequency dependence instead of reflection intensity to aid in identification. This figure is reprinted with permission from [24]. 25

3.1 At 40% bandwidth, this radar data needs significant interpolation to separate r_s and $\dot{r}_s$. 27
3.2 At 10% bandwidth, this radar data may be small enough to neglect resampling without causing too much error in r_s and $\dot{r}_s$ estimation. 28
3.3 Two dimensional Fourier transform in k-space exchanges (k_x, k_y) with (x, y). In this example, the two-dimensional wavefront on the left undergoes 3 full periods on the k_x axis and 2 periods on the k_y axis, so its transform yields Dirac's delta function at $x = 3$, $y = 2$. 30
3.4 Fourier Doppler processing analyzes a block of interrogations at a time. Within a block, data is first resampled (indicated with blue circles) before further processing to analytically separate r_s and $\dot{r}_s$. This technique is also used for modern spectral processing algorithms discussed in the next section. 30

3.5 These are frames taken from an Range-Doppler-Time-Intensity animation of our synthetic data in Chapter 2. Each frame is produced from 10 interrogations by first applying Hamming windows across the frequency and time dimensions, then computing the two-dimensional inverse DFT, and plotting the resulting magnitudes. 31
3.6 The upper image is the same RTI plot from Fig. 2.3. The lower image is called a *Doppler vs. Time vs. Intensity* (DTI) plot, and shows the range-rates of a target scene's scatterers as they change over time due to acceleration and deceleration. 32
3.7 These plots of range and range-rate versus time were generated from two-dimensional state-space observability factorization of the synthetic data described in Chapter 2. The 20% bandwidth data was resampled with a spline interpolation as described Section 3.1. 43
3.8 These plots of range and range-rate versus time were generated similarly to those of Fig 3.7, except that cubic interpolation was used instead of spline. Note the severe degradation of the scatterer having -6 dB reflection intensity. 44
3.9 These plots of range and range-rate versus time were generated similarly to those of Fig 3.7, except that state-space balanced factorization was used instead of observability factorization. Note how some of the artifacts in Fig 3.7 are suppressed. 45
3.10 These range and range-rate versus time plots were generated without interpolating as is the assumption of narrow bandwidth algorithms. The 20% bandwidth data was purposefully not resampled to show an example of the corruption that occurs when narrow bandwidth algorithms are glibly applied to wide bandwidth data. 46
3.11 These range and range-rate plots versus time were generated using observability factorizations and *stretch processing* instead of spline or cubic interpolation. The bottom-most plot highlights the deviations that occur when the scatterers' range-rates were in close proximity. 49
3.12 These range and range-rate plots versus time were generated using observability factorizations and *narrow-subband stretch processing* instead of interpolation. Compared to Fig. 3.11, the bottom-most plot here shows how the scatterers' range-rates no longer exhibit deviations when they are in close proximity. 51
3.13 These range and range-rate plots versus time were generated using LFTI state-space processing. 55

4.1 Raw error data from a 1000-trial test of the Observability method with stretch processing. The scatterers identified with red X's were classified as outliers. 57
4.2 Comparison of RMS error of estimating a scatterer's reflection magnitude versus SNR for different algorithms. 59
4.3 Comparison of RMS error of estimating a scatterer's reflection phase offset versus SNR for different algorithms. 60
4.4 Comparison of RMS error of estimating a scatterer's range versus SNR for different algorithms. 61

4.5 Comparison of RMS error of estimating a scatterer's range-rate versus SNR for different algorithms. 62
4.6 Comparison of RMS error of estimating the reference scatterer's reflection magnitude versus the range separation of the interfering scatterer for different algorithms. The lower plot is a magnified view of the upper plot. 65
4.7 Comparison of RMS error of estimating the reference scatterer's reflection phase offset versus the range separation of the interfering scatterer for different algorithms. 66
4.8 Comparison of RMS error of estimating the reference scatterer's range versus the range separation of the interfering scatterer for different algorithms. . . 67
4.9 Comparison of RMS error of estimating the reference scatterer's range-rate versus the range separation of the interfering scatterer for different algorithms. 68
4.10 Comparison of RMS error of estimating the reference scatterer's reflection magnitude versus the range-rate separation of the interfering scatterer for different algorithms. 70
4.11 Comparison of RMS error of estimating the reference scatterer's reflection phase offset versus the range-rate separation of the interfering scatterer for different algorithms. 71
4.12 Comparison of RMS error of estimating the reference scatterer's range versus the range-rate separation of the interfering scatterer for different algorithms. 72
4.13 Comparison of RMS error of estimating the reference scatterer's range-rate versus the range-rate separation of the interfering scatterer for different algorithms. 73

B.1 Cramer-Rao lower bounds on estimation variance for identifying a scatterer's attributes: reflection magnitude, reflection phase offset, range, range-rate, and range-acceleration. These bounds are plotted for radar data having the same center frequency, bandwidth, interrogation rate, and window as used in the validation tests of Chapter 4. Though the scatterer attributes have differing units, we present them without units on the same graph for visual comparison. 84

List of Tables

1.1 Common abbreviations 2
1.2 Math notation 3

2.1 GTD-predicted α for various scattering features. 24

Chapter 1

Introduction

In this research, we set out to investigate the *Joint-Doppler processing* of *ultra-wideband* signals. Here, ultra-wideband refers to any signal whose ratio of bandwidth over center frequency exceeds approximately 40%, and we use Joint-Doppler processing to refer to a whole class of problems such as simultaneously estimating an unknown radio transmitter's carrier frequency and direction of arrival using an antenna array, or using a single [radar] antenna to estimate the range and velocity of an unknown ballistic threat. As an example to highlight the general applicability of this research, we would also use Joint-Doppler processing to describe algorithms that would be employed by advanced radar systems using a two-dimensional antenna array to identify the range, velocity, azimuth, and elevation of a number of uncooperative targets.

Algorithms already exist for Joint-Doppler estimation, and we may broadly categorize their solution strategies as either polynomial realizations or state-space parameterizations. Polynomial realizations such as MUSIC DOA Triangulation, G-MUSIC, and Mutual Information (summarized and evaluated in [18]) may be used to process any amount of signal bandwidth and sensor configuration, but consume exorbitant processing resources especially when the particular method must search some error surface or manifold in a global optimization strategy. In contrast, state-space parameterizations such as those described in [26, 50, 35, 8, 45] employ data interpolation or narrow bandwidth assumptions in order to obtain a form which admits a separation of solution components and isolate these components via matrix factoring techniques. The state-space approaches produce more robust model parameterizations, especially those methods based on the direct singular value decomposition (SVD) of the input data [47]. Therefore, the present research investigates novel *direct state-space* formulations with emphasis on processing large signal bandwidths.

We have organized this dissertation as follows: Chapter 2 presents a brief background in the context of radar signal processing and super-resolution direct state-space approaches. We discuss the challenges of wide bandwidth Doppler processing and propose new algorithmic strategies in Chapter 3. Then in Chapter 4, we compare the performance of the new and existing strategies and the theoretical limits. Lastly, we summarize our findings and provide conclusions in Chapter 5.

Additionally, this dissertation uses many abbreviations and algebraic notations, and defines many of them only once. For the reader's convenience, Tables 1.1 and 1.2 list these abbreviations and notations, respectively.

CRB	Cramer-Rao Bound
DFT	Discrete Fourier Transform
DSS	Direct State-Space
EM	Electromagnetic
FFT	Fast Fourier Transform
GTD	Geometric Theory of Diffraction
LFTI	Linear Fast-Time Invariant
LTI	Linear Time Invariant
PRF	Pulse Repetition Frequency
RCS	Radar Cross-Section
RMS	Root-Mean-Square
RTI	Range vs. Time vs. Intensity
SAR	Synthetic Aperture Radar
SVD	Singular Value Decomposition
UWB	Ultra-Wide Bandwidth Systems

Table 1.1: Common abbreviations used throughout this dissertation.

a	Scalar
$\vec{a}$	Vector
A	Matrix
$\mathrm{A} \in \mathbb{C}^{m \times n}$	A is a complex matrix having m rows and n columns.
$\mathrm{A} \in \mathbb{R}^{m \times n}$	A is a real matrix having m rows and n columns.
A^*	Conjugate the elements of A
A^T	Transpose of A
A^H	Hermitian Transpose of A, defined as $\left(\mathrm{A}^T\right)^*$ or $\left(\mathrm{A}^*\right)^T$
$\mathrm{A}^\dagger$	Moore-Penrose pseudo-inverse of A [21, p. 257]
I	Identity matrix
I_m	Identity matrix having m rows and m columns
$a(t)$	Continuous function of t in time domain
$\hat{a}(f)$	Continuous function of f in frequency domain
a_k	Discrete function of k in time domain
$\hat{a}_n$	Discrete function of n in frequency domain
$\mathrm{A} \triangleq \ldots$	Define A as ...
$\mathrm{A} \Leftarrow \ldots$	Assign A equal to result of ...
$\mathrm{A} \overset{\text{svd}}{\Rightarrow} \mathrm{U}\Sigma\mathrm{V}^H$	Decompose A using the singular value decomposition into $\mathrm{U}\Sigma\mathrm{V}^H$
$\mathrm{A} \overset{\text{eig}}{\Rightarrow} \mathrm{V}\Lambda\mathrm{V}^{-1}$	Decompose A using the eigen decomposition into $\mathrm{V}\Lambda\mathrm{V}^{-1}$
$\ln(a)$	Natural logarithm of a
$\operatorname{diag}\mathrm{A}$	Retrieve the diagonal elements of A
$\lVert\mathrm{A}\rVert_F$	Frobenius norm of A
$\mathcal{F}$	Fourier transform operator
$\mathcal{H}\,\vec{a}$	Arrange the elements of $\vec{a}$ into a Hankel matrix.
$a(t) * b(t)$	Convolution of $a(t)$ with $b(t)$ is defined as $\int_0^t a(\tau)\, b(t-\tau)\, \mathrm{d}\tau$.

Table 1.2: Common math notation used throughout this dissertation.

Chapter 2

Radar Signal Processing Background

While the algorithms we discuss in the next chapter deal directly with Doppler processing, we will use this chapter to derive a simplified signal model to serve as background and establish a common vocabulary. We establish this model in the context of radar signal processing and later show the generalization to other applications via a simple substitution of variables.

2.1 Target Impulse Response

We center our derivation upon a simple arrangement of a radar system observing some ballistic target, depicted in Fig. 2.1. To detect the range to the target, our ideal radar measures the delay time, τ, for an electromagnetic (EM) impulse to reflect off the target and return to the radar antenna. We assume the target is sufficiently far from the antenna such that the EM wavefront is planar and perpendicular to the direction of propagation.

Keller's *Geometric Theory of Diffraction* (GTD) [32] states that the back scattered field from a complex surface tends to be well approximated by a set of discrete scattering centers, so in Fig. 2.1, we depict the target as having 3 *scatterers*, each having independent strengths, Γ_s, and EM propagation delays, τ_s. To describe the scenario analytically, we express the

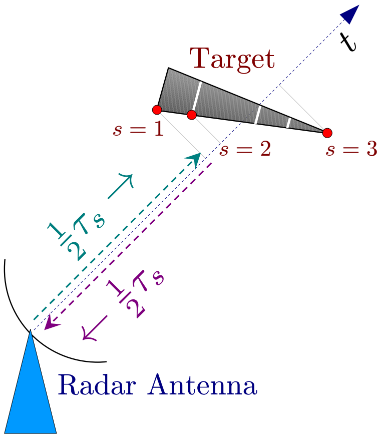

Target
t
$s=1$
$s=2$
$s=3$
$\frac{1}{2}\tau_s \rightarrow$
$\leftarrow \frac{1}{2}\tau_s$
Radar Antenna

Figure 2.1: A simplified radar scenario consisting of a target having 3 prominent scattering centers identified as $s \in \{1, 2, 3\}$.

transmitted pulse, v_{tx}, using Dirac's delta function in Eq. 2.1, and the received signal, v_{rx}, as the sum of received reflections from the target, Eq. 2.2.

$$v_{\text{tx}}(t) = \delta(t) \tag{2.1}$$

$$v_{\text{rx}}(t) = \sum_s \Gamma_s \, \delta(t - \tau_s) \tag{2.2}$$

To determine the range to the scattering centers (r_s), we divide the round-trip delay in half and scale by the speed of light:

$$r_s = \frac{\tau_s}{2} c \tag{2.3}$$

Or alternatively, we shall substitute variables to express the original received signal in terms of range to the target's scatterers using a substitution of variables, Eq. 2.5.

$$\tau_s = \frac{2}{c} r_s \tag{2.4}$$

$$v_{\text{rx}}(t) = \sum_s \Gamma_s \, \delta\big(t - \tfrac{2}{c} r_s\big) \tag{2.5}$$

Given that we assumed the transmitted signal was a theoretical impulse function, Eq. 2.5 is also known as the target's *impulse response*, and instead of using v_{rx}, we use g to denote this special function:

$$g(t) = \sum_s \Gamma_s \, \delta\big(t - \tfrac{2}{c} r_s\big) \tag{2.6}$$

For the case of an arbitrarily complex transmit waveform, we may determine the received waveform that results by convolving the transmitted signal with target's impulse response:

$$v_{\mathrm{rx}}(t) = g(t) * v_{\mathrm{tx}}(t) \tag{2.7}$$

However to simplify matters, we will side-step the convolution by switching the remainder of our derivation and analysis to the frequency domain via the Fourier transform, $\mathcal{F}$, and the convolution theorem [54, p. 169]. By Fourier-transforming $g(t)$ into $\hat{g}(f)$ and $v_{\mathrm{tx}}(t)$ into $\hat{v}_{\mathrm{tx}}(f)$, the frequency domain received signal function may be obtained from the product of $\hat{g}(f)$ and $\hat{v}_{\mathrm{tx}}(f)$ (Eq. 2.12). We denote the frequency domain representation of a function or variable with a small $\wedge$ instead of capitalization, because we reserve capitalization for matrices.

$$\hat{g}(f) \triangleq \mathcal{F}\, g(t) \tag{2.8}$$

$$= \sum_s \Gamma_s e^{-2\pi j f \frac{2}{c} r_s} \tag{2.9}$$

$$\hat{v}_{\mathrm{tx}}(f) \triangleq \mathcal{F}\, v_{\mathrm{tx}}(t) \tag{2.10}$$

$$\hat{v}_{\mathrm{rx}}(f) \triangleq \mathcal{F}\, v_{\mathrm{rx}}(t) \tag{2.11}$$

$$\hat{v}_{\mathrm{rx}}(f) = \hat{g}(f) \cdot \hat{v}_{\mathrm{tx}}(f) \tag{2.12}$$

By expressing everything in the frequency domain, we have the ability to trivially identify the impulse response of an unknown target by dividing the transmitted signal by the received signal:

$$\hat{g}(f) = \frac{\hat{v}_{\mathrm{rx}}(f)}{\hat{v}_{\mathrm{tx}}(f)} \tag{2.13}$$

Radar systems use many different transmit waveforms, but when processing the reflected data, most use some method of matched filtering such as that of Eq. 2.13 to extract the impulse response of the unknown target. Therefore, processing algorithms may focus on handling the general frequency-domain impulse response described by Eq. 2.9.

The last step we perform in establishing our example radar processing model is to put Eq. 2.9 into the form that a digital system would use. Up to this point when considering time delays (t or τ) or frequencies (f), we assumed these quantities were continuous in nature. However in the course of digital signal processing, these quantities are sampled at known intervals. In the case of Eq. 2.9, we replace f with f_n to represent the fact that the system only considers some discrete frequency set where the subscript n is used to denote a particular frequency of the set. Eqs. 2.14 and 2.15, are two possible schemes for analytically describing this set of frequencies spaced Δf apart.

$$f_n = n\Delta f + f_0 \quad \text{where } n \in \{0, 1, 2, \ldots\} \tag{2.14}$$

$$f_n = n\Delta f \quad \text{where } n \in \{n_0, n_0 + 1, n_0 + 2, \ldots\} \tag{2.15}$$

Both are equivalent if $f_0 = n_0\Delta f$; Eq. 2.14 is more commonly used in other literature where Δf refers to the frequency sampling interval, and f_0 is the first frequency of the set. However, we use Eq. 2.15 which omits f_0 in favor of starting n at n_0 instead of 0. Eliminating f_0 in exchange for introducing n_0 turns out to simplify our analytic expressions. Substituting Eq. 2.15 into Eq. 2.9 yields Eq. 2.16.

$$\hat{g}_n = \sum_s \Gamma_s e^{-4\pi j \frac{n\Delta f}{c} r_s} \tag{2.16}$$

2.2 Fourier Processing

Referring back to Fig. 2.1, an *interrogation* refers to the process of a radar sending an EM pulse toward an unknown *target scene* and recording any reflections. This target scene often consists of multiple targets, and because they are constantly moving, the radar system interrogates the scene many times per second, and for each, the radar computes the impulse response of the whole scene and passes that frequency-domain data to our algorithms. The rate at which a radar interrogates a target scene is called the *pulse repetition frequency* (PRF), and the system may vary this rate during the course of operation to satisfy mission requirements.

We represent the complex data from one interrogation as a column vector whose dimension corresponds to the number of measured frequency-domain sample points. To process this data and determine the composition of the target scene, the most traditional approach is to apply the inverse Discrete Fourier Transform (DFT) to obtain a new vector representing reflection strength versus time-delay between the radar and target(s) [39, p. 47–48].

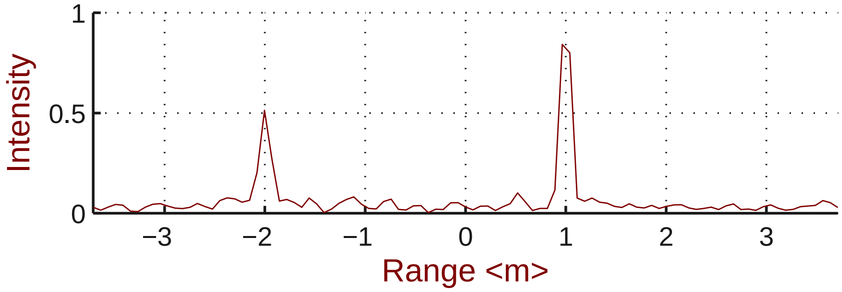


Figure 2.2: An inverse DFT applied to a frequency domain impulse response yields a map of reflection strength versus time delay or range.

The example range-intensity plot in Fig. 2.2 is composed from a synthetic signal having 2 GHz bandwidth centered at 9 GHz, and sampled in 20 MHz intervals. The simulated scatterers at 1 m and -2 m ranges have reflection coefficients of 1.0 and 0.5 respectively, and the signal to noise ratio (SNR) is 10 dB. Note how the scatterer at 1 m has a reflection magnitude of ≈0.8 instead of 1.0. This is due to how the DFT^{-1} is limited to identifying scatterers at only distinct ranges, called *range gates*, resulting in spectrally distributing the reflected energy into neighboring gates. This phenomenon is called *spectral leakage*, and ultimately, blunts the scatterers reflection intensity while introducing additional smaller false scatterers as sidelobes to the true location [39, p. 74–75].

Assuming that our targets' velocities are small compared to the transmitted EM wave's duration[1], we may model the target as being invariant during each interrogation. If we were to record this data for the duration of a target's flight, we organize the data as a matrix where each column is a successive interrogation data set in time. Computing the inverse DFT of each column and showing the resulting matrix as an image where each pixel is color coded by the magnitude of a matrix element, yields a *Range versus Time versus Intensity* (RTI) plot.

Analyzing the [500 Hz PRF] RTI plot of Fig. 2.3 reveals the two synthetic scatterers

[1] Example: The ALCOR system located at the Reagan Test Site uses a pulse duration of 10 μs. A target traveling at mach 20 would move only 6.8 cm during this time. Other radar pulse durations include 2, 50 and 565 μs. [7, 10, 28]

undergoing a rotational motion which is representative of the spinning used for stabilizing a ballistic missile's reentry vehicle [7]. Note that if two scatterers are too close in proximity, they become indistinguishable in this analysis since their EM wavefronts coherently combine such as at ≈0.6 s in Fig. 2.3.

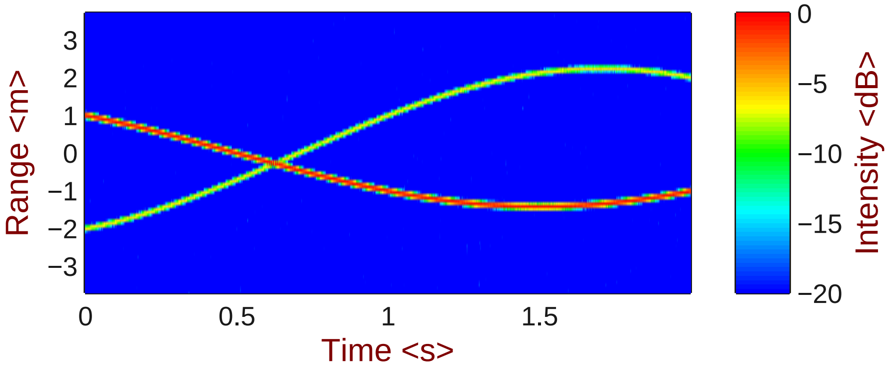


Figure 2.3: Using color to portray reflection coefficient and stacking multiple range-intensity plots collected over time produces a plot of range versus time versus intensity, which gives insight to target scene dynamics.

2.3 State-Space Theory

Now we turn our attention to a newer processing methodology based upon State-Space Theory [47]. State-Space Theory in control engineering concerns itself with the mathematical modeling of systems having some input(s), output(s), and internal state variables whose relations may be expressed via systems of first-order differential equations. In processing a target scene's impulse response, we treat the whole scene as an unknown system, where our transmitted and reflected EM waves are the respective input and output of the system as illustrated in Fig. 2.4.

Through analyzing the system's impulse response, we may employ *System Identification* to reveal the unknown parameters composing the system [30, p. 325]. In control engineering terms, our earlier assumption that targets' velocities are small with respect to pulse duration translates into our system being *Linear Time Invariant* (LTI), which greatly simplifies system identification as we will now show.

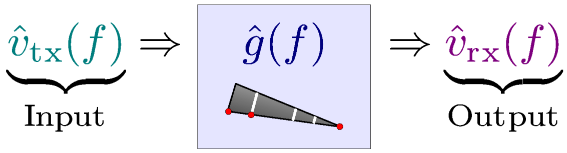

$\underbrace{\hat{v}_{\mathrm{tx}}(f)}_{\text{Input}} \Rightarrow$
$\hat{g}(f)$
$\Rightarrow \underbrace{\hat{v}_{\mathrm{rx}}(f)}_{\text{Output}}$

Figure 2.4: A systems-level block diagram expressing the target's impulse function as a linear operation on some input signal to produce an output.

In the next subsections, we first show that a target scene's frequency domain impulse response (Eq. 2.16) can be written in the form of an LTI state-space system. Following that, we then derive algebraic system identification and present example output.

2.3.1 State-Space Modeling of Frequency Domain Impulse Response

We start this algebraic derivation by introducing the LTI *discrete-time state-space equations*:

$$\begin{aligned} \vec{x}_{n+1} &= \mathrm{A}\vec{x}_n + \mathrm{B}\vec{u}_n \\ \vec{y}_n &= \mathrm{C}\vec{x}_n + \mathrm{D}\vec{u}_n \end{aligned} \qquad n = 0, 1, 2, \ldots \tag{2.17}$$

where $\vec{u}_n$ and $\vec{y}_n$ are the sequences of vector inputs and outputs, respectively. The vector $\vec{x}_n$ is called the *state* of the system, and matrix A defines how the system's state evolves from time n to $n+1$. The matrix B specifies how the input ($\vec{u}_n$) influences the current state ($\vec{x}_n$), and C defines how the current state maps to the output. D provides a means for direct coupling from inputs to outputs, but in the case of radar signal processing, $\mathrm{D} \triangleq 0$ assuming there is no direct transmitter to receiver signal leakage. If the system has q inputs, r outputs, and s state variables, then $\vec{x}_n \in \mathbb{C}^{s\times 1}$, $\vec{u}_n \in \mathbb{C}^{q\times 1}$, $\vec{y}_n \in \mathbb{C}^{r\times 1}$, $\mathrm{A} \in \mathbb{C}^{s\times s}$, $\mathrm{B} \in \mathbb{C}^{s\times q}$, $\mathrm{C} \in \mathbb{C}^{r\times s}$ and $\mathrm{D} \in \mathbb{C}^{r\times q}$.

By applying an impulse input to the system's input and assuming zero initial state,

$$\vec{u}_n = \delta_n \triangleq \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \tag{2.18}$$

$$\vec{x}_0 = 0 \tag{2.19}$$

we obtain the impulse response as:

$$\begin{aligned} \vec{y}_0 &= \mathrm{D} & \vec{x}_1 &= \mathrm{B} \\ \vec{y}_1 &= \mathrm{CB} & \vec{x}_2 &= \mathrm{AB} \\ \vec{y}_2 &= \mathrm{CAB} & \vec{x}_3 &= \mathrm{A}^2\mathrm{B} \\ \vec{y}_3 &= \mathrm{CA}^2\mathrm{B} & \vec{x}_4 &= \mathrm{A}^3\mathrm{B} \\ &\vdots & &\vdots \\ \vec{y}_n &= \mathrm{CA}^{n-1}\mathrm{B} \qquad n \geq 1 \end{aligned} \tag{2.20}$$

Through substitution, a target scene's impulse response (Eq. 2.16) can be written in identical form to that of the state-space impulse response:

$$\begin{aligned} \hat{g}_n &= \sum_s \Gamma_s e^{-4\pi j \frac{n\Delta f}{c} r_s} \qquad \text{from Eq. 2.16} \\ &= \sum_s \Gamma_s \left(e^{-4\pi j \frac{\Delta f}{c} r_s} \right)^n \\ &= \underbrace{\begin{bmatrix} \Gamma_1 & \Gamma_2 & \dots \end{bmatrix}}_{\mathrm{C}} \underbrace{\begin{bmatrix} e^{-j4\pi \frac{\Delta f}{c} r_1} & 0 & \dots \\ 0 & e^{-j4\pi \frac{\Delta f}{c} r_2} & \\ \vdots & & \ddots \end{bmatrix}^n}_{\mathrm{A}^n} \underbrace{\begin{bmatrix} 1 \\ 1 \\ \vdots \end{bmatrix}}_{\mathrm{B}} \end{aligned} \tag{2.21}$$

$$\hat{g}_n = \mathrm{CA}^n\mathrm{B} \tag{2.22}$$

Thus,

$$y_n \overset{\triangle}{=} \hat{g}_{n-1} \qquad n \geq 1 \tag{2.23}$$

□

2.3.2 Direct State-Space System Identification

In the previous section, we have shown how a target scene's impulse response can be represented in state-space. In this section we derive the *Direct State-Space* (DSS) system identification algorithm which directly estimates A, B and C directly from the target scene's frequency-domain impulse response, which in turn provides the scatterers' strengths (Γ_s)

and distances (r_s) from the radar. We will do this by first using algebraic factorizations to isolate and solve for A, and then we will use an over-determined system to solve for B and C. Though principally established by Kung [33, 34], DSS system identification has been reinvestigated several times since under different names such as ESPRIT [42, 43] and Matrix-Pencil [27].

Solving for A via Observability (or Controllability) Processing

For a given interrogation impulse response $\hat{g}_n$ from Eq. 2.16, we arrange the data into a *Hankel* matrix, $\mathrm{H} \in \mathbb{C}^{N_R \times N-N_R+1}$, where N is the total number of data samples and N_R is the number of rows. Picking $N_R = \frac{2}{3}N$ yields optimal estimation accuracy, as found in [25, 27, 52] and mentioned again in [13].

$$\mathrm{H} \triangleq \begin{bmatrix} \hat{g}_{n_0+0} & \hat{g}_{n_0+1} & \cdots & \hat{g}_{n_0+N-N_R} \\ \hat{g}_{n_0+1} & \hat{g}_{n_0+2} & & \vdots \\ \vdots & & \ddots & \\ \hat{g}_{n_0+N_R-1} & \cdots & & \hat{g}_{n_0+N-1} \end{bmatrix} \tag{2.24}$$

Recalling $\hat{g}_n = \mathrm{CA}^n\mathrm{B}$ (from Eq. 2.22), we substitute the elements of H and show that the Hankel form admits a factorization which isolates B and C:

$$\begin{aligned} \mathrm{H} &= \begin{bmatrix} \mathrm{CA}^{n_0+0}\mathrm{B} & \mathrm{CA}^{n_0+1}\mathrm{B} & \dots & \mathrm{CA}^{n_0+N-N_R}\mathrm{B} \\ \mathrm{CA}^{n_0+1}\mathrm{B} & \mathrm{CA}^{n_0+2}\mathrm{B} & & \vdots \\ \vdots & & \ddots & \\ \mathrm{CA}^{n_0+N_R-1}\mathrm{B} & \dots & & \mathrm{CA}^{n_0+N-1}\mathrm{B} \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} \mathrm{CA}^{\frac{n_0}{2}+0} \\ \mathrm{CA}^{\frac{n_0}{2}+1} \\ \vdots \\ \mathrm{CA}^{\frac{n_0}{2}+N_R-1} \end{bmatrix} \begin{bmatrix} \mathrm{A}^{\frac{n_0}{2}+0}\mathrm{B} & \mathrm{A}^{\frac{n_0}{2}+1}\mathrm{B} & \dots & \mathrm{A}^{\frac{n_0}{2}+N-N_R}\mathrm{B} \end{bmatrix}}_{\mathrm{H}=\mathcal{O}\mathcal{C}} \end{aligned} \tag{2.25}$$

From control system theory, we identify the left and right matrices ($\mathcal{O}$ and $\mathcal{C}$) from factoring the Hankel as the *Observability* and *Controllability* matrices, respectively. Defining

$\mathcal{O}_-$ and $\mathcal{O}_+$ as $\mathcal{O}$ with either the last row or the first row deleted,

$$\mathcal{O}_- \triangleq \begin{bmatrix} \mathrm{CA}^{\frac{n_0}{2}+0} \\ \mathrm{CA}^{\frac{n_0}{2}+1} \\ \vdots \\ \mathrm{CA}^{\frac{n_0}{2}+N_R-2} \end{bmatrix} \qquad \mathcal{O}_+ \triangleq \begin{bmatrix} \mathrm{CA}^{\frac{n_0}{2}+1} \\ \mathrm{CA}^{\frac{n_0}{2}+2} \\ \vdots \\ \mathrm{CA}^{\frac{n_0}{2}+N_R-1} \end{bmatrix} \tag{2.26}$$

we may solve the linear system

$$\mathcal{O}_-\mathrm{A} = \mathcal{O}_+ \tag{2.27}$$

for A using Least-Squares (LS), Total-Least-Squares (TLS) [20] or Unitary-TLS [3] approaches [47, p. 295]. We call this method of solving for A from $\mathcal{O}$ *observability-based DSS processing*, or observability processing for short.

Alternatively, we could have solved for A from $\mathcal{C}$ via a similar procedure, in which case we could call this method *controllability-based DSS processing* and illustrate it briefly in Eqs. 2.28–2.30. With regard to the size of H, we would set the number of columns to be $\frac{2}{3}N$ (instead of rows) by setting $N_R = \frac{1}{3}N$, but in doing so, this procedure becomes analytically identical to that of observability processing, except for all of the components being transposed. Thus in the remainder of this research, we may not specifically refer to controllability processing, but any discussion regarding observability processing applies equally here as well.

$$\mathcal{C}_- \triangleq \left[\mathrm{A}^{\frac{n_0}{2}+0}\mathrm{B} \quad \mathrm{A}^{\frac{n_0}{2}+1}\mathrm{B} \quad \ldots \quad \mathrm{A}^{\frac{n_0}{2}+N-N_R-1}\mathrm{B}\right] \tag{2.28}$$

$$\mathcal{C}_+ \triangleq \left[\mathrm{A}^{\frac{n_0}{2}+1}\mathrm{B} \quad \mathrm{A}^{\frac{n_0}{2}+2}\mathrm{B} \quad \ldots \quad \mathrm{A}^{\frac{n_0}{2}+N-N_R}\mathrm{B}\right] \tag{2.29}$$

$$\mathrm{A}\mathcal{C}_- = \mathcal{C}_+ \tag{2.30}$$

In both of the processing methods, there are innumerable ways to factor H into $\mathcal{O}$ and $\mathcal{C}$ due to the non-uniqueness of state-space representations [30, p. 53] [47, p. 294] even including trivial cases such as $\mathcal{O} \triangleq \mathrm{I}$, $\mathcal{C} \triangleq \mathrm{H}$ [12, p. 203], and thus, factoring to solve for A does not directly produce the diagonal canonical form we picked in Eq. 2.21. However, the obtained alternative representation will be related to our canonical one via a *similarity* *transformation*, and we denote the transformed parameters for this using $\sim$ as shown in Eq. 2.31 and Eq. 2.32. Compactly stated here, A, B, and C may be transformed so that instead of having $\hat{g}_n = \mathrm{C}\mathrm{A}^n\mathrm{B}$, we have $\hat{g}_n = \left(\mathrm{C}\mathrm{T}^{-1}\right)\left(\mathrm{T}\mathrm{A}\mathrm{T}^{-1}\right)^n(\mathrm{T}\mathrm{B}) = \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^n\widetilde{\mathrm{B}}$ for some non-singular T. See Appendix A for more details.

$$\mathrm{H} = \begin{bmatrix} \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{\frac{n_0}{2}+0} \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{\frac{n_0}{2}+1} \\ \vdots \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{\frac{n_0}{2}+N_R-1} \end{bmatrix} \begin{bmatrix} \widetilde{\mathrm{A}}^{\frac{n_0}{2}+0}\widetilde{\mathrm{B}} & \widetilde{\mathrm{A}}^{\frac{n_0}{2}+1}\widetilde{\mathrm{B}} & \dots & \widetilde{\mathrm{A}}^{\frac{n_0}{2}+N-N_R}\widetilde{\mathrm{B}} \end{bmatrix} \tag{2.31}$$

$$= \widetilde{\mathcal{O}}\widetilde{\mathcal{C}} \tag{2.32}$$

$$\widetilde{\mathcal{O}}_- \triangleq \begin{bmatrix} \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{\frac{n_0}{2}+0} \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{\frac{n_0}{2}+1} \\ \vdots \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{\frac{n_0}{2}+N_R-2} \end{bmatrix} \qquad \widetilde{\mathcal{O}}_+ \triangleq \begin{bmatrix} \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{\frac{n_0}{2}+1} \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{\frac{n_0}{2}+2} \\ \vdots \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{\frac{n_0}{2}+N_R-1} \end{bmatrix} \tag{2.33}$$

$$\widetilde{\mathcal{O}}_-\widetilde{\mathrm{A}} = \widetilde{\mathcal{O}}_+ \tag{2.34}$$

To factor H into $\widetilde{\mathcal{O}}$ and $\widetilde{\mathcal{C}}$, the usual choice is based upon the singular value decomposition (SVD) (Eq. 2.35) for its relative insensitivity to matrix perturbations as well as its ability to construct optimal low-rank approximations [47, p. 289]. In our radar example, we have matrix perturbations stemming from the additive noise present in the received signal, measured by SNR, and the rank approximations become important as a tool to estimate the number of scatterers as well as suppressing noise.

For a given matrix $\mathrm{X} \in \mathbb{C}^{q\times r}$, the SVD produces three new matrices, $\mathrm{U} \in \mathbb{C}^{q\times q}$, $\mathrm{V} \in \mathbb{C}^{r\times r}$ and $\Sigma \in \mathbb{R}^{q\times r}$, where $\mathrm{U}\mathrm{U}^H = \mathrm{U}^H\mathrm{U} = \mathrm{I}_q$, $\mathrm{V}\mathrm{V}^H = \mathrm{V}^H\mathrm{V} = \mathrm{I}_r$ and $\Sigma = diag(\sigma_1, \sigma_2, \dots)$ with $\sigma_n \geq \sigma_{n+1}$. [21, p. 70–73]

$$\mathrm{X} \overset{\text{svd}}{\Rightarrow} \mathrm{U}\Sigma\mathrm{V}^H \tag{2.35}$$

However, SVD rank truncation does not preserve Hankel structure, and there are numerous avenues of research into this topic such as [37] which states “no analytic form for [Hankel-structure preserving rank reduction] exists to date.” One of the more promising techniques is the $O(n^2 \log n)$ Fast-Hankel SVD algorithm in [38, 53], but unfortunately, the work is focused on one-dimensional, range-only Hankel matrices such as those presented in this chapter for introductory purposes and not directly applicable to the multidimensional (or "matrix-enhanced") Hankels which will be introduced in Section 3.3 for processing range and range-rate simultaneously. Another point about the Fast-Hankel SVD is that it is tailored for square Hankel matrices and further work is needed if it is to be applied to matrices $\in \mathbb{C}^{\frac{2}{3}N \times \frac{1}{3}N}$ where N is the total number of data samples. Regardless of this limitation of the SVD with regard to preserving Hankel structure, "the SVD is [still] the computational tool of choice in modern signal processing" [47], and is employed in the present research.

Say our target scene contains 15 scatterers and our radar system collects the impulse response at 100 distinct frequencies, observability processing would build a Hankel matrix having 67 rows and 34 columns. Without noise, we could test the matrix rank which provides a direct measure of the number of scatterers, but with noise, the matrix becomes full rank. Additive noise manifests itself in the singular spectrum of H by introducing singular values which are typically much smaller than the matrix's singular values. Consequently, we identify the larger singular values with the *signal space* of our data and the smaller singular values with the *noise space*. Letting s stand for the number of large singular values, the SVD permits us to identify the closest norm-2 rank-s Hankel matrix via truncation of U, Σ and V^H [47]:

$$\mathrm{H} \overset{\text{svd}}{\Rightarrow} \mathrm{U}\Sigma\mathrm{V}^H \tag{2.36}$$

$$\mathrm{U}_{\mathrm{tr}} \triangleq \mathrm{U}_{1:s,:} \qquad \text{first } s \text{ columns of U} \tag{2.37}$$

$$\Sigma_{\mathrm{tr}} \triangleq \Sigma_{1:s,1:s} \qquad \text{first } s \text{ singular values of } \Sigma \tag{2.38}$$

$$\mathrm{V}_{\mathrm{tr}} \triangleq \mathrm{V}_{1:s,:} \qquad \text{first } s \text{ columns of V} \tag{2.39}$$

In order to distinguish the appropriate cut-off s, the Akaike Information Criterion (AIC) [2] and Minimum Description Length (MDL) [49, 48] are often utilized in array processing for estimating the number of sources [13]. However, there is ongoing research in source number estimation, pointing to the suboptimality due to these methods often over-estimating s and thus introducing a number of false sources [19, 11]. Instead, another common approach, which is to "plot the Hankel singular values to visualize the trade-off between model order and identification accuracy", [4] is suitable for non-automated processing of specialized data sets but not for real-time radar signature processing. For automated processing, we apply an empirically derived heuristic based on selecting the largest singular value, σ_1, and any smaller values that are within 20 dB of the largest as shown in Eq. 2.40.

$$\max_s 20 \log_{10} \frac{\sigma_1}{\sigma_s} \leq 20\text{dB} \tag{2.40}$$

Once we possess U_{tr}, Σ_{tr} and V_{tr}, we obtain $\widetilde{\mathcal{O}}$ and $\widetilde{\mathcal{C}}$ via Eq. 2.41 and Eq. 2.42,[2] and then we may obtain $\widetilde{\text{A}}$ by solving either Eq. 2.27 or Eq. 2.30. Eq. 2.43 shows this solution using a Least-Squares approach. Finding T to diagonalize $\widetilde{\text{A}}$ into A may be accomplished by the eigen decomposition as shown in Eq. 2.44.

$$\widetilde{\mathcal{O}} \Leftarrow \text{U}_{\text{tr}} \sqrt{\Sigma_{\text{tr}}} \tag{2.41}$$

$$\widetilde{\mathcal{C}} \Leftarrow \sqrt{\Sigma_{\text{tr}}} \text{V}_{\text{tr}}^H \tag{2.42}$$

$$\widetilde{\text{A}} \Leftarrow \widetilde{\mathcal{O}}_-^\dagger \widetilde{\mathcal{O}}_+ \qquad \text{or} \qquad \widetilde{\text{A}} \Leftarrow \widetilde{\mathcal{C}}_+ \widetilde{\mathcal{C}}_-^\dagger \tag{2.43}$$

$$\widetilde{\text{A}} \overset{\text{eig}}{\Rrightarrow} \text{TAT}^{-1} \tag{2.44}$$

Alternatively, the solution for A and T may also be solved as a generalized eigenvalue problem from $\widetilde{\mathcal{O}}$ in Eq. 2.45 or from $\widetilde{\mathcal{C}}$ in Eq. 2.46.

$$\widetilde{\mathcal{O}}_- \text{TA} = \widetilde{\mathcal{O}}_+ \text{T} \tag{2.45}$$

$$\text{AT}^{-1} \widetilde{\mathcal{C}}_- = \text{T}^{-1} \widetilde{\mathcal{C}}_+ \tag{2.46}$$

From Eq. 2.21, we recall that the diagonal elements of A encode the scatterers' ranges, so we only need to compute the natural logarithm of $\widetilde{\text{A}}$'s eigenvalues and divide off $-j4\pi\frac{\Delta f}{c}$

[2] This particular factoring style of Eq. 2.41 and Eq. 2.42 produces what is called a *balanced realization* in state space theory, because $\mathcal{O}\mathcal{O}^H = \mathcal{C}\mathcal{C}^H$ [12, p. 205].

as shown here:

$$\mathrm{A} \triangleq \begin{bmatrix} e^{-j4\pi\frac{\Delta f}{c}r_1} & 0 & \dots & 0 \\ 0 & e^{-j4\pi\frac{\Delta f}{c}r_2} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & e^{-j4\pi\frac{\Delta f}{c}r_s} \end{bmatrix} \tag{2.47}$$

$$\begin{bmatrix} r_1 & r_1 & \dots & r_s \end{bmatrix} \Leftarrow \frac{\ln\left(\operatorname{diag}\mathrm{A}\right)}{-j4\pi\frac{\Delta f}{c}} \tag{2.48}$$

To find the reflection strengths of each scatterer, we could finish solving for $\widetilde{\mathrm{B}}$ and $\widetilde{\mathrm{C}}$ from $\widetilde{\mathcal{C}}$ and $\widetilde{\mathcal{O}}$, respectively, and then applying the appropriate system transformations to set $\mathrm{B} = \begin{bmatrix} 1 & 1 & \dots \end{bmatrix}^T$. However, we will abbreviate such mechanisms via rewriting Eq. 2.22 into a linear system fitting C to our original data $\hat{g}_n$:

$$\begin{bmatrix} \hat{g}_{n_0} & \hat{g}_{n_0+1} & \dots \end{bmatrix} = \mathrm{C} \begin{bmatrix} \mathrm{A}^{n_0}\mathrm{B} & \mathrm{A}^{n_0+1}\mathrm{B} & \dots \end{bmatrix} \tag{2.49}$$

$$\mathrm{C} = \begin{bmatrix} \hat{g}_{n_0} & \hat{g}_{n_0+1} & \dots \end{bmatrix} \begin{bmatrix} \mathrm{A}^{n_0}\mathrm{B} & \mathrm{A}^{n_0+1}\mathrm{B} & \dots \end{bmatrix}^{\dagger} \tag{2.50}$$

$$\mathrm{C} \triangleq \begin{bmatrix} \Gamma_{n_0+1} & \Gamma_{n_0+2} & \dots \end{bmatrix} \tag{2.51}$$

By solving for the reflection strengths without needing both $\widetilde{\mathcal{C}}$ and $\widetilde{\mathcal{O}}$, we may speed up the numerical evaluation of the SVD of Eq. 2.36 in that we do not need to compute V when using observability processing or U when using controllability processing. Additionally, in typical processing cases, the number of frequencies measured during an interrogation tends to far out-number the number of scatterers in the target scene, so Eq. 2.50 is over-determined.

Solving for A via Balanced Processing

In the previous section, A was only solved using either $\mathcal{O}$ or $\mathcal{C}$, and never both simultaneously, and this seems to be the common approach in the signal estimation community as highlighted in 1992: “It seems that using both [$\mathcal{O}$ and $\mathcal{C}$] matrices might yield better estimates, but such attempt has not been successful.” [26] However, at least 7 years earlier in the field of control systems theory, such a method had been described and often referred to as either the *Eigensystem Realization Algorithm* [29, 4] or as *Balanced Realization from*

Markov Parameters [12, p. 203]. The research is still continuing in the mechanical and control theory fields as evidenced by [46, 9, 5], but there is a notable lack of reference to this technique in the signal processing communities.[3]

Continuing within the naming scheme of the present research, we call this method *balanced DSS processing* and begin our description by arranging the interrogation impulse response data ($\hat{g}_n$ from Eq. 2.16) into two equal sized Hankel matrices, H_{n_0} and H_{n_0+1}, as shown in Eq. 2.52, where each matrix is comprised of one sample less than the total number of frequency-domain samples in an interrogation, N. Similar to observability processing, the number of rows in H_{n_0} and H_{n_0+1} should be two to three times the number of columns [29, 9], so we pick $N_R = \frac{2}{3}N$.

$$\mathrm{H}_{n_0} \triangleq \begin{bmatrix} \hat{g}_{n_0} & \hat{g}_{n_0+1} & \cdots & \hat{g}_{n_0+N-N_R-1} \\ \hat{g}_{n_0+1} & \hat{g}_{n_0+2} & & \vdots \\ \vdots & & \ddots & \\ \hat{g}_{n_0+N_R-1} & \cdots & & \hat{g}_{n_0+N-2} \end{bmatrix} \tag{2.52}$$

$$\mathrm{H}_{n_0+1} \triangleq \begin{bmatrix} \hat{g}_{n_0+1} & \hat{g}_{n_0+2} & \cdots & \hat{g}_{n_0+N-N_R} \\ \hat{g}_{n_0+2} & \hat{g}_{n_0+3} & & \vdots \\ \vdots & & \ddots & \\ \hat{g}_{n_0+N_R} & \cdots & & \hat{g}_{n_0+N-1} \end{bmatrix} \tag{2.53}$$

Substituting the elements of H_{n_0} and H_{n_0+1} via $\hat{g}_n = \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^n\widetilde{\mathrm{B}}$ again reveals factorizations

[3] We attribute this communication gap between the signals processing and control systems communities to their oft-diverging vocabulary.

isolating $\widetilde{\mathrm{B}}$ and $\widetilde{\mathrm{C}}$:

$$\begin{aligned}
\mathrm{H}_{n_0} &= \begin{bmatrix} \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{n_0}\widetilde{\mathrm{B}} & \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{n_0+1}\widetilde{\mathrm{B}} & \dots \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{n_0+1}\widetilde{\mathrm{B}} & \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{n_0+2}\widetilde{\mathrm{B}} & \\ \vdots & & \ddots \end{bmatrix} \\
&= \begin{bmatrix} \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{\frac{n_0}{2}} \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{\frac{n_0}{2}+1} \\ \vdots \end{bmatrix} \begin{bmatrix} \widetilde{\mathrm{A}}^{\frac{n_0}{2}}\widetilde{\mathrm{B}} & \widetilde{\mathrm{A}}^{\frac{n_0}{2}+1}\widetilde{\mathrm{B}} & \dots \end{bmatrix} && (2.54) \\
&= \widetilde{\mathcal{O}}\widetilde{\mathcal{C}} && (2.55) \\
\mathrm{H}_{n_0+1} &= \begin{bmatrix} \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{n_0+1}\widetilde{\mathrm{B}} & \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{n_0+2}\widetilde{\mathrm{B}} & \dots \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{n_0+2}\widetilde{\mathrm{B}} & \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{n_0+3}\widetilde{\mathrm{B}} & \\ \vdots & & \ddots \end{bmatrix} \\
&= \begin{bmatrix} \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{\frac{n_0}{2}} \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^{\frac{n_0}{2}+1} \\ \vdots \end{bmatrix} \widetilde{\mathrm{A}} \begin{bmatrix} \widetilde{\mathrm{A}}^{\frac{n_0}{2}}\widetilde{\mathrm{B}} & \widetilde{\mathrm{A}}^{\frac{n_0}{2}+1}\widetilde{\mathrm{B}} & \dots \end{bmatrix} && (2.56) \\
&= \widetilde{\mathcal{O}}\widetilde{\mathrm{A}}\widetilde{\mathcal{C}} && (2.57)
\end{aligned}$$

Note how Eqs. 2.54 and 2.56 reveal that H_{n_0} and H_{n_0+1} share common components, $\widetilde{\mathcal{O}}$ and $\widetilde{\mathcal{C}}$, and the only difference is that H_{n_0+1} has an additional $\widetilde{\mathrm{A}}$. Factoring H_{n_0} into $\widetilde{\mathcal{O}}$ and $\widetilde{\mathcal{C}}$, permits us to divide these components from H_{n_0+1} to isolate $\widetilde{\mathrm{A}}$, and we accomplish this factoring using the SVD in a similar fashion to the observability processing described in the previous section. Eq. 2.63 shows using the eigen decomposition to identify the transformation matrix T to obtain a diagonal A from $\widetilde{\mathrm{A}}$.

$$\mathrm{H}_{n_0} \overset{\text{svd}}{\Rightarrow} \mathrm{U}\Sigma\mathrm{V}^H \tag{2.58}$$

$$s \triangleq \text{the number of large } \sigma_i \in \Sigma \tag{2.59}$$

$$\widetilde{\mathcal{O}} \Leftarrow \mathrm{U}_{1:s,:}\sqrt{\Sigma_{1:s,1:s}} \tag{2.60}$$

$$\widetilde{\mathcal{C}} \Leftarrow \sqrt{\Sigma_{1:s,1:s}}\mathrm{V}^H_{1:s,:} \tag{2.61}$$

$$\widetilde{\mathrm{A}} \Leftarrow \widetilde{\mathcal{O}}^\dagger \mathrm{H}_{n_0+1}\widetilde{\mathcal{C}}^\dagger \tag{2.62}$$

$$\widetilde{\mathrm{A}} \overset{\text{eig}}{\Rightarrow} \mathrm{T}\mathrm{A}\mathrm{T}^{-1} \tag{2.63}$$

Note that the pseudo inverse of $\widetilde{\mathcal{O}}$ and $\widetilde{\mathcal{C}}$ in Eq. 2.62 is available at no additional computational cost for balanced processing because we do not have to truncate $\widetilde{\mathcal{O}}$ (or $\widetilde{\mathcal{C}}$) to form $\mathcal{O}_-$ and $\mathcal{O}_+$ (or $\mathcal{C}_-$ and $\mathcal{C}_+$) as we did for observability processing in Eq. 2.43. Since these matrices were composed from the unitary and diagonal outputs from the SVD, their pseudo inverses are

$$\widetilde{\mathcal{O}}^\dagger \Leftarrow \sqrt{\Sigma_{1:s,1:s}}^{-1}\mathrm{U}^H_{1:s,:} \tag{2.64}$$

$$\widetilde{\mathcal{C}}^\dagger \Leftarrow \mathrm{V}_{1:s,:}\sqrt{\Sigma_{1:s,1:s}}^{-1} \tag{2.65}$$

The balanced processing procedure to identify the remaining unknowns given A is the same as the step used in observability processing in Eq. 2.50.

2.3.3 Example Output

To provide a qualitative comparison of the difference in results obtained for Fourier and DSS processing, this section first presents sample results from the simple synthetic data set of isotropic point scatterers introduced earlier, and then presents an example of processing a representative target whose radar signature was measured at a compact-range.

Continuing from the example data used for Fig. 2.3, Fig. 2.5 presents analysis results from both observability and balanced form Direct State-Space processing. The upper plot was formed via observability processing, and the lower plot was formed via balanced processing. Visually, both processing methods appear to produce similar results, but we will withhold further analysis until Chapter 4.

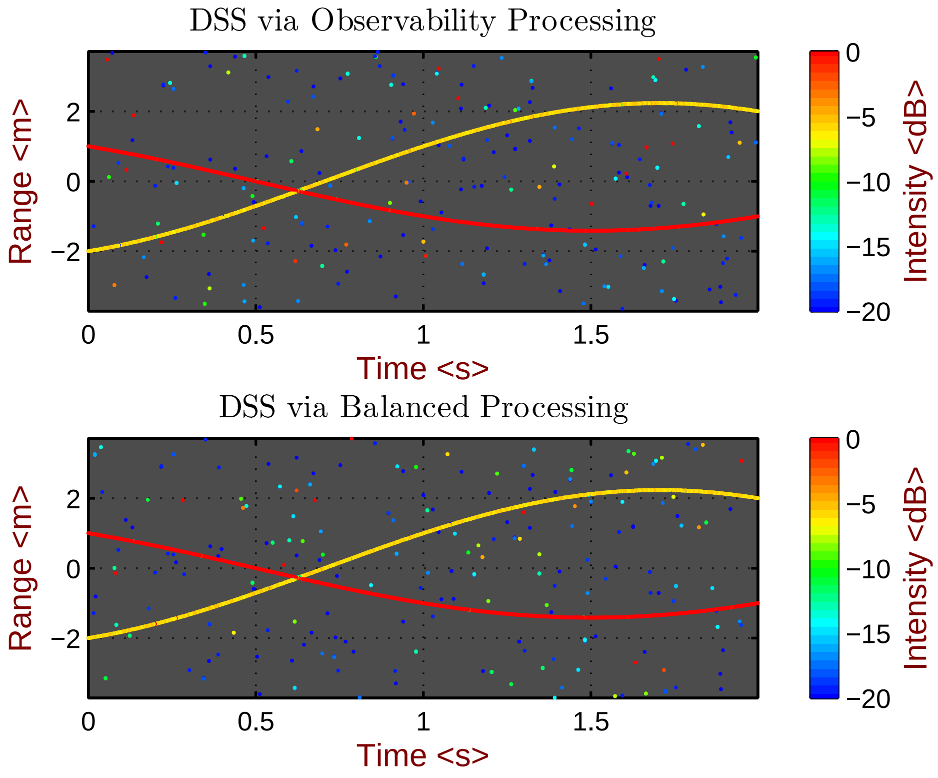


Figure 2.5: Using color to portray reflection coefficient and stacking multiple range-intensity plots collected over time produces a plot of range versus time versus intensity, which gives insight to target scene motion dynamics.

For an example of the utility of state-space processing versus Fourier processing, we briefly introduce a test target, “LL-Cone” in Fig. 2.6. The target’s radar response over 2-18 GHz was measured in an anechoic chamber at Point Mugu, CA, and has been used extensively at MIT Lincoln Laboratory for electromagnetic diffraction studies and algorithm validation. A detailed specification for this object was provided by Lincoln Laboratory [17]. Though many other studies exist including [13, 8, 45], we will only highlight a couple of processing examples regarding side-lobe suppression and attributed scattering centers in our prior work developing the Parametric Scattering Law Identification (PSLI) algorithm [24].

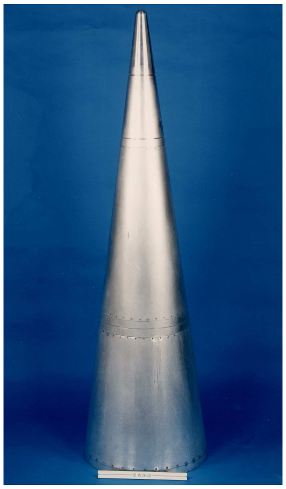


Figure 2.6: LL-Cone.

We first highlight Fig. 2.7, which is a side-by-side comparison of direct state-space’s superior scatterer resolving ability as compared to the main-lobe blurring of Fourier processing. Unlike Fourier processing, state-space processing does not introduce sidelobes which typically obfuscate radar signature interpretation. In the Fourier plot on the left, the double base edge diffraction at aspect angles of ±100 degrees are blurred together. In the state-space plot on the right, the leading and trailing side of the base edge is clearly resolved as two scattering centers.

The second result to highlight, Fig. 2.8, is a plot of scattering feature range versus aspect angle, but instead of color coding the scatterers by reflection intensity, they are color coded

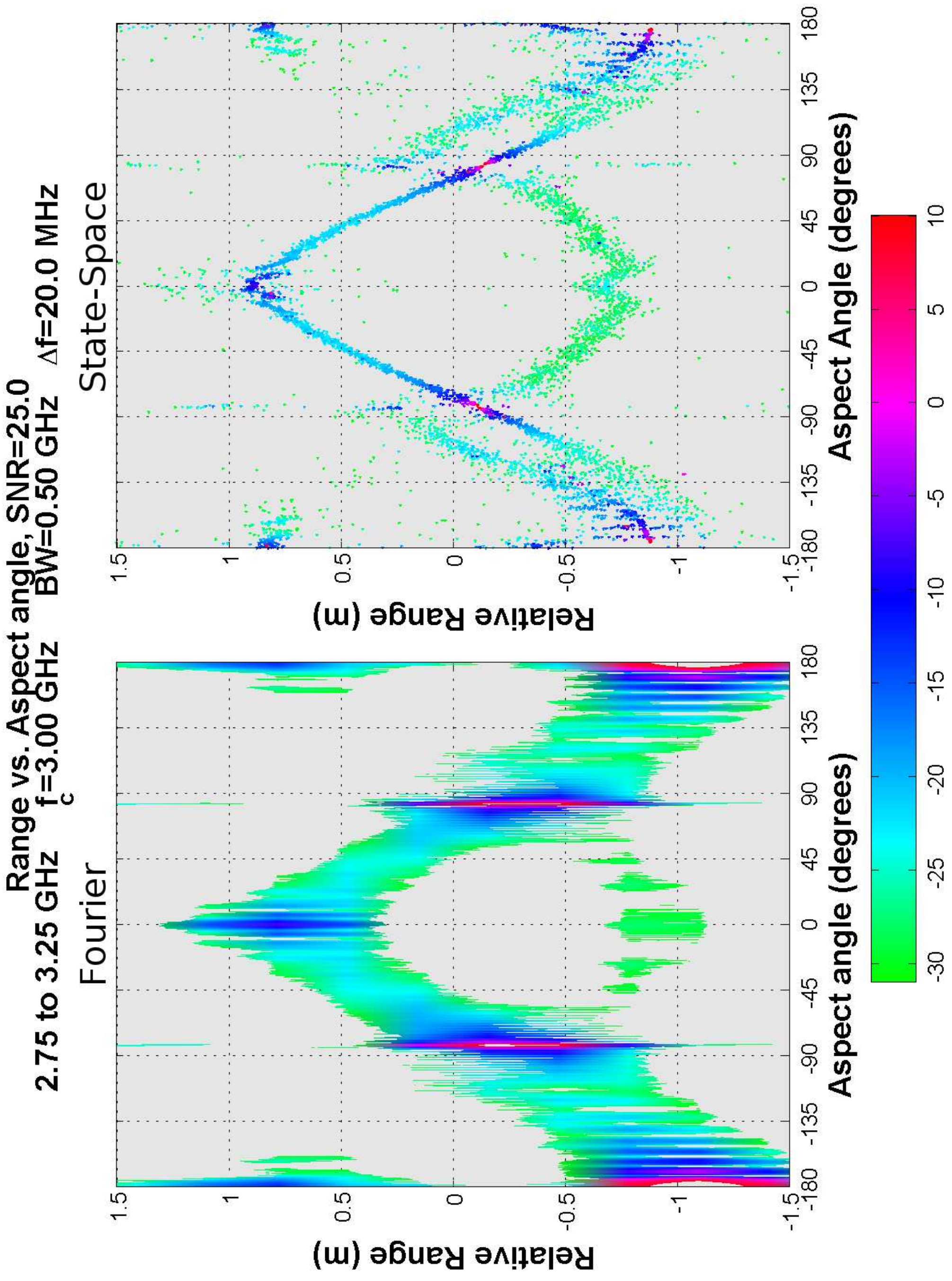


Figure 2.7: Fourier vs. State-Space, 2.75-3.25 GHz, 25 dB SNR. This figure is reprinted with permission from [24].

by each scatterer's frequency dependence, which may be described as f^{α}. Keller's *Geometric Theory of Diffraction* [32] states that "scattering centers exhibit a frequency dependence proportional to a power-law parameter, f^{α}, where $\alpha = n/2$ for small positive and negative integers, n." [24] Using Table 2.3.3 as a key, each scatterer's frequency dependence provides clues as to the underlying physical geometry. PSLI [24] is a fully automated algorithm for identifying α, which is made possible through direct state-space's discrete nature in reflector/source identification.

Scattering Geometries	Power Law
Discontinuity of curvature on edge	$-4/2$
Cone tip	$-2/2$
Curved-edge diffraction	$-1/2$
Doubly curved surface, straight edge	$0/2$
Singly curved surface (cylinder)	$+1/2$
Corner reflector, plate	$+2/2$
Groove, duct	$+3/2$
Rayleigh scattering	$+4/2$

Table 2.1: GTD-predicted α for various scattering features.

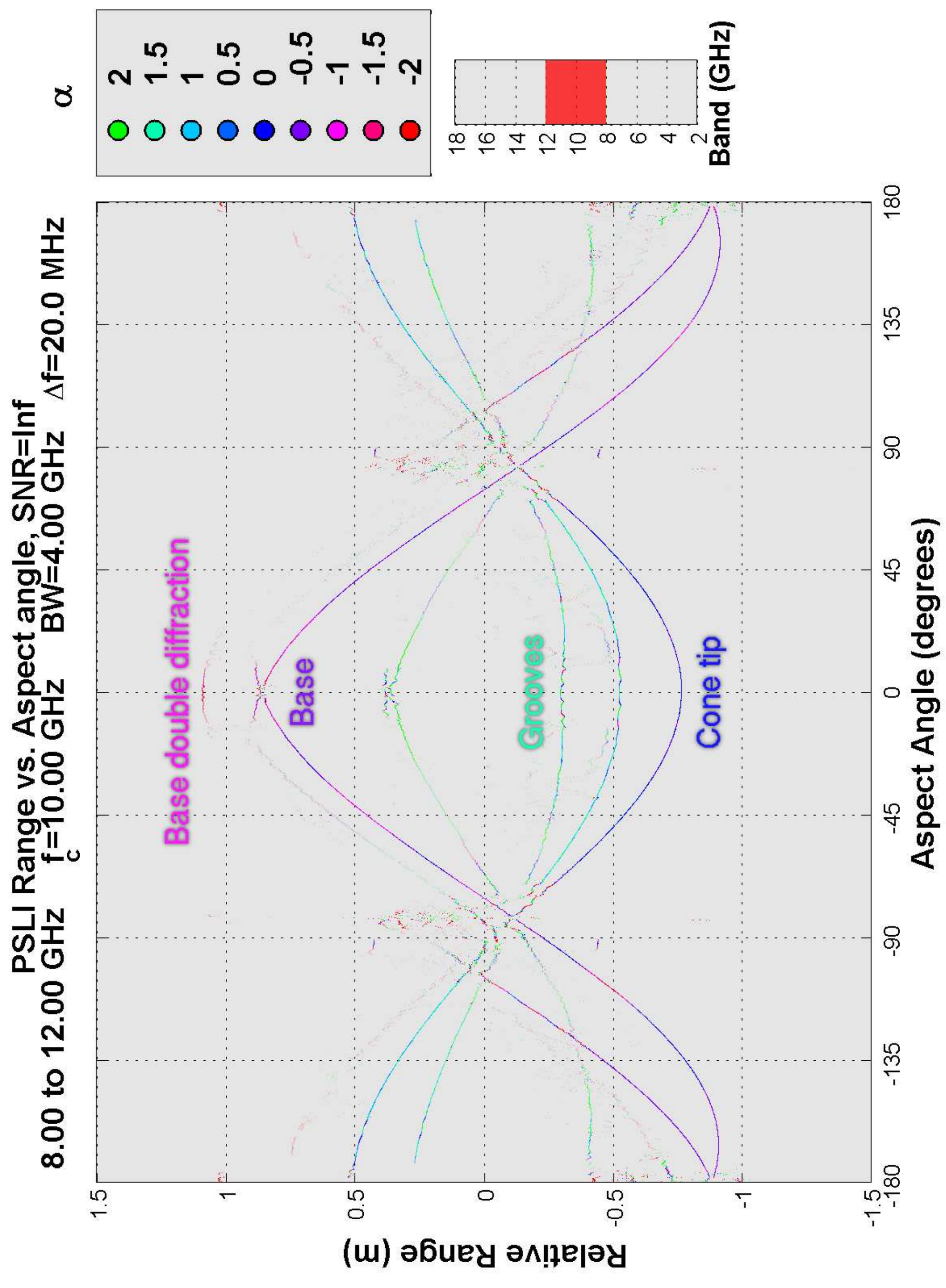


Figure 2.8: A plot of range versus aspect angle of LL-Cone's scattering features processed over 8-12 GHz. The individual scatterers are color coded by their frequency dependence instead of reflection intensity to aid in identification. This figure is reprinted with permission from [24].

Chapter 3

Joint-Doppler Processing

The previous chapter presented two methods for estimating range and reflection intensity of scatterers in a target scene, Fourier processing and Direct State-Space analysis. This chapter extends the processing to estimate the change in range per unit time, *range-rate*, by first presenting the established Fourier and DSS Joint-Doppler equivalents, and then introduces our new processing methodologies in an exploration to identify algorithmic extensions and potentially reduce limitations in present algorithms.

We start by presenting our original radar problem Eq. 2.16, except we added $\dot{r}_s t$ to r_s to permit any s scatterer's range to change linearly over time:

$$\hat{g}_n(t) = \sum_s \Gamma_s e^{-j4\pi \frac{n\Delta f}{c} \cdot (r_s + \dot{r}_s t)} \tag{3.1}$$

Now assuming that our radar interrogates the target scene at a given pulse repetition frequency (PRF), we discretize time by substituting $t_k = k\Delta t$ where $\Delta t = \frac{1}{\text{PRF}}$. This results in $\hat{g}_{n,k}$ which designates the target scene's frequency domain response for the n-th frequency and k-th interrogation:

$$\hat{g}_{n,k} = \sum_s \Gamma_s e^{-j4\pi \frac{n\Delta f}{c} \cdot (r_s + \dot{r}_s k\Delta t)} \tag{3.2}$$

3.1 Separating Range and Range-Rate

By solving for ranges and range-rates simultaneously, we are tackling a joint estimation problem. To show this, we expand the exponential of Eq. 3.2 as

$$\hat{g}_{n,k} = \sum_s \Gamma_s e^{-j4\pi \frac{n\Delta f}{c} r_s} e^{-j4\pi \frac{n\Delta f k \Delta t}{c} \dot{r}_s} \tag{3.3}$$

and highlight the simple product of $n\Delta f k\Delta t$ in the resulting exponential, which complicates attempts to solve for range and range-rates independently.

As discussed in [8, p.526], we can separate the complex exponentials for range and range-rate by time-domain resampling $\hat{g}_{n,k}$. Mathematically, we write this as a substitution

$$n\Delta f \Delta t \Rightarrow f_0 \Delta t' \tag{3.4}$$

where f_0 is the lowest frequency of the target scene interrogation. We denote the resulting data as $\tilde{\hat{g}}_{n,k}$ in Eq. 3.5. Fig. 3.1 illustrates this process.

$$\tilde{\hat{g}}_{n,k} = \sum_s \Gamma_s e^{-j4\pi \frac{n\Delta f}{c} r_s} e^{-j4\pi \frac{f_0 k \Delta t'}{c} \dot{r}_s} \tag{3.5}$$

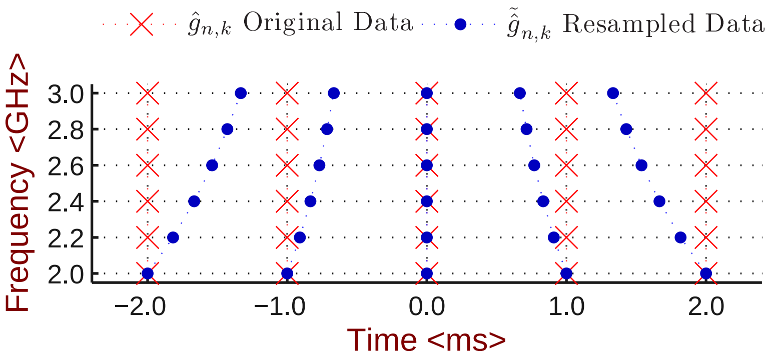


Figure 3.1: At 40% bandwidth, this radar data needs significant interpolation to separate r_s and $\dot{r}_s$.

As expressed in Eq. 3.4, the precise parameters employed in the resampling depends on the frequency and bandwidth of the signal under consideration. Percent bandwidth is the relative measure of bandwidth versus center frequency,

$$\%\text{BW} = \frac{\text{BW}}{f_{\text{center}}}$$

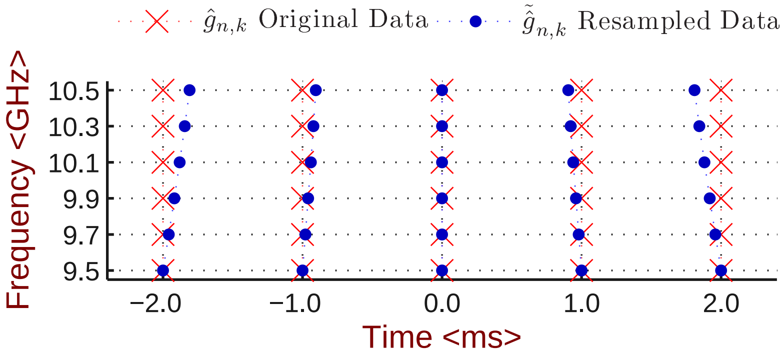


Figure 3.2: At 10% bandwidth, this radar data may be small enough to neglect resampling without causing too much error in r_s and $\dot{r}_s$ estimation.

and data with a percent bandwidth under 5-10% does not need significant interpolation as illustrated in Fig. 3.2. Older radar systems tend to have relatively small %BW, so many methods employ a narrowband assumption in one form or another and thus skip this resampling step. Newer systems are designed to have more than 20 %BW to fulfill the demand for higher system resolutions, and they may accomplish this with either a single radio system, or by *fusing* multiple smaller %BW systems together [13, 14]. Inadvertently processing wideband data with narrowband algorithms establishes ripe conditions for degraded range and range-rate estimates, so it is important to keep the focus on our algorithms with respect to an eye for future bandwidth capabilities.

3.2 Fourier Processing

Rewriting Eq. 3.5, we recognize this function as a two-dimensional discrete Fourier transform [39, p. 158] of Γ_s in k-space,

$$\begin{aligned}
\tilde{\tilde{g}}_{n,k} &= \sum_s \Gamma_s e^{-j4\pi \frac{n\Delta f}{c} r_s} e^{-j4\pi \frac{f_0 k \Delta t'}{c} \dot{r}_s} \qquad \text{from Eq. 3.5} \\
&= \sum_s \Gamma_s e^{-j2\pi\left(\frac{2r_s}{c} n\Delta f + \frac{2 f_0 \dot{r}_s}{c} k \Delta t'\right)} && (3.6) \\
n\Delta f &\Rightarrow f_n \qquad k\Delta t' \Rightarrow t'_k && (3.7) \\
\tilde{\tilde{g}}(f, t') &= \sum_s \Gamma_s e^{-j2\pi\left(\frac{2r_s}{c} f + \frac{2 f_0 \dot{r}_s}{c} t'\right)} && (3.8) \\
k_r &\triangleq \frac{2f}{c} \qquad k_{\dot{r}} \triangleq \frac{2 f_0 t'}{c} && (3.9) \\
\tilde{\tilde{g}}(k_r, k_{\dot{r}}) &= \sum_s \Gamma_s e^{-j2\pi(k_r r_s + k_{\dot{r}} \dot{r}_s)} && (3.10)
\end{aligned}$$

so we may use the inverse transform to identify Γ_s versus range and range-rate:

$$\tilde{g}(r, \dot{r}) \triangleq \mathcal{F}^{-1}_{r,\dot{r}}\Big\{\tilde{\tilde{g}}(k_r, k_{\dot{r}})\Big\} \qquad (3.11)$$

$$\tilde{g}(r, \dot{r}) = \sum_s \Gamma_s\, \delta(r - r_s)\, \delta(\dot{r} - \dot{r}_s) \qquad (3.12)$$

Sometimes called “block” processing, Fourier Doppler processing is accomplished via processing a small number of interrogations at a time, and then advancing this window as shown in Fig. 3.4. Within this window, data is first resampled to separate r_s and $\dot{r}_s$, then scaled by a *window function*, and finally processed with a two-dimensional inverse discrete Fourier transform.

Applying window functions to the data before the DFT^{-1} mitigates the effects of spectral leakage (called sidelobes) at the sacrifice of increased main lobe width resulting in blurred target scenes. There are many choices for window functions such as Hanning, Hamming, Taylor and Blackman-Harris to name a few, and each window has its own trade-off between sidelobe suppression and main lobe width [23]. In this research, we use the Hamming window when presenting Fourier processing examples.

The output of this processing produces a two-dimensional plot of target scene scattering intensity versus range and range-rate, and the plot is updated over time as the sliding

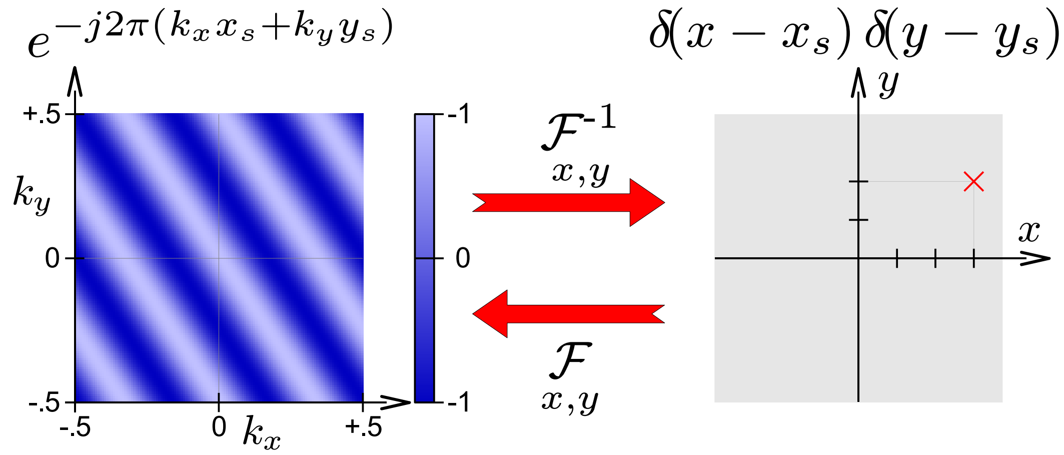


Figure 3.3: Two dimensional Fourier transform in k-space exchanges (k_x, k_y) with (x, y). In this example, the two-dimensional wavefront on the left undergoes 3 full periods on the k_x axis and 2 periods on the k_y axis, so its transform yields Dirac's delta function at $x = 3$, $y = 2$.

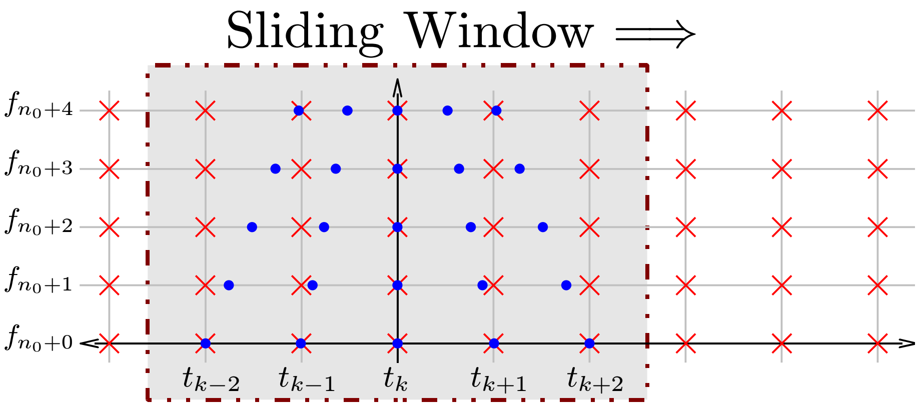


Figure 3.4: Fourier Doppler processing analyzes a block of interrogations at a time. Within a block, data is first resampled (indicated with blue circles) before further processing to analytically separate r_s and $\dot{r}_s$. This technique is also used for modern spectral processing algorithms discussed in the next section.

window advances. The animated plot is called an RDTI, which stands for *Range vs. Doppler vs. Time vs. Intensity*, and Fig. 3.5 shows several still-frames produced from our synthetic data of Chapter 2. Note how the two scatterers in Fig. 3.5 stay well resolved from one another even when they temporarily share the same range. In a simpler range-only Fourier plot such as the RTI of Fig. 2.3, an automated tracking system may have difficulty tracking scatterers in a complex target scene as they cross over one another.

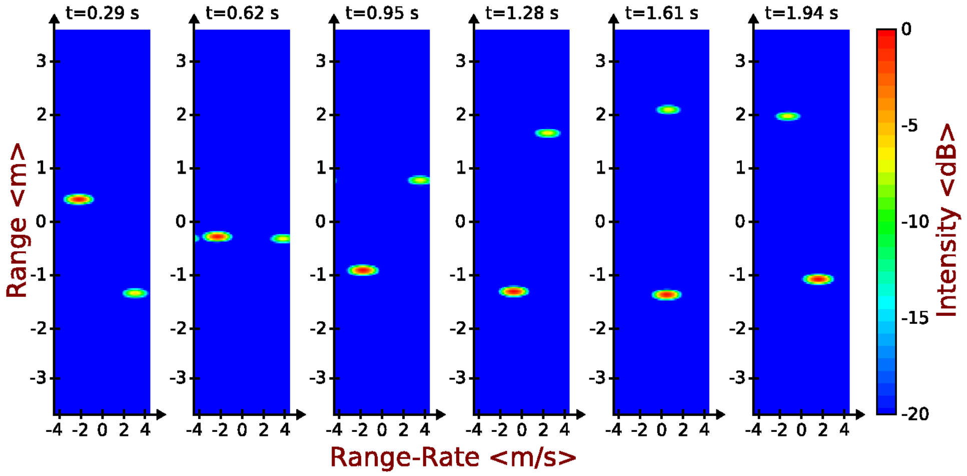


Figure 3.5: These are frames taken from an Range-Doppler-Time-Intensity animation of our synthetic data in Chapter 2. Each frame is produced from 10 interrogations by first applying Hamming windows across the frequency and time dimensions, then computing the two-dimensional inverse DFT, and plotting the resulting magnitudes.

Alternatively, Fig. 3.6 presents the range and range-rates as separate RTI and *DTI* (Doppler vs. Time vs. Intensity) plots. Each column of the DTI plot is formed by a one-dimensional inverse DFT applied to 10 interrogations at a time, along the center frequency. Since each DTI column is formed from only 10 samples across time at a single frequency, the resulting range-rate (Doppler) resolution is highly susceptible to noise spikes and suffers significant blurring as compared to the RTI. In fact, some of this blurring may also be observed in the RDTI still frames of Fig. 3.5 in the form of each scatterer appearing elliptical, that is, being more compact in the range dimension than in the range-rate dimension. However, the two-dimensional DFT employed for the RDTI frames is not as susceptible to noise spikes due to its utilization of consecutive interrogations to average out noise.

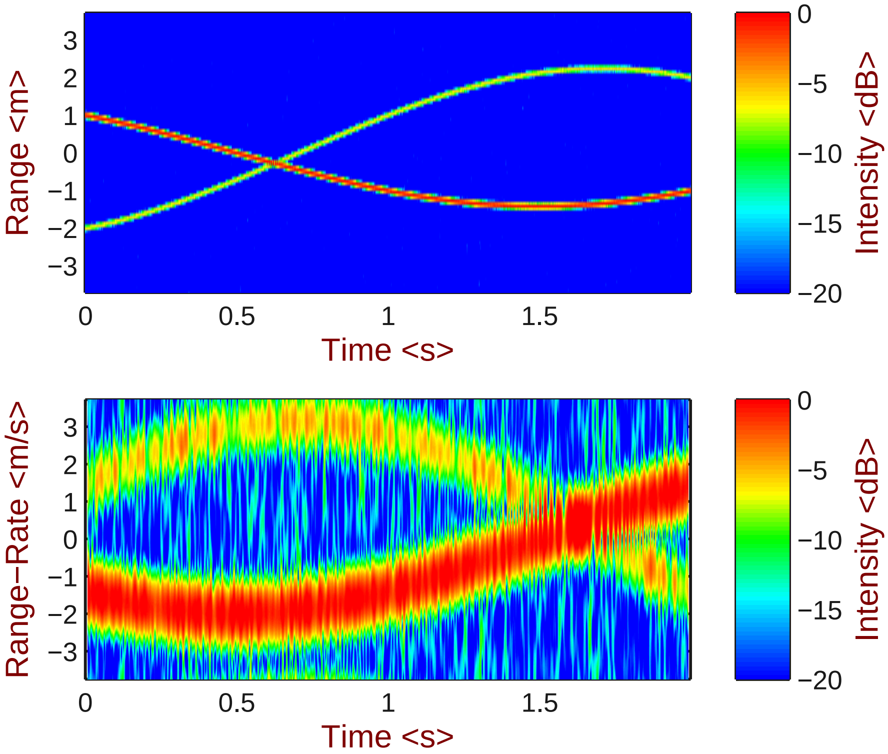


Figure 3.6: The upper image is the same RTI plot from Fig. 2.3. The lower image is called a *Doppler vs. Time vs. Intensity* (DTI) plot, and shows the range-rates of a target scene's scatterers as they change over time due to acceleration and deceleration.

We could have selected a wider block size to increase our range-rate resolution in Fig. 3.6, except doing so would incur more blurring of the velocity information about an uncooperative target with rapidly changing motion dynamics. The selection of block-size depends upon on a radar analyst's processing needs versus target dynamics. We will continue with this small block size to challenge the algorithms in the next section.

Fourier Doppler processing is very fast and may be processed in real-time with live radar data feeds to provide instantaneous diagnostic feedback for system operators. The downside to this approach is that range and range-rate estimates are confined to a grid, and if a scatterer fell between grid locations, its spectral energy is smeared into adjacent grid locations. As with the case of range-only Fourier processing, this spectral leakage degrades reflection coefficient estimates, and impairs ability to identify weak scatterers, or scatterers too close to one another [8]. A better approach would employ an algorithm without a fixed solution grid.

3.3 Joint-Doppler Direct State-Space System Identification

The methods presented here are intrinsically related to other modern multidimensional methods such as Multidimensional ESPRIT [50, 35, 8], Matrix-Enhanced Matrix-Pencil [26] and C-MIMO [45] with the chief difference being one of a vocabulary translation [47]. We will continue with the algebraic and state-space terminology from Chapter 2, except now we generalize to multidimensional data factoring and call this *Multidimensional Direct State-Space.* We shall formulate the derivations in a similar fashion to the range-only DSS method presented earlier, and start by demonstrating how Eq. 3.5 can be written in a form akin to that of a state-space impulse response:

$$\tilde{\tilde{g}}_{n,k} = \sum_s \Gamma_s e^{-j2\pi\left(\frac{2r_s}{c}n\Delta f + \frac{2f_0\dot{r}_s}{c}k\Delta t'\right)} \qquad \text{from Eq. 3.5} \tag{3.13}$$

$$= \underbrace{\begin{bmatrix}\Gamma_1 & \Gamma_2 & \ldots\end{bmatrix}}_{\mathrm{C}} \underbrace{\begin{bmatrix} e^{\frac{-j4\pi r_1\Delta f}{c}} & 0 & \ldots \\ 0 & e^{\frac{-j4\pi r_2\Delta f}{c}} & \\ \vdots & & \ddots \end{bmatrix}^n}_{\mathrm{A}_r^n} \underbrace{\begin{bmatrix} e^{\frac{-j4\pi f_0\dot{r}_1\Delta t'}{c}} & 0 & \ldots \\ 0 & e^{\frac{-j4\pi f_0\dot{r}_2\Delta t'}{c}} & \\ \vdots & & \ddots \end{bmatrix}^k}_{\mathrm{A}_{\dot{r}}^k} \underbrace{\begin{bmatrix}1\\1\\\vdots\end{bmatrix}}_{\mathrm{B}} \tag{3.14}$$

$$= \mathrm{C}\mathrm{A}_r^n\mathrm{A}_{\dot{r}}^k\mathrm{B} \tag{3.15}$$

Note that A_r and $\mathrm{A}_{\dot{r}}$ commute since they are diagonal matrices and will not lose their commutability under state-space system equivalence transformations discussed in Section 2.3.2.

Having set the stage, we will first describe the extension of observability processing introduced earlier to higher dimensional (range and range-rate) processing, and then follow with the multidimensional extension of balanced processing. It should be noted that while many variants of multidimensional observability processing exist in the signals processing realm, the multidimensional form of balanced processing has not been developed until the present research.

3.3.1 Multidimensional DSS Observability (or Controllability) Processing

The process described in this section bears many similarities to other established processes such as Multidimensional ESPRIT [50, 35, 8] and Coupled-MIMO [45]. We emphasize how closely related the established algorithms are by referring to this general description as *Multidimensional, DSS Observability Processing.* As a note, this method is not to be confused with the Multiple-Input Multiple-Output (MIMO) algorithms applied in control systems. The fundamental difference there is that the observed input and output data is sampled over a single dimension of time or frequency, but in the signals processing realm, there may be two or more time or frequency dimensions as in the case of time and k-space as depicted in Fig. 3.3.

We start this description by first forming a block Hankel matrix, H, over the two dimensions of $\tilde{\tilde{g}}_{n,k}$ in Eq. 3.16, and then we show that H factors into partitioned observability and controllability matrices given in Eq. 3.17. By block Hankel, we mean that each partition is a smaller Hankel matrix over index k for a fixed n, and on a larger scale, the n partitions themselves are organized in the same anti-diagonal pattern associated with Hankel matrices as well. For example, partition (1,1) of Eq. 3.16 is a Hankel matrix composed of $\tilde{\tilde{g}}_{0,k}\forall k$ ($\tilde{\tilde{g}}_{0,0}$, $\tilde{\tilde{g}}_{0,1}$, $\tilde{\tilde{g}}_{0,2}$, $\ldots$). Partitions (2,1) and (1,2) are Hankels of $\tilde{\tilde{g}}_{1,k}\forall k$, and partition (2,2) is a Hankel of $\tilde{\tilde{g}}_{2,k}\forall k$. For brevity, we have started our index numbers at 0 instead of n_0 and k_0, but the same factorization holds for (n_0, k_0) as well.

$$
\begin{aligned}
\mathrm{H} &\triangleq \underset{n,k}{\mathcal{H}}\left\{\tilde{\tilde{g}}_{n,k}\right\} \qquad (3.16)\\
&= \underset{n}{\mathcal{H}}\left\{\underset{k}{\mathcal{H}}\left\{\tilde{\tilde{g}}_{n,k}\right\}\right\}\\
&= \left[\begin{array}{c|c|c}
\mathcal{H}_k\left\{\tilde{\tilde{g}}_{0,k}\right\} & \mathcal{H}_k\left\{\tilde{\tilde{g}}_{1,k}\right\} & \ldots \\ \hline
\mathcal{H}_k\left\{\tilde{\tilde{g}}_{1,k}\right\} & \mathcal{H}_k\left\{\tilde{\tilde{g}}_{2,k}\right\} & \ldots \\ \hline
\vdots & \vdots & \ddots
\end{array}\right]
= \left[\begin{array}{ccc|ccc|c}
\tilde{\tilde{g}}_{0,0} & \tilde{\tilde{g}}_{1,0} & \cdots & \tilde{\tilde{g}}_{0,1} & \tilde{\tilde{g}}_{1,1} & \cdots & \\
\tilde{\tilde{g}}_{1,0} & \tilde{\tilde{g}}_{2,0} & \cdots & \tilde{\tilde{g}}_{1,1} & \tilde{\tilde{g}}_{2,1} & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \\ \hline
\tilde{\tilde{g}}_{0,1} & \tilde{\tilde{g}}_{1,1} & \cdots & \tilde{\tilde{g}}_{0,2} & \tilde{\tilde{g}}_{1,2} & \cdots & \\
\tilde{\tilde{g}}_{1,1} & \tilde{\tilde{g}}_{2,1} & \cdots & \tilde{\tilde{g}}_{1,2} & \tilde{\tilde{g}}_{2,2} & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \\ \hline
\multicolumn{3}{c|}{\vdots} & \multicolumn{3}{c|}{\vdots} & \ddots
\end{array}\right]\\
&= \left[\begin{array}{ccc|ccc|c}
\mathrm{CA}_r^0\mathrm{A}_r^0\mathrm{B} & \mathrm{CA}_r^1\mathrm{A}_r^0\mathrm{B} & \ldots & \mathrm{CA}_r^0\mathrm{A}_r^1\mathrm{B} & \mathrm{CA}_r^1\mathrm{A}_r^1\mathrm{B} & \ldots & \\
\mathrm{CA}_r^1\mathrm{A}_r^0\mathrm{B} & \mathrm{CA}_r^2\mathrm{A}_r^0\mathrm{B} & \ldots & \mathrm{CA}_r^1\mathrm{A}_r^1\mathrm{B} & \mathrm{CA}_r^2\mathrm{A}_r^1\mathrm{B} & \ldots & \ldots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \\ \hline
\mathrm{CA}_r^0\mathrm{A}_r^1\mathrm{B} & \mathrm{CA}_r^1\mathrm{A}_r^1\mathrm{B} & \ldots & \mathrm{CA}_r^0\mathrm{A}_r^2\mathrm{B} & \mathrm{CA}_r^1\mathrm{A}_r^2\mathrm{B} & \ldots & \\
\mathrm{CA}_r^1\mathrm{A}_r^1\mathrm{B} & \mathrm{CA}_r^2\mathrm{A}_r^1\mathrm{B} & \ldots & \mathrm{CA}_r^1\mathrm{A}_r^2\mathrm{B} & \mathrm{CA}_r^2\mathrm{A}_r^2\mathrm{B} & \ldots & \ldots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \\ \hline
\multicolumn{3}{c|}{\vdots} & \multicolumn{3}{c|}{\vdots} & \ddots
\end{array}\right]
\end{aligned}
$$

$$= \underbrace{\left[\begin{array}{c} \mathrm{CA}_r^0\mathrm{A}_{\dot{r}}^0 \\ \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^0 \\ \vdots \\ \hline \mathrm{CA}_r^0\mathrm{A}_{\dot{r}}^1 \\ \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^1 \\ \vdots \\ \hline \vdots \end{array}\right]}_{\mathcal{O}} \underbrace{\left[\begin{array}{c|c|c} \mathrm{A}_r^0\mathrm{A}_{\dot{r}}^0\mathrm{B} \;\; \mathrm{A}_r^1\mathrm{A}_{\dot{r}}^0\mathrm{B} \; \ldots & \mathrm{A}_r^0\mathrm{A}_{\dot{r}}^1\mathrm{B} \;\; \mathrm{A}_r^1\mathrm{A}_{\dot{r}}^1\mathrm{B} \; \ldots & \ldots \end{array}\right]}_{\mathcal{C}}$$

$$\mathrm{H} = \mathcal{O}\mathcal{C} \tag{3.17}$$

We accomplish this factorization using the SVD, and thus the resulting state representation will not be in our canonical form of Eq. 3.14, but will be related via an equivalence transformation as discussed earlier and in Appendix A.

$$
\begin{aligned}
\mathrm{H} &\overset{\text{svd}}{\Rightarrow} \mathrm{U}\Sigma\mathrm{V}^H && \qquad (3.18)\\
\mathrm{U}_{\mathrm{tr}} &\triangleq \mathrm{U}_{1:s,:} \qquad \text{first } s \text{ columns of U} && \qquad (3.19)\\
\Sigma_{\mathrm{tr}} &\triangleq \Sigma_{1:s,1:s} \qquad \text{first } s \text{ singular values of } \Sigma && \qquad (3.20)\\
\mathrm{V}_{\mathrm{tr}} &\triangleq \mathrm{V}_{1:s,:} \qquad \text{first } s \text{ columns of V} && \qquad (3.21)\\
\widetilde{\mathcal{O}} &\Leftarrow \mathrm{U}_{\mathrm{tr}}\sqrt{\Sigma}_{\mathrm{tr}} && \qquad (3.22)\\
\widetilde{\mathcal{C}} &\Leftarrow \sqrt{\Sigma_{\mathrm{tr}}}\mathrm{V}_{\mathrm{tr}}^H && \qquad (3.23)
\end{aligned}
$$

From this point on we will continue with $\widetilde{\mathcal{O}}$, though applying the same techniques to $\widetilde{\mathcal{C}}$ would produced an analytically identical procedure, except for all variables being transposed as noted in Section 2.3.2. Illustrated in Eq. 3.24, we form $\widetilde{\mathcal{O}}_{-n}$ by removing the last row from each partition of $\widetilde{\mathcal{O}}$ and $\widetilde{\mathcal{O}}_{+n}$ by removing the first row from each partition of $\widetilde{\mathcal{O}}$. At the same time, we also form $\widetilde{\mathcal{O}}_{-k}$ from $\widetilde{\mathcal{O}}$ by removing the entire last partition and $\widetilde{\mathcal{O}}_{+k}$ from $\widetilde{\mathcal{O}}$ by removing the first partition.

$$\widetilde{\mathcal{O}}_{-n} \triangleq \left[\begin{array}{c} \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{0}\widetilde{\mathrm{A}}_{\dot{r}}^{0} \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{1}\widetilde{\mathrm{A}}_{\dot{r}}^{0} \\ \vdots \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{N-2}\widetilde{\mathrm{A}}_{\dot{r}}^{0} \\ \hline \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{0}\widetilde{\mathrm{A}}_{\dot{r}}^{1} \\ \vdots \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{N-2}\widetilde{\mathrm{A}}_{\dot{r}}^{1} \\ \hline \vdots \\ \hline \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{0}\widetilde{\mathrm{A}}_{\dot{r}}^{K-1} \\ \vdots \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{N-2}\widetilde{\mathrm{A}}_{\dot{r}}^{K-1} \end{array}\right] \quad \widetilde{\mathcal{O}}_{+n} \triangleq \left[\begin{array}{c} \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{1}\widetilde{\mathrm{A}}_{\dot{r}}^{0} \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{2}\widetilde{\mathrm{A}}_{\dot{r}}^{0} \\ \vdots \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{N-1}\widetilde{\mathrm{A}}_{\dot{r}}^{0} \\ \hline \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{1}\widetilde{\mathrm{A}}_{\dot{r}}^{1} \\ \vdots \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{N-1}\widetilde{\mathrm{A}}_{\dot{r}}^{1} \\ \hline \vdots \\ \hline \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{1}\widetilde{\mathrm{A}}_{\dot{r}}^{K-1} \\ \vdots \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{N-1}\widetilde{\mathrm{A}}_{\dot{r}}^{K-1} \end{array}\right] \quad \widetilde{\mathcal{O}}_{-k} \triangleq \left[\begin{array}{c} \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{0}\widetilde{\mathrm{A}}_{\dot{r}}^{0} \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{1}\widetilde{\mathrm{A}}_{\dot{r}}^{0} \\ \vdots \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{N-1}\widetilde{\mathrm{A}}_{\dot{r}}^{0} \\ \hline \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{0}\widetilde{\mathrm{A}}_{\dot{r}}^{1} \\ \vdots \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{N-1}\widetilde{\mathrm{A}}_{\dot{r}}^{1} \\ \hline \vdots \\ \hline \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{0}\widetilde{\mathrm{A}}_{\dot{r}}^{K-2} \\ \vdots \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{N-1}\widetilde{\mathrm{A}}_{\dot{r}}^{K-2} \end{array}\right] \quad \widetilde{\mathcal{O}}_{+k} \triangleq \left[\begin{array}{c} \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{0}\widetilde{\mathrm{A}}_{\dot{r}}^{1} \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{1}\widetilde{\mathrm{A}}_{\dot{r}}^{1} \\ \vdots \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{N-1}\widetilde{\mathrm{A}}_{\dot{r}}^{1} \\ \hline \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{0}\widetilde{\mathrm{A}}_{\dot{r}}^{2} \\ \vdots \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{N-1}\widetilde{\mathrm{A}}_{\dot{r}}^{2} \\ \hline \vdots \\ \hline \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{0}\widetilde{\mathrm{A}}_{\dot{r}}^{K-1} \\ \vdots \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}_r^{N-1}\widetilde{\mathrm{A}}_{\dot{r}}^{K-1} \end{array}\right] \tag{3.24}$$

Inspecting these truncated versions of $\widetilde{\mathcal{O}}$ reveals that $\widetilde{\mathcal{O}}_{-n}$ and $\widetilde{\mathcal{O}}_{+n}$ are related via a multiplication by $\widetilde{\mathrm{A}}_r$ (Eq. 3.25), while $\widetilde{\mathcal{O}}_{-k}$ and $\widetilde{\mathcal{O}}_{+k}$ are related via $\widetilde{\mathrm{A}}_{\dot{r}}$ (Eq. 3.26).

$$\widetilde{\mathcal{O}}_{-n}\widetilde{\mathrm{A}}_r = \widetilde{\mathcal{O}}_{+n} \tag{3.25}$$

$$\widetilde{\mathcal{O}}_{-k}\widetilde{\mathrm{A}}_{\dot{r}} = \widetilde{\mathcal{O}}_{+k} \tag{3.26}$$

Similar to solving for $\widetilde{\mathrm{A}}$ in the one-dimensional processing of Section 2.3.2, we may solve for $\widetilde{\mathrm{A}}_r$ and $\widetilde{\mathrm{A}}_{\dot{r}}$ using Least-Squares (LS), Total-Least-Squares (TLS) [20] or Unitary-TLS [3] approaches [47, p. 295], though for brevity, we illustrate the LS technique here with application of a pseudo inverse:

$$\widetilde{\mathrm{A}}_r = \widetilde{\mathcal{O}}^{\dagger}_{-n}\widetilde{\mathcal{O}}_{+n} \tag{3.27}$$

$$\widetilde{\mathrm{A}}_{\dot{r}} = \widetilde{\mathcal{O}}^{\dagger}_{-k}\widetilde{\mathcal{O}}_{+k} \tag{3.28}$$

The next step is to find a set of eigenvectors, T, which diagonalizes both $\widetilde{\mathrm{A}}_r$ and $\widetilde{\mathrm{A}}_{\dot{r}}$ in a similarity transformation, shown in Eq. 3.29. There are methods to find such T such as the Simultaneous Schur Decomposition [22] and the Joint Schur Decomposition [1]. However, their iterations do not reliably converge, so researchers continue the quest for an optimal solution [50, 35].

$$\mathrm{A}_r \Leftarrow \mathrm{T}^{-1}\widetilde{\mathrm{A}}_r\mathrm{T} \tag{3.29}$$

$$\mathrm{A}_{\dot{r}} \Leftarrow \mathrm{T}^{-1}\widetilde{\mathrm{A}}_{\dot{r}}\mathrm{T} \tag{3.30}$$

Simple attempts to bypass solving for T by separately diagonalizing $\widetilde{\mathrm{A}}_r$ and $\widetilde{\mathrm{A}}_{\dot{r}}$ and pairing the resulting eigenvectors break down in the case of two scatterers temporarily crossing in range or range-rate such as frequently occurs for spinning or tumbling targets. As two or more scatterers cross in range, A_r would contain duplicate eigenvalues, thus causing an eigen decomposition to return generalized eigenvectors which make pairing with the eigenvectors of $\widetilde{\mathrm{A}}_{\dot{r}}$ difficult. In a complex target scene, many scatterers may temporarily share similar range or range-rates as the objects spin and tumble.

For purposes of implementation simplicity, the present research employs a technique described by Burrows [8], wherein he computes the eigenvectors from a weighted sum of $\widetilde{\mathrm{A}}_r$ and $\widetilde{\mathrm{A}}_{\dot{r}}$ as expressed in Eq. 3.33 in order to achieve automatic eigenvalue pairing. He notes that a "choice for the weights is $c_r = 1$ and $c_{\dot{r}} = 0$, but in the presence of noise a more accurate estimate of the eigenvalues is obtained if both matrices are brought to bear by making both weights nonzero." He also observes that the approach fails in the rare situation where the weighted eigenvalue sum for one scatterer happens to match that of another scatterer, but that the break down may be avoided easily by changing c_r and $c_{\dot{r}}$. In the present research, we set the coefficients to the inverse of the Frobenius norm of each matrix to give equal weighting to both $\widetilde{\mathrm{A}}_r$ and $\widetilde{\mathrm{A}}_{\dot{r}}$.

$$c_r = 1/\left\|\widetilde{\mathrm{A}}_r\right\|_F \tag{3.31}$$

$$c_{\dot{r}} = 1/\left\|\widetilde{\mathrm{A}}_{\dot{r}}\right\|_F \tag{3.32}$$

$$\left(c_r\widetilde{\mathrm{A}}_r + c_{\dot{r}}\widetilde{\mathrm{A}}_{\dot{r}}\right) \overset{\mathrm{eig}}{\Rightarrow} \mathrm{T}\Lambda\mathrm{T}^{-1} \tag{3.33}$$

After computing A_r and $\mathrm{A}_{\dot{r}}$, we set up an over-determined linear system to solve for the scatterers' reflection intensities, $\begin{bmatrix}\Gamma_1 & \Gamma_2 & \ldots\end{bmatrix}$, in C of Eq. 3.36.

$$\mathrm{B} \triangleq \begin{bmatrix}1 & 1 & \ldots\end{bmatrix}^T \tag{3.34}$$

$$\left[\begin{array}{ccc|ccc|c} \tilde{\hat{g}}_{0,0} & \tilde{\hat{g}}_{1,0} & \ldots & \tilde{\hat{g}}_{0,1} & \tilde{\hat{g}}_{1,1} & \ldots & \ldots \end{array}\right] = \mathrm{C}\left[\begin{array}{ccc|ccc|c} \mathrm{A}_r^0\mathrm{A}_{\dot{r}}^0\mathrm{B} & \mathrm{A}_r^1\mathrm{A}_{\dot{r}}^0\mathrm{B} & \ldots & \mathrm{A}_r^0\mathrm{A}_{\dot{r}}^1\mathrm{B} & \mathrm{A}_r^1\mathrm{A}_{\dot{r}}^1\mathrm{B} & \ldots & \ldots \end{array}\right] \tag{3.35}$$

$$\mathrm{C} = \left[\begin{array}{ccc|ccc|c} \tilde{\hat{g}}_{0,0} & \tilde{\hat{g}}_{1,0} & \ldots & \tilde{\hat{g}}_{0,1} & \tilde{\hat{g}}_{1,1} & \ldots & \ldots \end{array}\right]\left[\begin{array}{ccc|ccc|c} \mathrm{A}_r^0\mathrm{A}_{\dot{r}}^0\mathrm{B} & \mathrm{A}_r^1\mathrm{A}_{\dot{r}}^0\mathrm{B} & \ldots & \mathrm{A}_r^0\mathrm{A}_{\dot{r}}^1\mathrm{B} & \mathrm{A}_r^1\mathrm{A}_{\dot{r}}^1\mathrm{B} & \ldots & \ldots \end{array}\right]^\dagger \tag{3.36}$$

This section laid out a generalized description for a two-dimensional state-space algorithm to identify the ranges and range-rates of scatterers in an unknown target scene. In practice though, the need for three-dimensional and higher systems frequently arises in signals processing such as when we have a linear array of radar antennas and need to estimate the range, range-rate, and direction to each unknown scatterer. The algorithm given above is trivially extended to higher dimensional processing by further increasing the levels of nesting in the block Hankel of Eq. 3.16. Given a three-dimensional data set, say $Z_{u,v,w}$, the resulting block Hankel is defined by Eq. 3.37, which when factored, forms partitioned $\widetilde{\mathcal{O}}$ and $\widetilde{\mathcal{C}}$ matrices such that

$$\mathrm{H} \triangleq \underset{u,v,w}{\mathcal{H}}\{Z_{u,v,w}\} = \underset{u}{\mathcal{H}}\left\{\underset{v,w}{\mathcal{H}}\{Z_{u,v,w}\}\right\} = \underset{u}{\mathcal{H}}\left\{\underset{v}{\mathcal{H}}\left\{\underset{w}{\mathcal{H}}\{Z_{u,v,w}\}\right\}\right\} \tag{3.37}$$

$$\mathrm{H} \overset{\text{svd}}{\Rightarrow} \mathrm{U}\Sigma\mathrm{V}^{H} \tag{3.38}$$

$$s \triangleq \text{the number of large } \sigma_i \in \Sigma \tag{3.39}$$

$$\widetilde{\mathcal{O}} \Leftarrow \mathrm{U}_{1:s,:}\sqrt{\Sigma_{1:s,1:s}} \tag{3.40}$$

$$\widetilde{\mathcal{C}} \Leftarrow \sqrt{\Sigma_{1:s,1:s}}\mathrm{V}^{H}_{1:s,:} \tag{3.41}$$

$$\widetilde{\mathcal{O}}_{-u} \triangleq \dots \quad \widetilde{\mathcal{O}}_{+u} \triangleq \dots \quad \widetilde{\mathcal{O}}_{-v} \triangleq \dots \quad \widetilde{\mathcal{O}}_{+v} \triangleq \dots \quad \widetilde{\mathcal{O}}_{-w} \triangleq \dots \quad \widetilde{\mathcal{O}}_{+w} \triangleq \dots \tag{3.42}$$

$$\widetilde{\mathrm{A}}_{u} = \widetilde{\mathcal{O}}^{\dagger}_{-u}\widetilde{\mathcal{O}}_{+u} \qquad \widetilde{\mathrm{A}}_{v} = \widetilde{\mathcal{O}}^{\dagger}_{-v}\widetilde{\mathcal{O}}_{+v} \qquad \widetilde{\mathrm{A}}_{w} = \widetilde{\mathcal{O}}^{\dagger}_{-w}\widetilde{\mathcal{O}}_{+w} \tag{3.43}$$

$$\vdots$$

3.3.2 Multidimensional DSS Balanced Processing

The previous section laid out the generalized form of multidimensional processing via the observability matrix, $\mathcal{O}$, and is well established in literature. In this section, we present a new multidimensional extension of balanced processing which is the higher dimensional generalization to the Eigensystem Realization Algorithm [29] or Balanced Realization from Markov Parameters [12, p. 203]. We present this algorithm first two-dimensionally, to solve for our scatterer's ranges and range-rates, and then discuss the higher-dimensional processing at the end.

We start this formulation with the formation of a Hankel matrix, $\mathrm{H}_{0,0}$ in Eq. 3.44, and show that it factors into the observability and controllability matrices $\mathcal{O}$ and $\mathcal{C}$ in Eq. 3.45. The chief difference between this Hankel and that from observability processing (Eq. 3.16) is that this Hankel only uses the first $N-1$ and $K-1$ elements of $\tilde{\tilde{g}}_{n,k}$ instead of all N and K elements.

$$\mathrm{H}_{0,0} \overset{\triangle}{=} \underset{n\in[0..N-2],k\in[0..K-2]}{\mathcal{H}}\left\{\tilde{\tilde{g}}_{n,k}\right\} = \underset{k\in[0..K-2]}{\mathcal{H}}\left\{\underset{n\in[0..N-2]}{\mathcal{H}}\left\{\tilde{\tilde{g}}_{n,k}\right\}\right\} \tag{3.44}$$

$$= \left[\begin{array}{c|c|c} \mathcal{H}_{n\in[0..N-2]}\left\{\tilde{\tilde{g}}_{n,0}\right\} & \mathcal{H}_{n\in[0..N-2]}\left\{\tilde{\tilde{g}}_{n,1}\right\} & \cdots \\ \hline \mathcal{H}_{n\in[0..N-2]}\left\{\tilde{\tilde{g}}_{n,1}\right\} & \mathcal{H}_{n\in[0..N-2]}\left\{\tilde{\tilde{g}}_{n,2}\right\} & \cdots \\ \hline \vdots & \vdots & \ddots \end{array}\right] = \left[\begin{array}{ccc|ccc|c} \tilde{\tilde{g}}_{0,0} & \tilde{\tilde{g}}_{1,0} & \cdots & \tilde{\tilde{g}}_{0,1} & \tilde{\tilde{g}}_{1,1} & \cdots & \\ \tilde{\tilde{g}}_{1,0} & \tilde{\tilde{g}}_{2,0} & \cdots & \tilde{\tilde{g}}_{1,1} & \tilde{\tilde{g}}_{2,1} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \\ \hline \tilde{\tilde{g}}_{0,1} & \tilde{\tilde{g}}_{1,1} & \cdots & \tilde{\tilde{g}}_{0,2} & \tilde{\tilde{g}}_{1,2} & \cdots & \\ \tilde{\tilde{g}}_{1,1} & \tilde{\tilde{g}}_{2,1} & \cdots & \tilde{\tilde{g}}_{1,2} & \tilde{\tilde{g}}_{2,2} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \\ \hline & \vdots & & & \vdots & & \ddots \end{array}\right]$$

$$= \left[\begin{array}{ccc|ccc|c} \mathrm{CA}_r^0\mathrm{A}_{\dot{r}}^0\mathrm{B} & \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^0\mathrm{B} & \cdots & \mathrm{CA}_r^0\mathrm{A}_{\dot{r}}^1\mathrm{B} & \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^1\mathrm{B} & \cdots & \\ \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^0\mathrm{B} & \mathrm{CA}_r^2\mathrm{A}_{\dot{r}}^0\mathrm{B} & \cdots & \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^1\mathrm{B} & \mathrm{CA}_r^2\mathrm{A}_{\dot{r}}^1\mathrm{B} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \\ \hline \mathrm{CA}_r^0\mathrm{A}_{\dot{r}}^1\mathrm{B} & \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^1\mathrm{B} & \cdots & \mathrm{CA}_r^0\mathrm{A}_{\dot{r}}^2\mathrm{B} & \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^2\mathrm{B} & \cdots & \\ \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^1\mathrm{B} & \mathrm{CA}_r^2\mathrm{A}_{\dot{r}}^1\mathrm{B} & \cdots & \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^2\mathrm{B} & \mathrm{CA}_r^2\mathrm{A}_{\dot{r}}^2\mathrm{B} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \\ \hline & \vdots & & & \vdots & & \ddots \end{array}\right]$$

$$= \left[\begin{array}{c} \mathrm{CA}_r^0\mathrm{A}_{\dot{r}}^0 \\ \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^0 \\ \vdots \\ \hline \mathrm{CA}_r^0\mathrm{A}_{\dot{r}}^1 \\ \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^1 \\ \vdots \\ \hline \vdots \end{array}\right] \left[\begin{array}{ccc|ccc|c} \mathrm{A}_r^0\mathrm{A}_{\dot{r}}^0\mathrm{B} & \mathrm{A}_r^1\mathrm{A}_{\dot{r}}^0\mathrm{B} & \cdots & \mathrm{A}_r^0\mathrm{A}_{\dot{r}}^1\mathrm{B} & \mathrm{A}_r^1\mathrm{A}_{\dot{r}}^1\mathrm{B} & \cdots & \cdots \end{array}\right]$$

$$\mathrm{H}_{0,0} = \mathcal{O}\mathcal{C} \tag{3.45}$$

Then we form $\mathrm{H}_{1,0}$ which is similar to $\mathrm{H}_{0,0}$ except for shifting the index range of n by using $n \in [1..N-1]$ instead of $n \in [0..N-2]$. Eq. 3.46 shows that $\mathrm{H}_{1,0}$ factors into the same $\mathcal{O}$ and $\mathcal{C}$ as $\mathrm{H}_{0,0}$ with an additional A_r arising between.

$$\mathrm{H}_{1,0} \triangleq \underset{n\in[1..N-1],k\in[0..K-2]}{\mathcal{H}}\left\{\tilde{\hat{g}}_{n,k}\right\} = \underset{k\in[0..K-2]}{\mathcal{H}}\left\{\underset{n\in[1..N-1]}{\mathcal{H}}\left\{\tilde{\hat{g}}_{n,k}\right\}\right\} \tag{3.46}$$

$$= \left[\begin{array}{c|c|c} \mathcal{H}_{n\in[1..N-1]}\left\{\tilde{\hat{g}}_{n,0}\right\} & \mathcal{H}_{n\in[1..N-1]}\left\{\tilde{\hat{g}}_{n,1}\right\} & \cdots \\ \hline \mathcal{H}_{n\in[1..N-1]}\left\{\tilde{\hat{g}}_{n,1}\right\} & \mathcal{H}_{n\in[1..N-1]}\left\{\tilde{\hat{g}}_{n,2}\right\} & \cdots \\ \hline \vdots & \vdots & \ddots \end{array}\right] = \left[\begin{array}{ccc|ccc} \tilde{\hat{g}}_{1,0} & \tilde{\hat{g}}_{2,0} & \cdots & \tilde{\hat{g}}_{1,1} & \tilde{\hat{g}}_{2,1} & \cdots \\ \tilde{\hat{g}}_{2,0} & \tilde{\hat{g}}_{3,0} & \cdots & \tilde{\hat{g}}_{2,1} & \tilde{\hat{g}}_{3,1} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ \hline \tilde{\hat{g}}_{1,1} & \tilde{\hat{g}}_{2,1} & \cdots & \tilde{\hat{g}}_{1,2} & \tilde{\hat{g}}_{2,2} & \cdots \\ \tilde{\hat{g}}_{2,1} & \tilde{\hat{g}}_{3,1} & \cdots & \tilde{\hat{g}}_{2,2} & \tilde{\hat{g}}_{3,2} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \end{array}\right]$$

$$= \left[\begin{array}{ccc|ccc|c} \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^0\mathrm{B} & \mathrm{CA}_r^2\mathrm{A}_{\dot{r}}^0\mathrm{B} & \ldots & \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^1\mathrm{B} & \mathrm{CA}_r^2\mathrm{A}_{\dot{r}}^1\mathrm{B} & \ldots & \\ \mathrm{CA}_r^2\mathrm{A}_{\dot{r}}^0\mathrm{B} & \mathrm{CA}_r^3\mathrm{A}_{\dot{r}}^0\mathrm{B} & \ldots & \mathrm{CA}_r^2\mathrm{A}_{\dot{r}}^1\mathrm{B} & \mathrm{CA}_r^3\mathrm{A}_{\dot{r}}^1\mathrm{B} & \ldots & \ldots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \\ \hline \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^1\mathrm{B} & \mathrm{CA}_r^2\mathrm{A}_{\dot{r}}^1\mathrm{B} & \ldots & \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^2\mathrm{B} & \mathrm{CA}_r^2\mathrm{A}_{\dot{r}}^2\mathrm{B} & \ldots & \\ \mathrm{CA}_r^2\mathrm{A}_{\dot{r}}^1\mathrm{B} & \mathrm{CA}_r^3\mathrm{A}_{\dot{r}}^1\mathrm{B} & \ldots & \mathrm{CA}_r^2\mathrm{A}_{\dot{r}}^2\mathrm{B} & \mathrm{CA}_r^3\mathrm{A}_{\dot{r}}^2\mathrm{B} & \ldots & \ldots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \\ \hline & \vdots & & & \vdots & & \ddots \end{array}\right]$$

$$= \left[\begin{array}{c} \mathrm{CA}_r^0\mathrm{A}_{\dot{r}}^0 \\ \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^0 \\ \vdots \\ \hline \mathrm{CA}_r^0\mathrm{A}_{\dot{r}}^1 \\ \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^1 \\ \vdots \\ \hline \vdots \end{array}\right] \mathrm{A}_r \left[\begin{array}{ccc|ccc|c} \mathrm{A}_r^0\mathrm{A}_{\dot{r}}^0\mathrm{B} & \mathrm{A}_r^1\mathrm{A}_{\dot{r}}^0\mathrm{B} & \ldots & \mathrm{A}_r^0\mathrm{A}_{\dot{r}}^1\mathrm{B} & \mathrm{A}_r^1\mathrm{A}_{\dot{r}}^1\mathrm{B} & \ldots & \ldots \end{array}\right]$$

$$\mathrm{H}_{1,0} = \mathcal{O}\mathrm{A}_r\mathcal{C} \tag{3.47}$$

Repeating the same process again but this time shifting the Hankel in index k instead of n yields:

$$\mathrm{H}_{0,1} \triangleq \underset{n\in[0..N-1],k\in[1..K-1]}{\mathcal{H}}\left\{\tilde{\hat{g}}_{n,k}\right\} = \underset{k\in[1..K-1]}{\mathcal{H}}\left\{\underset{n\in[0..N-2]}{\mathcal{H}}\left\{\tilde{\hat{g}}_{n,k}\right\}\right\} \tag{3.48}$$

$$= \left[\begin{array}{c} \mathrm{CA}_r^0\mathrm{A}_{\dot{r}}^0 \\ \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^0 \\ \vdots \\ \hline \mathrm{CA}_r^0\mathrm{A}_{\dot{r}}^1 \\ \mathrm{CA}_r^1\mathrm{A}_{\dot{r}}^1 \\ \vdots \\ \hline \vdots \end{array}\right] \mathrm{A}_{\dot{r}} \left[\begin{array}{ccc|ccc|c} \mathrm{A}_r^0\mathrm{A}_{\dot{r}}^0\mathrm{B} & \mathrm{A}_r^1\mathrm{A}_{\dot{r}}^0\mathrm{B} & \ldots & \mathrm{A}_r^0\mathrm{A}_{\dot{r}}^1\mathrm{B} & \mathrm{A}_r^1\mathrm{A}_{\dot{r}}^1\mathrm{B} & \ldots & \ldots \end{array}\right]$$

$$\mathrm{H}_{0,1} = \mathcal{O}\mathrm{A}_{\dot{r}}\mathcal{C} \tag{3.49}$$

So by using the SVD to factor $\mathrm{H}_{0,0}$ into $\widetilde{\mathcal{O}}$ and $\widetilde{\mathcal{C}}$ as discussed in earlier sections, we may divide these matrices from $\mathrm{H}_{1,0}$ to isolate $\widetilde{\mathrm{A}}_r$, and from $\mathrm{H}_{0,1}$ to isolate $\widetilde{\mathrm{A}}_{\dot{r}}$. As with one-dimensional balanced processing, the pseudo inverses $\widetilde{\mathcal{O}}^{\dagger}$ and $\widetilde{\mathcal{C}}^{\dagger}$ may be easily obtained via Eqs. 2.64 and 2.65.

$$\mathrm{H}_{0,0} \overset{\text{svd}}{\Rightarrow} \mathrm{U}\Sigma\mathrm{V}^{H} \tag{3.50}$$

$$s \triangleq \text{the number of large } \sigma_i \in \Sigma \tag{3.51}$$

$$\widetilde{\mathcal{O}} \Leftarrow \mathrm{U}_{1:s,:}\sqrt{\Sigma_{1:s,1:s}} \tag{3.52}$$

$$\widetilde{\mathcal{C}} \Leftarrow \sqrt{\Sigma_{1:s,1:s}}\mathrm{V}^{H}_{1:s,:}\widetilde{\mathrm{A}}_r \qquad \Leftarrow \widetilde{\mathcal{O}}^{\dagger}\mathrm{H}_{1,0}\widetilde{\mathcal{C}}^{\dagger}\widetilde{\mathrm{A}}_{\dot{r}} \Leftarrow \widetilde{\mathcal{O}}^{\dagger}\mathrm{H}_{0,1}\widetilde{\mathcal{C}}^{\dagger} \tag{3.53}$$

Then using the joint eigen decomposition of Eq. 3.33, we find T, which diagonalizes both $\widetilde{\mathrm{A}}_r$ and $\widetilde{\mathrm{A}}_{\dot{r}}$ into A_r and $\mathrm{A}_{\dot{r}}$ for automated pairing of the range and range-rate estimates. Lastly, we use the over-determined linear system of Eq. 3.36 to identify the scatterers amplitudes, $\begin{bmatrix}\Gamma_1 & \Gamma_2 & \dots\end{bmatrix}$, in C.

3.3.3 Example Output

Like Fourier Doppler processing, these state-space algorithms are also applied to a subset of the interrogation data as delimited by a sliding window as depicted in Fig. 3.4. Fig. 3.7 shows example RTI and DTI plots from the example data set processed using MATLAB's spline function for the interpolation stage discussed in Section 3.1 and the state-space observability factorization of Section 3.3.1. In both figures, note the drastically reduced occurrence of false scatterers in the range-versus-time plot as compared to the range-only processing results of Fig. 2.5; this may be attributed to the algorithm being able to fuse multiple adjacent interrogations to average out noise.

Fig. 3.8 presents the results of using identical processing to that of Fig. 3.7, but using cubic interpolation instead of spline interpolation, and we observe significant degradation and artifacts when both scatterers are at their maximum absolute range-rates.

Fig. 3.9 shows the results of using spline interpolation with the balanced factorization approach described in Section 3.3.2. Here, we find that the results are very similar to those of observability processing, except that some of the interpolation artifacts are further suppressed compared to Fig. 3.7. We will withhold further performance comparisons between

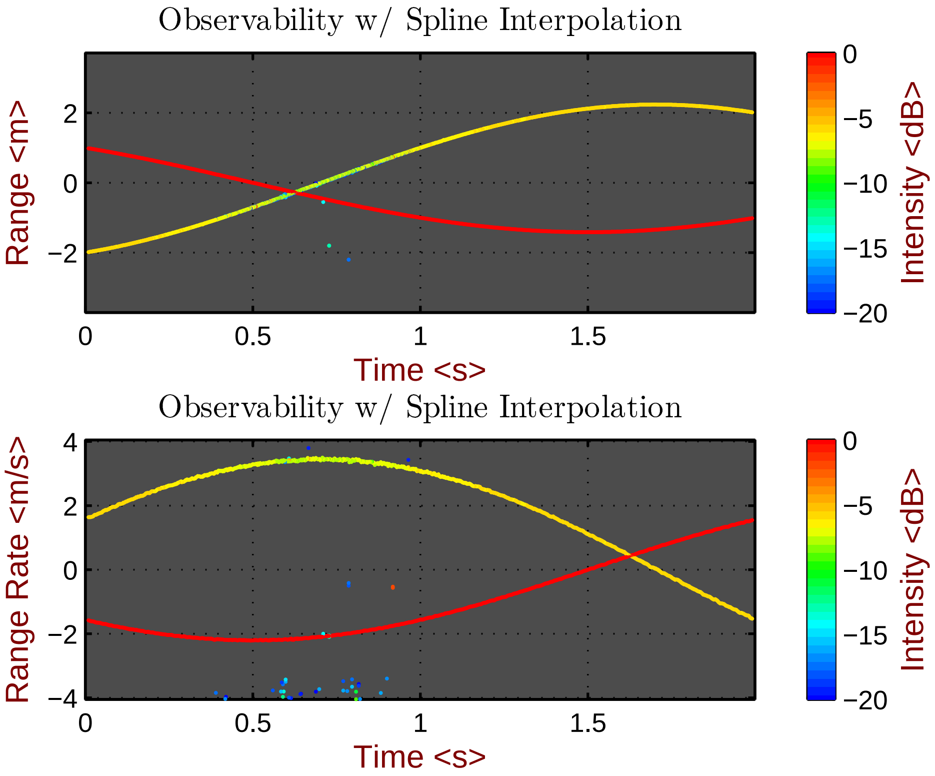


Figure 3.7: These plots of range and range-rate versus time were generated from two-dimensional state-space observability factorization of the synthetic data described in Chapter 2. The 20% bandwidth data was resampled with a spline interpolation as described Section 3.1.

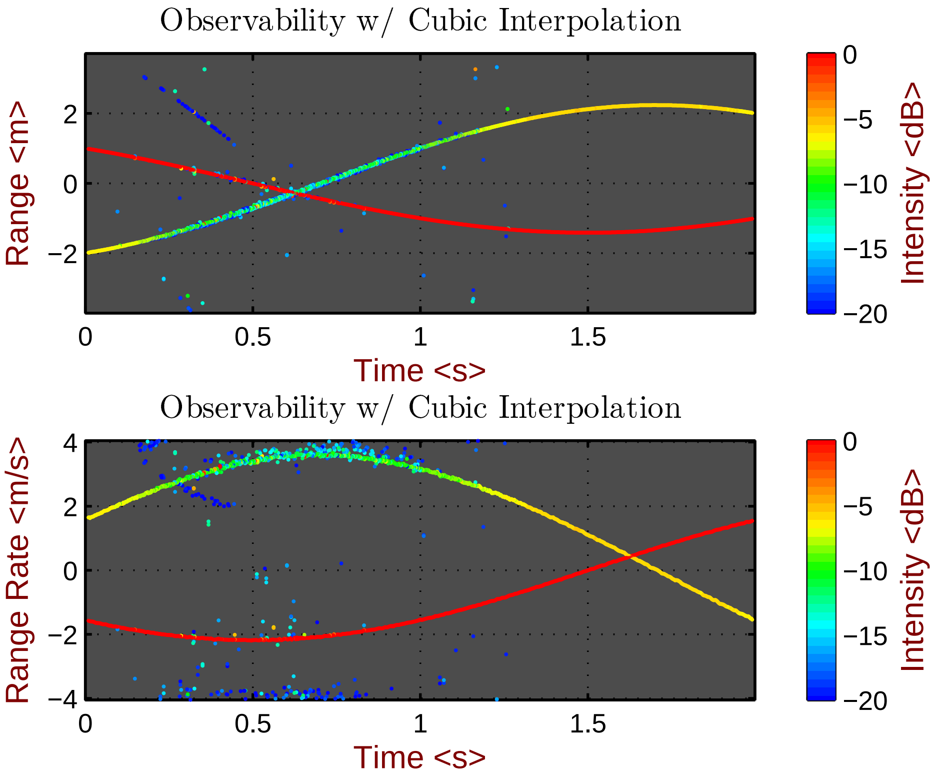


Figure 3.8: These plots of range and range-rate versus time were generated similarly to those of Fig 3.7, except that cubic interpolation was used instead of spline. Note the severe degradation of the scatterer having -6 dB reflection intensity.

the multidimensional observability and balanced factorizations until Chapter 4 but to say this apparent improvement was not substantiated under extensive evaluation.

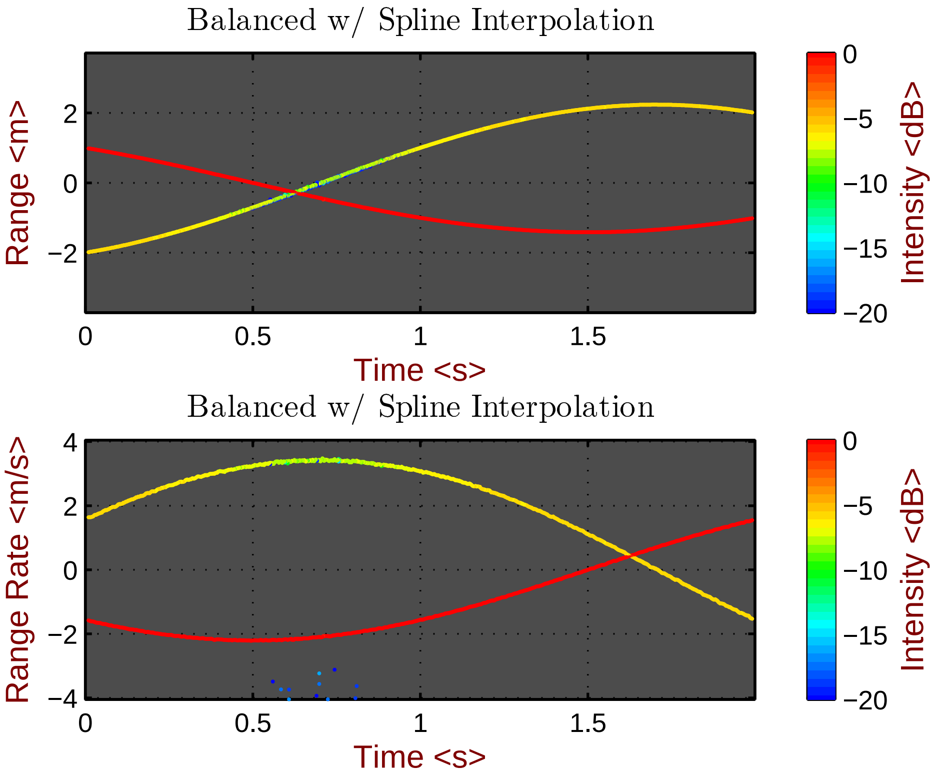


Figure 3.9: These plots of range and range-rate versus time were generated similarly to those of Fig 3.7, except that state-space balanced factorization was used instead of observability factorization. Note how some of the artifacts in Fig 3.7 are suppressed.

As a side note, *if* we had incorrectly assumed the data set contained sufficiently narrow bandwidth and skipped the interpolation stage, we would have obtained the plots shown in Fig. 3.10. The corruption seen while the scatterers have maximal absolute range-rate (during the time interval from 0.25 to 1.0 seconds) is directly caused by the algebraic model breaking down as it attempts to reconstitute the original input data from sums of complex exponentials.

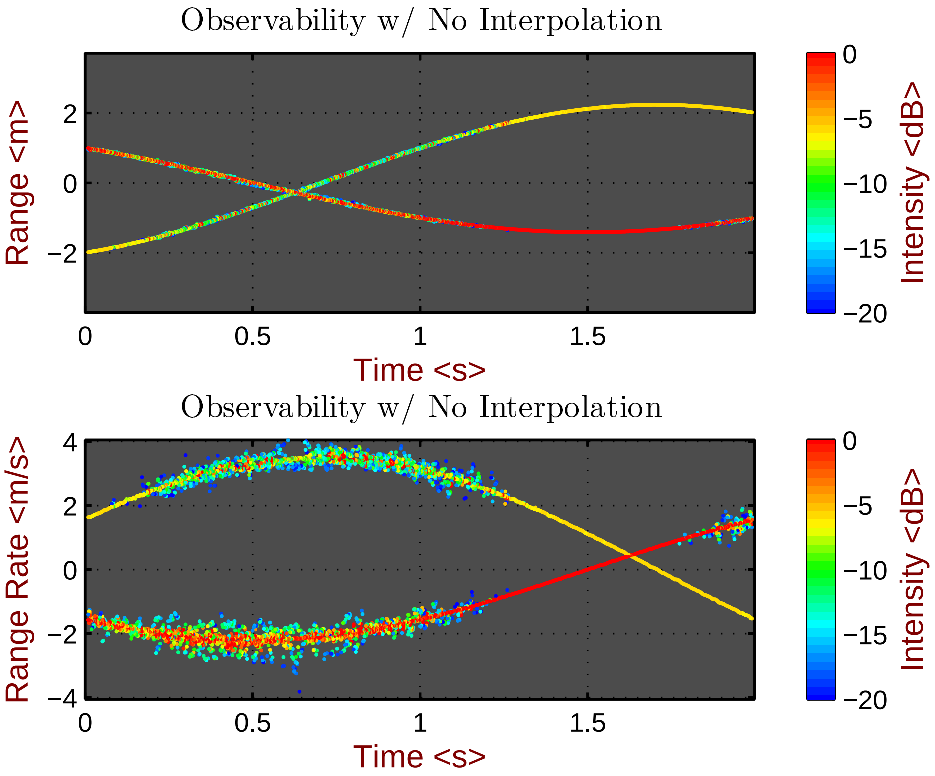


Figure 3.10: These range and range-rate versus time plots were generated without interpolating as is the assumption of narrow bandwidth algorithms. The 20% bandwidth data was purposefully not resampled to show an example of the corruption that occurs when narrow bandwidth algorithms are glibly applied to wide bandwidth data.

3.3.4 An Alternative to Interpolation

Referring back to Fig. 3.7, note the amplitude attenuation at 0.6 seconds of the scatterer that should ideally be found to have an amplitude of -6 dB in our synthetic data set, indicated by the greenish tint. Also during this time, several low-intensity artifacts appear clustered about a range-rate of -4 meters per second. We attribute these artifacts and amplitude attenuation to the break down of the spline interpolation process as the scatterer's range rate approaches the Nyquist sampling limit, which is $\pm\frac{c}{4f_0\Delta t} \approx \pm 4$ meters per second.

Instead of interpolating each row in the sliding data block of Figs. 3.1 and 3.4, we propose to fit a state-space model which should provide a closer fit to the input data than the generalized polynomials of spline and cubic interpolation. Each row of the data block contains the target scene's interrogation response for a single frequency, so we regroup the components of Eq. 3.3 to restate it in a form well modeled by a one-dimensional state-space technique, Eq. 3.54.

$$\begin{aligned}
\hat{g}_{n,k} &= \sum_s \underbrace{\Gamma_s e^{-j4\pi\frac{n\Delta f}{c} r_s}}_{\gamma_{n,s}} e^{-j4\pi\frac{n\Delta f k\Delta t}{c}\dot{r}_s} \qquad\qquad \text{restated from Eq. 3.3} \\
&= \sum_s \gamma_{n,s} e^{-j4\pi\frac{n\Delta f k\Delta t}{c}\dot{r}_s} \\
&= \underbrace{\begin{bmatrix} \gamma_{n,1} & \gamma_{n,2} & \cdots \end{bmatrix}}_{\mathrm{C}_n} \underbrace{\begin{bmatrix} e^{-j4\pi\frac{n\Delta f\Delta t}{c}\dot{r}_1} & 0 & \cdots \\ 0 & e^{-j4\pi\frac{n\Delta f\Delta t}{c}\dot{r}_2} & \\ \vdots & & \ddots \end{bmatrix}^k}_{\mathrm{A}_n^k} \underbrace{\begin{bmatrix} 1 \\ 1 \\ \vdots \end{bmatrix}}_{\mathrm{B}} \\
\hat{g}_{n,k} &= \mathrm{C}_n \mathrm{A}_n^k \mathrm{B} \qquad (3.54)
\end{aligned}$$

At each row, n, in the data block, $\hat{g}_{n,k}$, we solve for A_n and C_n using the balanced factorization state-space approach of Section 3.3.2. Next, we directly scale the angles of A_n's eigenvalues using Eq. 3.55 where f_0 is the same as in Eq. 3.4 in order to achieve the same separation of range and range-rate described in Section 3.1.

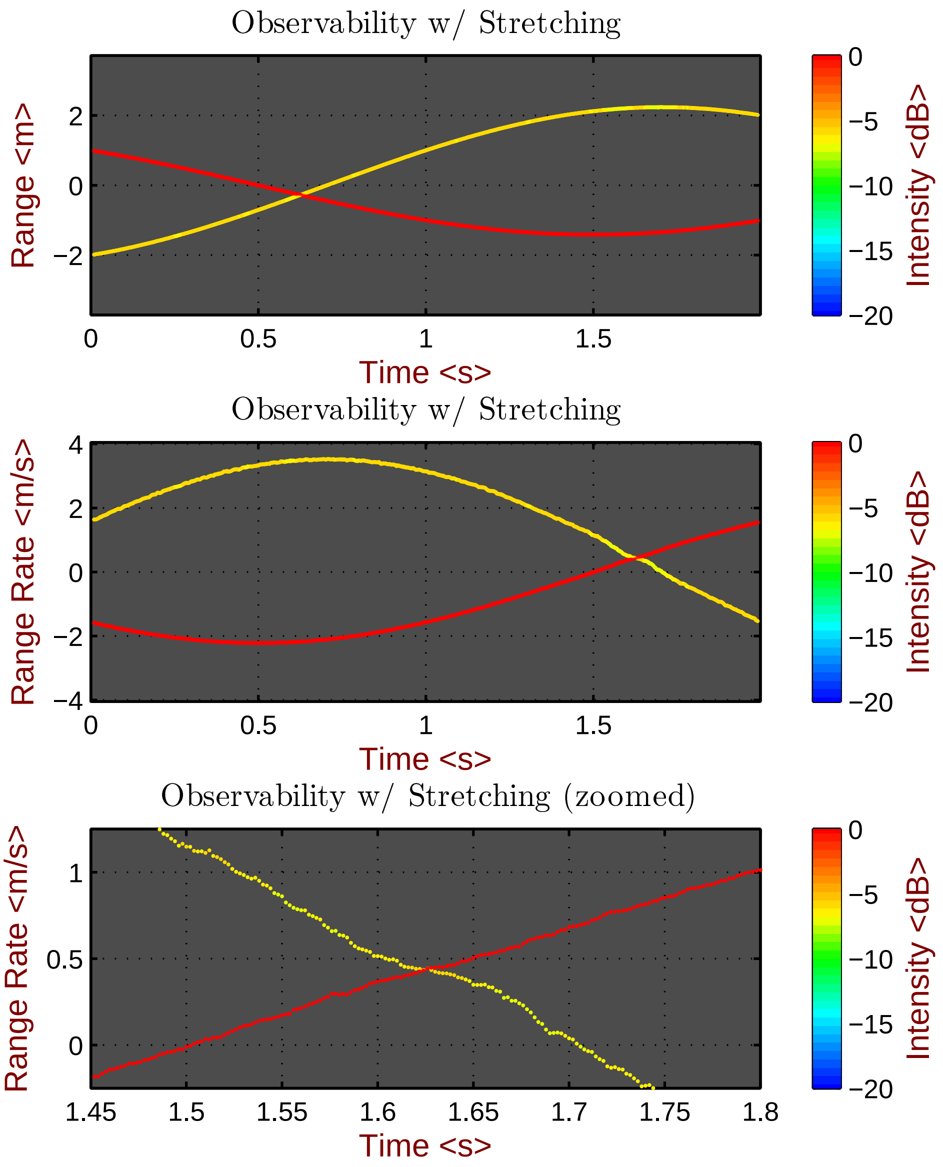
$$\begin{aligned}
\check{\mathrm{A}}_n &\Leftarrow \mathrm{A}_n^{\frac{f_0}{n\Delta f}} && (3.55)\\
&= \begin{bmatrix} e^{-j4\pi\frac{n\Delta f\Delta t}{c}\dot{r}_1} & 0 & \cdots \\ 0 & e^{-j4\pi\frac{n\Delta f\Delta t}{c}\dot{r}_2} & \\ \vdots & & \ddots \end{bmatrix}^{\frac{f_0}{n\Delta f}} \\
&= \begin{bmatrix} e^{-j4\pi\frac{f_0\Delta t}{c}\dot{r}_1} & 0 & \cdots \\ 0 & e^{-j4\pi\frac{f_0\Delta t}{c}\dot{r}_2} & \\ \vdots & & \ddots \end{bmatrix}
\end{aligned}$$

Lastly, we evaluate this system over k in Eq. 3.56 to reconstruct the input data, and we repeat this procedure for each n separately. We refer to this method as *stretch processing* to highlight how the state-space representation provides a means for the direct modification of the poles describing our input data.

$$\tilde{\hat{g}}_{n,k} = \mathrm{C}_n \check{\mathrm{A}}_n^{\,k} \mathrm{B} \qquad (3.56)$$

Fig. 3.11 presents the results of using stretch processing in conjunction with observability factorization of Section 3.3.1. Compared to the results in Fig. 3.7 where spline interpolation was used, stretch processing did not introduce artifacts when the scatterer's range-rates were near their maximums. However, we do find that stretch processing introduces deviations when the scatterers' range-rates were in close proximity. Unfortunately this is to be expected, because the one-dimensional state-space model fit at each frequency will break down as the eigenvalues representing the range-rates become closer together.

To fix this, we generalize stretch processing such that instead of processing each frequency independently, we process overlapping narrow (1 %BW) subbands using the two-dimensional balanced-factorization of Section 3.3.1. From each subband's representation (C, A_r, $\mathrm{A}_{\dot{r}}$, B), we scale the angles of $\mathrm{A}_{\dot{r}}$'s eigenvalues in Eq. 3.57, and then reevaluate the center frequency of each subband by iterating over k in Eq. 3.58.


Observability w/ Stretching

Observability w/ Stretching

Observability w/ Stretching (zoomed)

Figure 3.11: These range and range-rate plots versus time were generated using observability factorizations and *stretch processing* instead of spline or cubic interpolation. The bottom-most plot highlights the deviations that occur when the scatterers' range-rates were in close proximity.

$$\check{\mathrm{A}}\boldsymbol{r} \Leftarrow \mathrm{A}_{\dot{r}}{}^{\frac{f_0}{n\Delta f}} \tag{3.57}$$

$$\tilde{\hat{g}}_{n,k} = \mathrm{C}\mathrm{A}_r^n \check{\mathrm{A}}\boldsymbol{r}^k \mathrm{B} \tag{3.58}$$

In Fig. 3.12, we see the that the two-dimensional stretch processing no longer exhibits the deviations that occurred when the scatterers' range-rates were too close.

3.4 Linear Fast-Time Invariant (LFTI) DSS Systems

In this section, we introduce a unique approach to separating the range and range-rate components of Eq. 3.3 while simultaneously removing the simultaneous diagonalization step of Eq. 3.29 necessary for the typical multidimensional state-space methods. We will do this by restating our two-dimensional state-space problem of Section 3.3 as a one dimensional system, and to get started, we introduce *linear fast-time invariant* (LFTI) systems. In "fast-time invariant" systems, we assume the system or target scene is LTI during one interrogation, but between interrogations, the scatterer positions encoded within the state transition matrix (STM) may change significantly. In Eq. 3.59, we denote the STM as $\mathrm{A}(t)$ where (t) stresses that this matrix may change over time, but not during the interrogation period while the radar system is interrogating over the f_n frequencies.

$$\begin{aligned} \vec{x}_{n+1} &= \mathrm{A}(t)\vec{x}_n + \mathrm{B}u_n \\ y_n &= \mathrm{C}\vec{x}_n + \mathrm{D}u_n \end{aligned} \tag{3.59}$$

3.4.1 Algebraic Formulation

From Eq. 2.16 and Eq. 3.1, we further generalize r_s as an arbitrary function of time, $r_s(t)$ in Eq. 3.60, and write Eq. 3.61.

$$r_s(t) \triangleq r_s + \dot{r}_s t + \frac{1}{2}\ddot{r}_s t^2 + \dots \tag{3.60}$$

$$\hat{g}_n(t) = \sum_s \Gamma_s e^{-j4\pi\frac{n\Delta f}{c}\cdot r_s(t)} \tag{3.61}$$

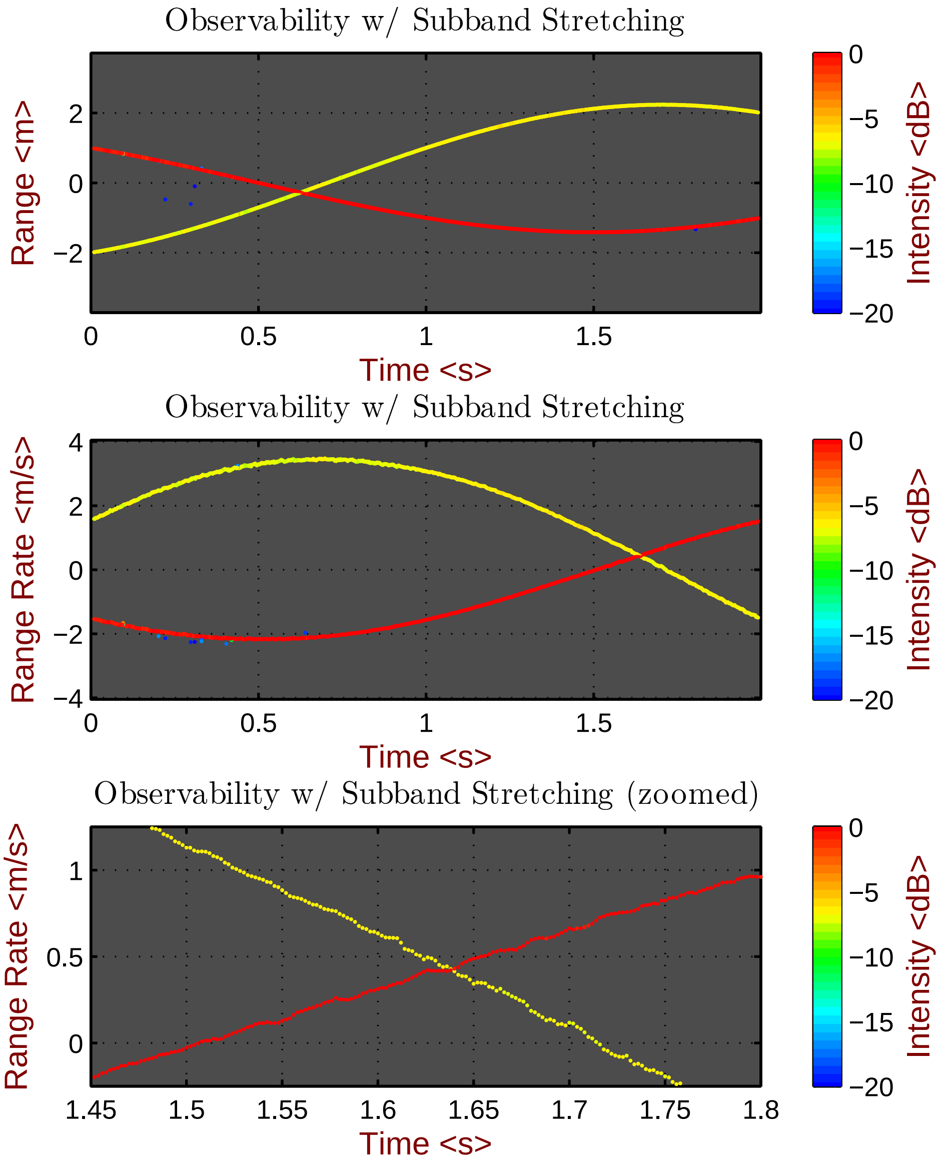


Figure 3.12: These range and range-rate plots versus time were generated using observability factorizations and *narrow-subband stretch processing* instead of interpolation. Compared to Fig. 3.11, the bottom-most plot here shows how the scatterers' range-rates no longer exhibit deviations when they are in close proximity.

For this formulation, we will initially take $r_s(t)$ to be $r_s + \dot{r}_s t$ excluding the second and higher derivative $(\frac{1}{2}\ddot{r}_s t^2 + \ldots)$ until later, and we start by showing that the resulting $\hat{g}_n(t)$ with its derivative with respect to time $(\frac{\partial}{\partial t}\hat{g}_n(t))$ may be written as the impulse response of a state-space system having two outputs, Eq. 3.66.

$$\hat{g}_n(t) = \sum_s \Gamma_s e^{-j4\pi\frac{n\Delta f}{c}\cdot(r_s+\dot{r}_s t)} \tag{3.62}$$

$$\frac{\partial}{\partial t}\hat{g}_n(t) = -j4\pi\frac{n\Delta f}{c}\sum_s \Gamma_s \dot{r}_s e^{-j4\pi\frac{n\Delta f}{c}\cdot(r_s+\dot{r}_s t)} \tag{3.63}$$

$$\begin{bmatrix} \hat{g}_n(t) \\ j\frac{c}{4\pi n\Delta f}\frac{\partial}{\partial t}\hat{g}_n(t) \end{bmatrix} = \sum_s \begin{bmatrix} \Gamma_s \\ \Gamma_s \dot{r}_s \end{bmatrix} e^{-j4\pi\frac{n\Delta f}{c}\cdot(r_s+\dot{r}_s t)} \tag{3.64}$$

$$\underbrace{\begin{bmatrix} \hat{g}_n(t) \\ j\frac{c}{4\pi n\Delta f}\frac{\partial}{\partial t}\hat{g}_n(t) \end{bmatrix}}_{\widehat{\mathrm{G}}_n(t)} = \underbrace{\begin{bmatrix} \Gamma_1 & \Gamma_2 & \ldots \\ \dot{r}_1\Gamma_1 & \dot{r}_2\Gamma_2 & \ldots \end{bmatrix}}_{\mathrm{C}} \underbrace{\begin{bmatrix} e^{-j4\pi\frac{\Delta f}{c}\cdot(r_1+\dot{r}_1 t)} & 0 & \ldots \\ 0 & e^{-j4\pi\frac{\Delta f}{c}\cdot(r_2+\dot{r}_2 t)} & \\ \vdots & & \ddots \end{bmatrix}^n}_{(\mathrm{A}(t))^n} \underbrace{\begin{bmatrix} 1 \\ 1 \\ \vdots \end{bmatrix}}_{\mathrm{B}} \tag{3.65}$$

$$\widehat{\mathrm{G}}_n(t) = \mathrm{C}(\mathrm{A}(t))^n\mathrm{B} \tag{3.66}$$

Given that Eq. 3.65 and Eq. 3.66 are nearly identical to our one-dimensional (range only) system defined earlier in Eq. 2.21 and Eq. 2.22, we may use either the balanced or observability DSS system identification methods defined in Section 2.3.2 to identify r_s and $\dot{r}_s$. The principal difference is that instead of using the individual samples $\hat{g}_n$ like before, we compose the Hankel matrices from 2-element vectors $\widehat{\mathrm{G}}_n(t)$ of the raw data $\hat{g}_n(t)$ with the time-derivative and scaled data, $j\frac{c}{4\pi n\cdot\Delta f}\frac{\partial}{\partial t}\hat{g}_n(t)$:

$$\widehat{\mathrm{G}}_n(t) \triangleq \begin{bmatrix} \hat{g}_n(t) \\ j\frac{c}{4\pi n\cdot\Delta f}\frac{\partial}{\partial t}\hat{g}_n(t) \end{bmatrix} \tag{3.67}$$

$$\mathrm{H}_0 \triangleq \begin{bmatrix} \widehat{\mathrm{G}}_0 & \widehat{\mathrm{G}}_1 & \ldots \\ \widehat{\mathrm{G}}_1 & \widehat{\mathrm{G}}_2 & \\ \vdots & & \ddots \end{bmatrix} \qquad \mathrm{H}_1 \triangleq \begin{bmatrix} \widehat{\mathrm{G}}_1 & \widehat{\mathrm{G}}_2 & \ldots \\ \widehat{\mathrm{G}}_2 & \widehat{\mathrm{G}}_3 & \\ \vdots & & \ddots \end{bmatrix} \tag{3.68}$$

Next we count the number of larger singular values in Σ and form $\widetilde{\mathcal{O}}$ and $\widetilde{\mathcal{C}}$ from the truncated U_{tr}, Σ_{tr} and V_{tr} matrices:

$$\mathrm{H}_0 \overset{\mathrm{svd}}{\Rrightarrow} \mathrm{U}\Sigma\mathrm{V}^H \tag{3.69}$$

$$\mathrm{U}_{\mathrm{tr}} \triangleq \mathrm{U}_{1:s,:} \qquad \text{first } s \text{ columns of U} \tag{3.70}$$

$$\Sigma_{\mathrm{tr}} \triangleq \Sigma_{1:s,1:s} \qquad \text{first } s \text{ singular values of } \Sigma \tag{3.71}$$

$$\mathrm{V}_{\mathrm{tr}} \triangleq \mathrm{V}_{1:s,:} \qquad \text{first } s \text{ columns of V} \tag{3.72}$$

$$\widetilde{\mathcal{O}} \Leftarrow \mathrm{U}_{\mathrm{tr}}\sqrt{\Sigma}_{\mathrm{tr}} \tag{3.73}$$

$$\widetilde{\mathcal{C}} \Leftarrow \sqrt{\Sigma_{\mathrm{tr}}}\mathrm{V}_{\mathrm{tr}}^H \tag{3.74}$$

Dividing $\widetilde{\mathcal{O}}$ and $\widetilde{\mathcal{C}}$ from H_1 yields $\widetilde{\mathrm{A}}$ in Eq. 3.75 which is eigen decomposed (Eq. 3.76) to reveal our canonical form first defined in Eq. 3.65 and reprinted in Eq. 3.77 for clarity.

$$\widetilde{\mathrm{A}} \Leftarrow \widetilde{\mathcal{O}}^\dagger \mathrm{H}_1 \widetilde{\mathcal{C}}^\dagger \tag{3.75}$$

$$\widetilde{\mathrm{A}} \overset{\mathrm{eig}}{\Rrightarrow} \mathrm{TAT}^{-1} \tag{3.76}$$

$$\mathrm{A}(t) = \begin{bmatrix} e^{-j4\pi\frac{\Delta f}{c}\cdot(r_1+\dot{r}_1 t)} & 0 & \cdots \\ 0 & e^{-j4\pi\frac{\Delta f}{c}\cdot(r_2+\dot{r}_2 t)} & \\ \vdots & & \ddots \end{bmatrix} \tag{3.77}$$

To isolate range-rates and reflection coefficients, we can also reuse our least-squares solver; solving with range-rates is the same as our previous, range-only problem. Given $\mathrm{B} = \begin{bmatrix}1 & 1 & \ldots\end{bmatrix}^T$,

$$\widehat{\mathrm{G}}_n = \mathrm{CA}^n\mathrm{B} \tag{3.78}$$

$$\begin{bmatrix}\widehat{\mathrm{G}}_0 & \widehat{\mathrm{G}}_1 & \ldots\end{bmatrix} = \mathrm{C}\begin{bmatrix}\mathrm{A}^0\mathrm{B} & \mathrm{A}^1\mathrm{B} & \ldots\end{bmatrix} \tag{3.79}$$

$$\mathrm{C} = \begin{bmatrix}\widehat{\mathrm{G}}_0 & \widehat{\mathrm{G}}_1 & \ldots\end{bmatrix}\begin{bmatrix}\mathrm{A}^0\mathrm{B} & \mathrm{A}^1\mathrm{B} & \ldots\end{bmatrix}^\dagger \tag{3.80}$$

$$\triangleq \begin{bmatrix}\Gamma_1 & \Gamma_2 & \ldots \\ \dot{r}_1\Gamma_1 & \dot{r}_2\Gamma_2 & \ldots\end{bmatrix} \tag{3.81}$$

The remaining solution step is the estimation of the time-derivative of each sample point in the data, and to do this, we employ a similar approach to that of stretch processing presented in Section 3.3.4. For each sample point measured at a given frequency within an interrogation, we use the samples from adjacent interrogations, specified by our block size, to fit a one-dimensional state-space model. Then we compute the derivative with respect to time of the model as shown in Eq. 3.82.

$$\begin{aligned}
\hat{g}_{n,k} &= \sum_s \underbrace{\Gamma_s e^{-j4\pi \frac{n\Delta f}{c} r_s}}_{\gamma_{n,s}} e^{-j4\pi \frac{n\Delta f k \Delta t}{c} \dot{r}_s} \qquad \text{restated from Eq. 3.3} \\
&= \sum_s \gamma_{n,s} e^{-j4\pi \frac{n\Delta f k \Delta t}{c} \dot{r}_s} \\
&= \underbrace{\begin{bmatrix} \gamma_{n,1} & \gamma_{n,2} & \dots \end{bmatrix}}_{\mathrm{C}_n} \underbrace{\begin{bmatrix} e^{-j4\pi \frac{n\Delta f \Delta t}{c} \dot{r}_1} & 0 & \dots \\ 0 & e^{-j4\pi \frac{n\Delta f \Delta t}{c} \dot{r}_2} & \\ \vdots & & \ddots \end{bmatrix}^k}_{\mathrm{A}_n^k} \underbrace{\begin{bmatrix} 1 \\ 1 \\ \vdots \end{bmatrix}}_{\mathrm{B}} \\
\hat{g}_{n,k} &= \mathrm{C}_n \mathrm{A}_n^k \mathrm{B} \\
\frac{\partial}{\partial t} \hat{g}_{n,k} &= \mathrm{C}_n \left(\frac{1}{\Delta t} \ln \mathrm{A}_n \right) \mathrm{A}_n^k \mathrm{B} \qquad \text{recall } t = k\Delta t
\end{aligned} \tag{3.82}$$

3.4.2 Example Output

We present the results of LFTI state-space processing in Fig. 3.13, and we find that while the range-rate estimation has a higher variance compared to the two-dimensional state-space models of Section 3.3, LFTI successfully produces automatically paired range and range-rate solution components without the need for interpolation or simultaneous eigen decomposition as required by those methods.

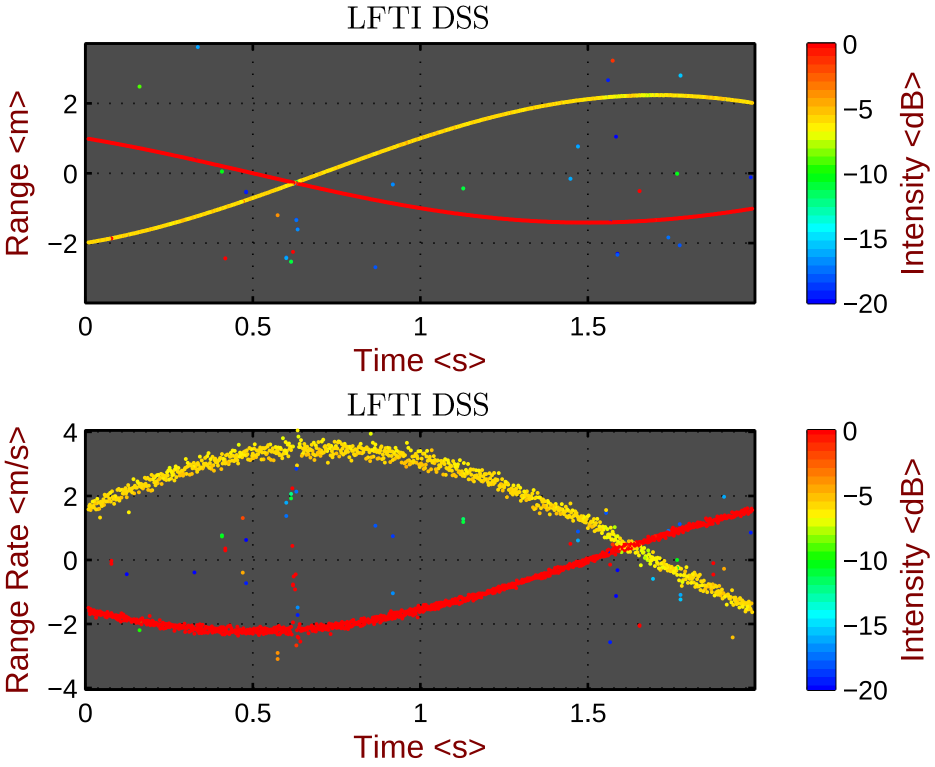


Figure 3.13: These range and range-rate plots versus time were generated using LFTI state-space processing.

Chapter 4

Method Validation

In this chapter, we evaluate the estimation performance of the algorithms presented in Chapter 3 with a series of three tests. The first test evaluates the accuracy of each algorithm when attempting to identify a single scatterer in a target scene, and the other two tests evaluate algorithm performance when a scatterer of interest is in close proximity to a second, interfering scatterer.

In all tests, these parameters are constant unless otherwise noted:

Center-Frequency	$f_c = 10$ GHz
Bandwidth	$f_{\text{BW}} = 2$ GHz
Frequency Sampling	$\Delta f = 20$ MHz
Interrogation Rate	$f_{\text{PRF}} = 500$ GHz ($\Delta t = 2$ ms)
Window Size	11 Interrogations (22 ms)

Also in each test, the algorithms are not given *a priori* knowledge as to the number of scatterers and must instead rely upon the heuristic described earlier in Eq. 2.40. Therefore, an algorithm may identify more than the expected one or two (truth) scatterers actually present in a test, but to keep computational run-time low, we limit the maximum number to 10 potential scatterers. From the potential scatterers, the error metrics are computed from the scatterer with the smallest 2-norm error in reflection magnitude, range, and range-rate. Of the smallest 2-norm error scatterers, any which fall outside of three standard deviations away from the mean error are identified as outliers and removed before the error calculation. This approach follows the common practice of using methods such as Kalman [31, 15] or particle filters [44] to post-process the estimates to remove spurious scatterers.

Fig. 4.1 shows one example error data set from where three spurious scatterer are identified with red X's, because their error values were at least three standard deviations away from the mean error for either their reflection coefficient, range, or range-rate attributes. Reflection phase error was not used for outlier classification due to the naturally high variance as identified in Appendix B when deriving the theoretical limit of estimation performance, the Cramer-Rao bound (CRB).

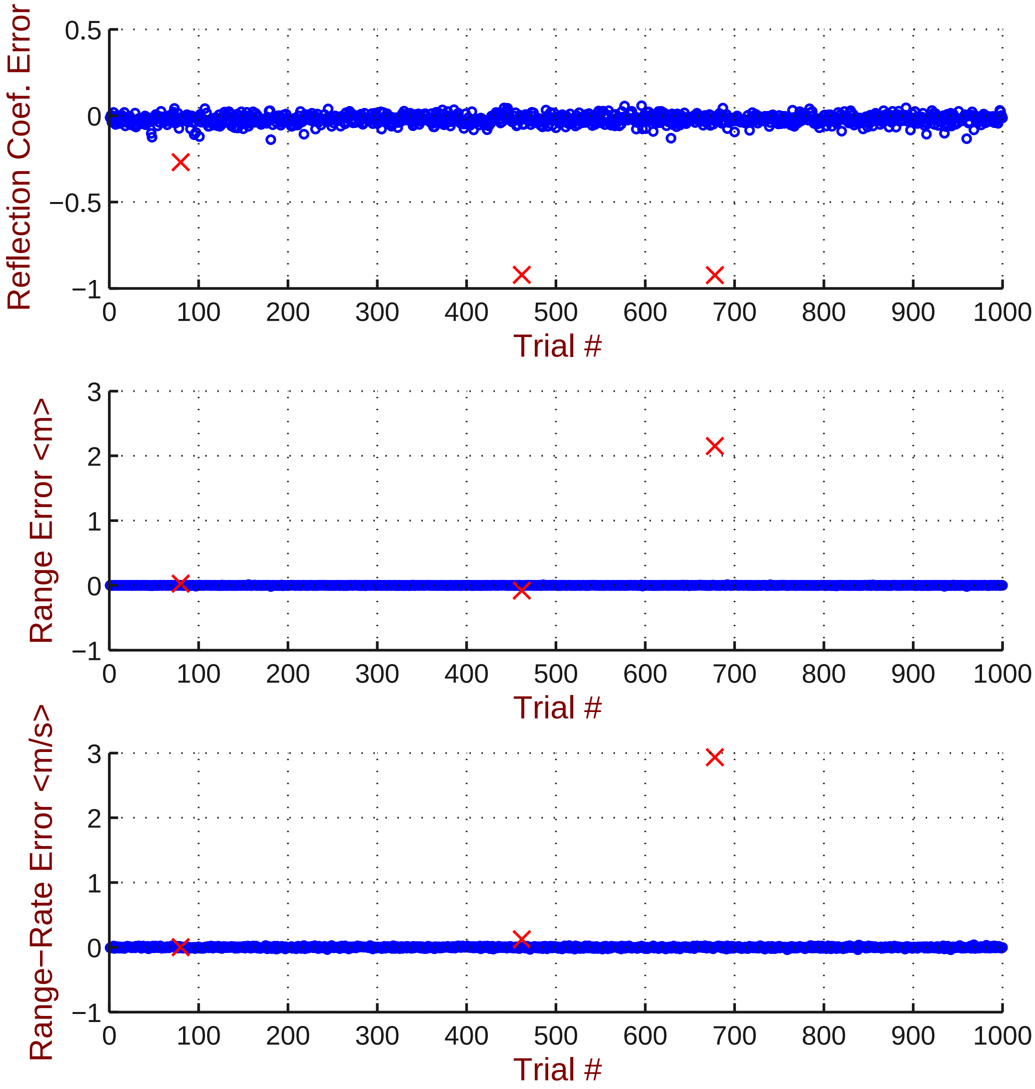


Figure 4.1: Raw error data from a 1000-trial test of the Observability method with stretch processing. The scatterers identified with red X's were classified as outliers.

4.1 One Scatterer

The purpose of this test is to measure the estimation error variance of each algorithm when identifying a target scene having a single scatterer. For a given SNR, which is varied from -9 dB to 12 dB, each algorithm processes a set of 1000 target scenes where the single scatterer of each scene has a constant reflection magnitude of 0 dB but a random range, range-rate and reflection phase offset. The root-mean-square (RMS) of the estimation error is taken over the 1000 trials for each of the scatterer's parameters—reflection magnitude, reflection phase offset, range, and range-rate—and presented in Figs. 4.2–4.5, respectively. In each figure, the CRB is plotted alongside the algorithms to serve as a benchmark for the best possible variance of any unbiased estimation algorithm. The derivation for this bound is given in Appendix B.

In Fig. 4.2 and Fig. 4.3, we present the RMS errors of each algorithm for estimating reflection magnitude and phase offset, and we note that the interpolation method appears to have more influence on estimation performance than does the option between observability and balanced processing methods. The methods employing our stretch processing approach the Cramer-Rao bound for magnitude estimation performance for SNR above 3 dB, but for poorer SNR, stretch processing performance falls back to that of LFTI. Also, we note that LFTI appears to have reduced estimation errors than the spline and cubic interpolation methods for higher SNR, and also that our subband stretch processing performed worse than cubic interpolation.

Fig. 4.4 presents the range estimation RMS errors, and we again observe the same trends that stretch processing tends to perform best, subband stretching performs worst, and LFTI performs favorably with respect to the spline or cubic interpolation methods for SNR above 3 dB. However in, Fig. 4.5, we find that LFTI's range-rate estimation performance sharply lags behind the other algorithms, and stretch processing has a clear lead.

In all four plots, we attribute subband stretch's notably poor performance to its reliance on a narrow-band assumption, albeit a 1% BW assumption, which would traditionally be considered very acceptable. We also note that LFTI processing compares favorably to the spline and cubic interpolation methods except for its range-rate estimates, but overall, the

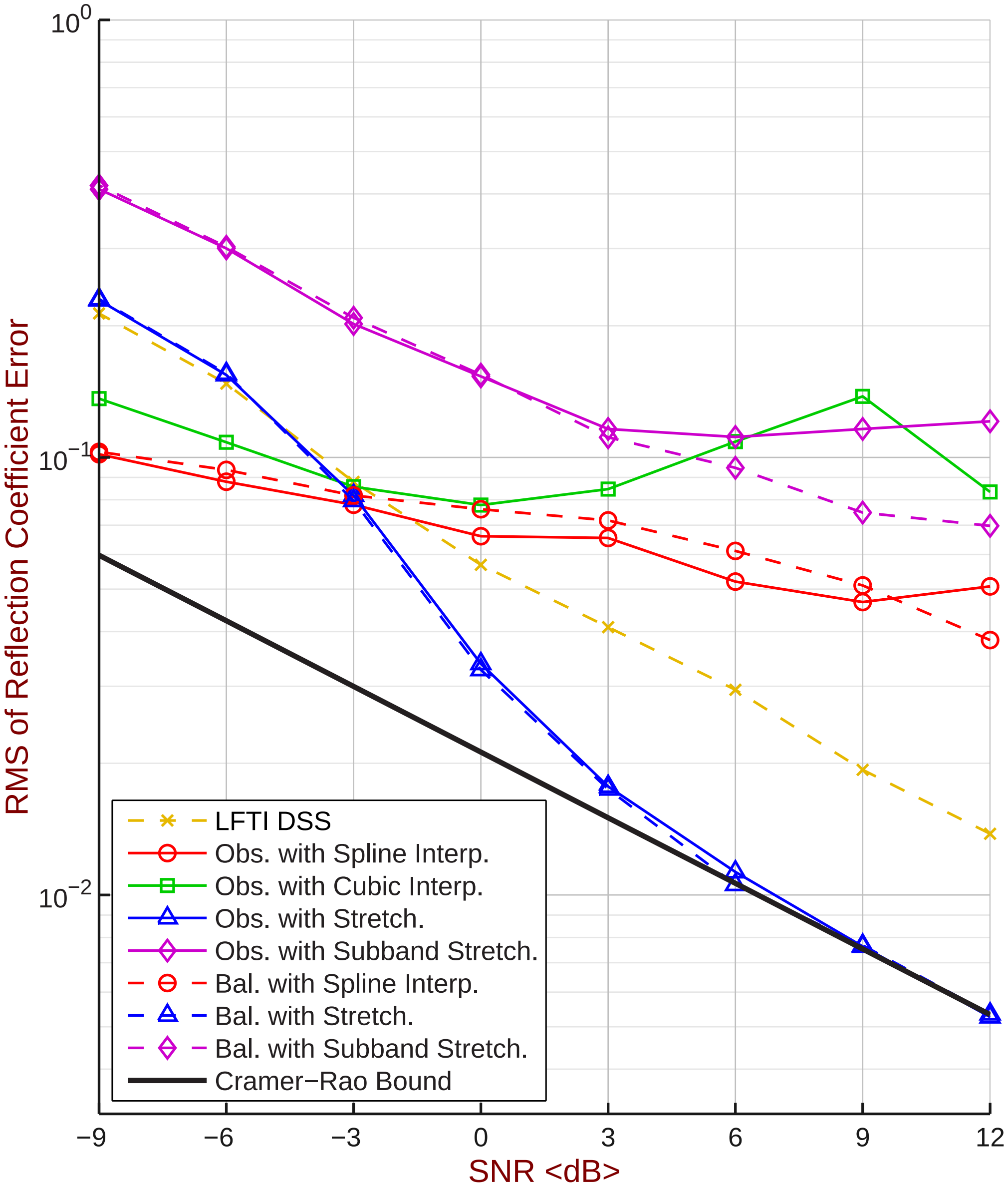


Figure 4.2: Comparison of RMS error of estimating a scatterer's reflection magnitude versus SNR for different algorithms.

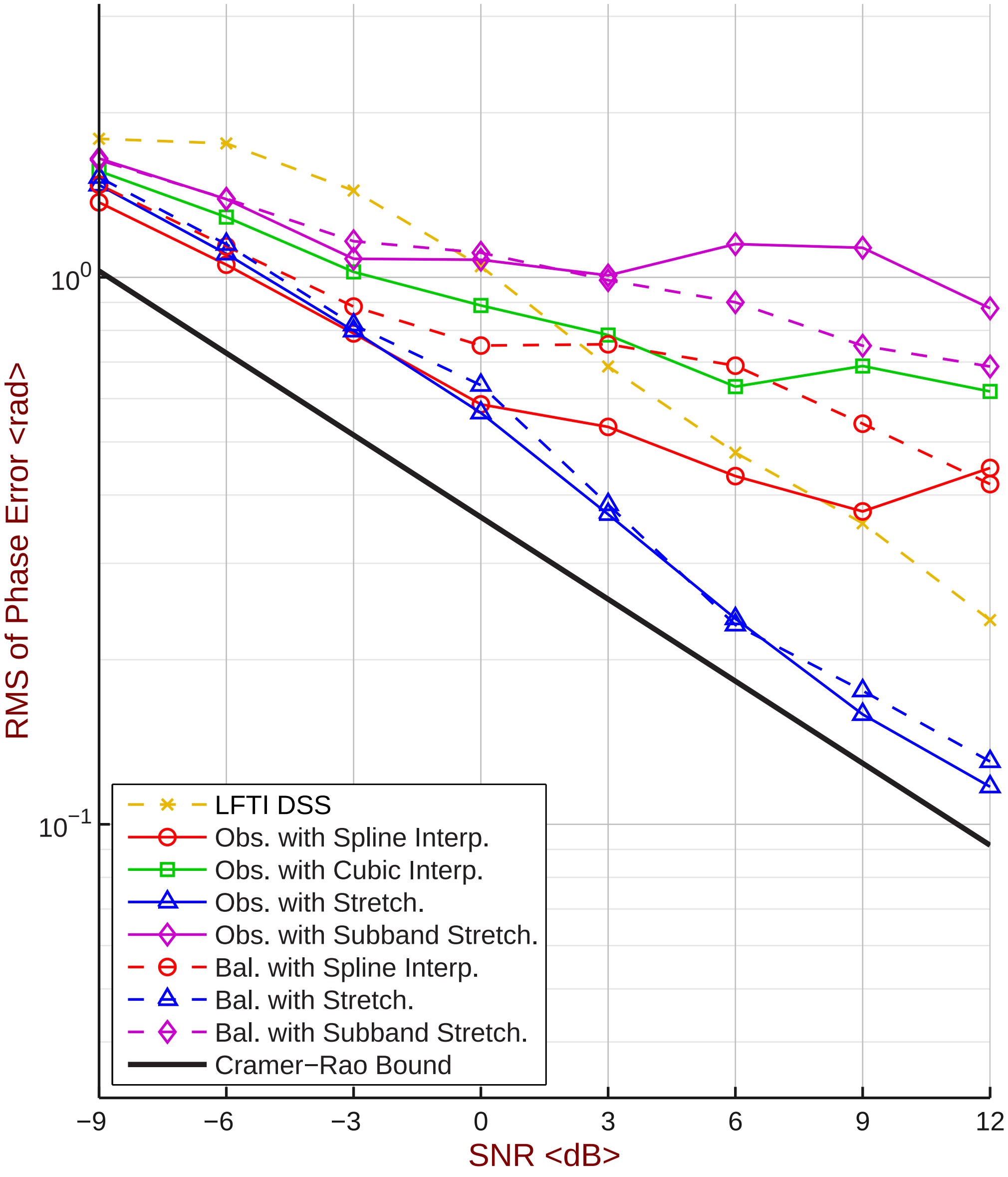


Figure 4.3: Comparison of RMS error of estimating a scatterer's reflection phase offset versus SNR for different algorithms.

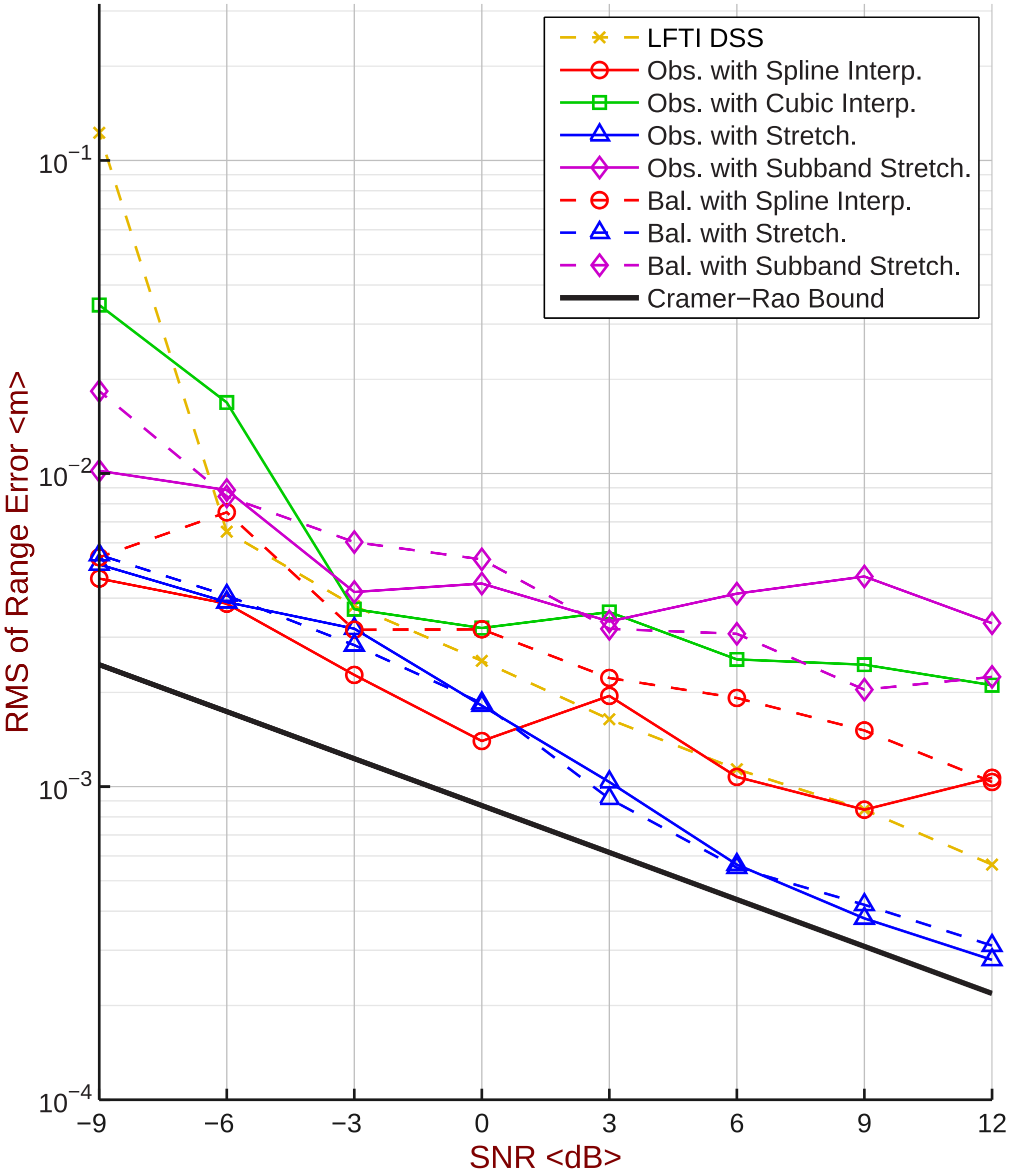


Figure 4.4: Comparison of RMS error of estimating a scatterer's range versus SNR for different algorithms.

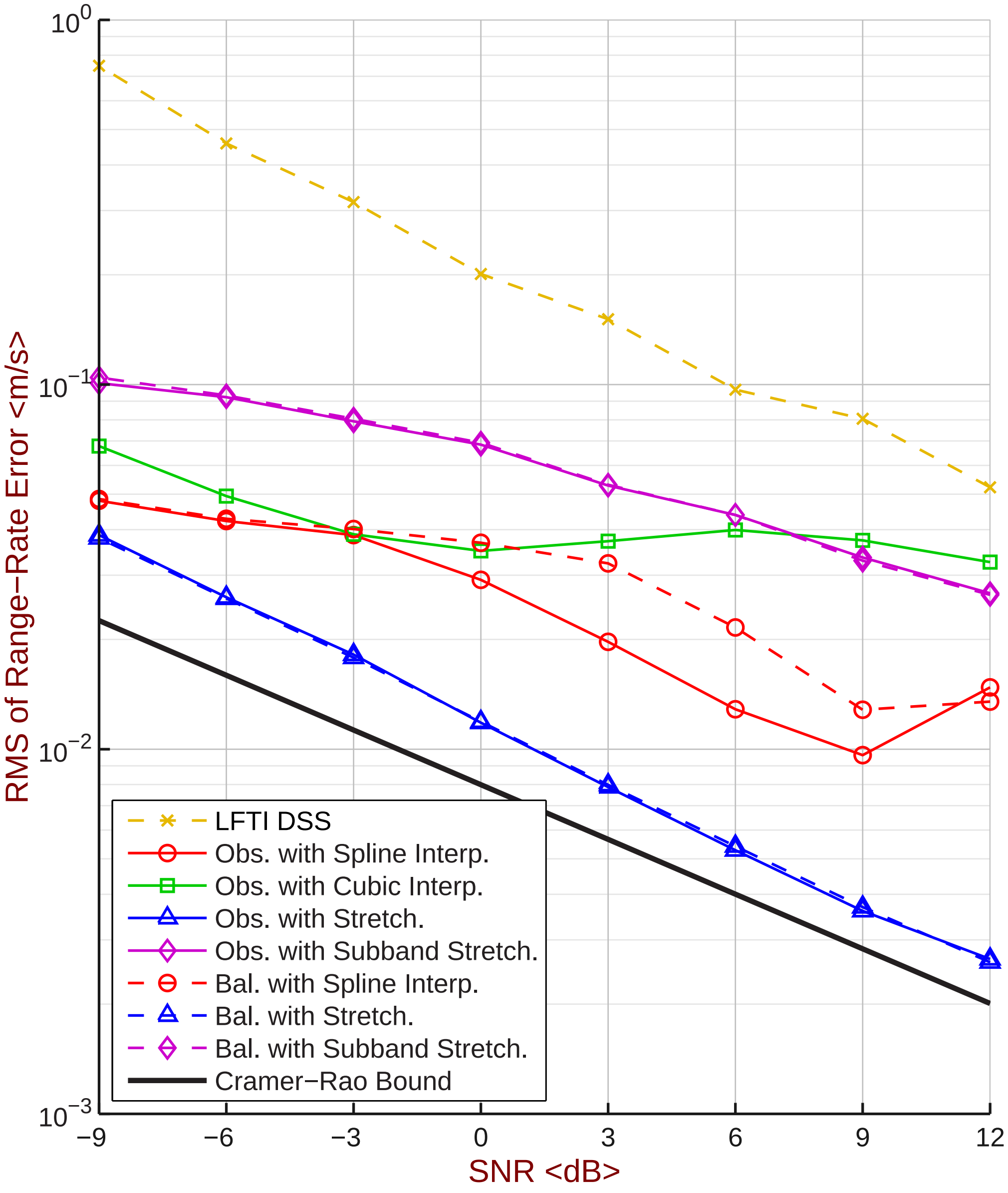


Figure 4.5: Comparison of RMS error of estimating a scatterer's range-rate versus SNR for different algorithms.

best performing methods are the stretching methods with either observability or balanced form state-space solvers.

4.2 Interfering Scatterers - Range Sweep

In this test, we evaluate the estimation performance of each algorithm in identifying a reference scatterer in the presence of a nearby, interfering scatterer having the same range-rate. The SNR is held constant at 6 dB, and both scatterers have unit reflection magnitude, random reflection phase offsets, and range-rate of 0 m/s as a worst-case scenario, having duplicate range-rate eigenvalues thus requiring the final solution to rely solely upon the range eigenvalues. The reference scatterer has a range of 0 m, and the range of the interfering scatterer is swept from 0 m to one quarter of the total range window in 32 steps. At each range step of the interferer, the RMS estimation error is taken over 1000 trials, and the results are presented in Figs. 4.6–4.9.

Similar to the figures of the previous test, the figures here also contain the Cramer-Rao bound, evaluated numerically as discussed in Appendix B. Additionally, the bound for the case of two scatterers tested in this section differs from that of the bounds plotted for one scatterer in Section 4.1 in that the two-scatterer bound depends upon the reflection phase angles of the scatterers, where as the one-scatterer bound does not. This precludes direct comparison of the test result with the analytically obtained, multivariate CRB because we randomize the phase offsets in every trial. The bound we include in Figs. 4.6–4.9 is numerically obtained as the mean of Cramer-Rao variance bound for the phases tested in each of the 1000 trials. This procedure provides the exact Cramer-Rao bound reduced over the reflection phase angles, and only the numerical evaluation is an approximation.

Observing the common trends across Figs. 4.6–4.9, all of the algorithms break down in estimating the reference scatterer's attributes when the interfering scatterer is less than ≈ 0.1 m. For interferer ranges greater than ≈ 0.1 m, we note that both balanced and observability with stretch processing methods perform the best throughout all of the tests, LFTI processing performs the worst, and sub-band stretching fluctuates periodically between the performance curves of stretch and LFTI processing as especially depicted by the phase offset and range errors depicted in Figs. 4.7 and 4.8, respectively.

Also in Fig. 4.8, we observe that the apparent RMS range error appears to drop as the interferer's range approaches that of the reference scatterer. On closer inspection, we found that once the scatterers were too close in range, the algorithms lumped them into a single scatterer whose resulting reflection magnitude and phase further elevated the estimation errors seen in Figs. 4.6 and 4.7, respectively.

4.3 Interfering Scatterers - Range-Rate Sweep

Similar to the test of the previous section, this test also evaluates each algorithm's performance in identifying a reference scatterer with a near-by interfering scatterer. Again, both scatterers have unit reflection magnitude, random reflection phase offsets, and the SNR is held constant at 6 dB, and the reference scatterer has a range of 0 m and a range-rate of 0 m/s. However, in this test, the interfering scatterer will be held at a constant range of 0 m, and have its range-rate swept from 0 m/s to one quarter of the range-rate window in 32 steps. As before, the RMS estimation error is taken over 1000 trials at each step, and we present the results in Figs. 4.10–4.13. The CRB curves depicted in these figures were obtained in the same fashion as those depicted in Section 4.2, and thus are numerical approximations of the lower bounds on estimation variance.

We observe that the algorithms' estimation variance of the reference scatterer's attributes rises significantly in the observability and balanced methods when the interfering scatterer approaches closer than approximately ≈ 0.8 m/s. In Fig. 4.13, we find that unlike previous tests, the stretch processing methods have the highest range-rate estimation variance, surpassed only by LFTI which maintains a constant standard deviation across the varying interferer range-rate, but in Fig. 4.12, stretch processing appears to have a variance similar to that of the other methods in the breakdown region where the interferer's range-rate is less than 0.8 m.

In Fig. 4.13, we note that the algorithms' range-rate estimation of the reference scatterer appears to improve as the interferer's range-rate approaches that of the reference. This is the same failure mechanism observed earlier in the interferer range sweep of Fig. 4.8. When

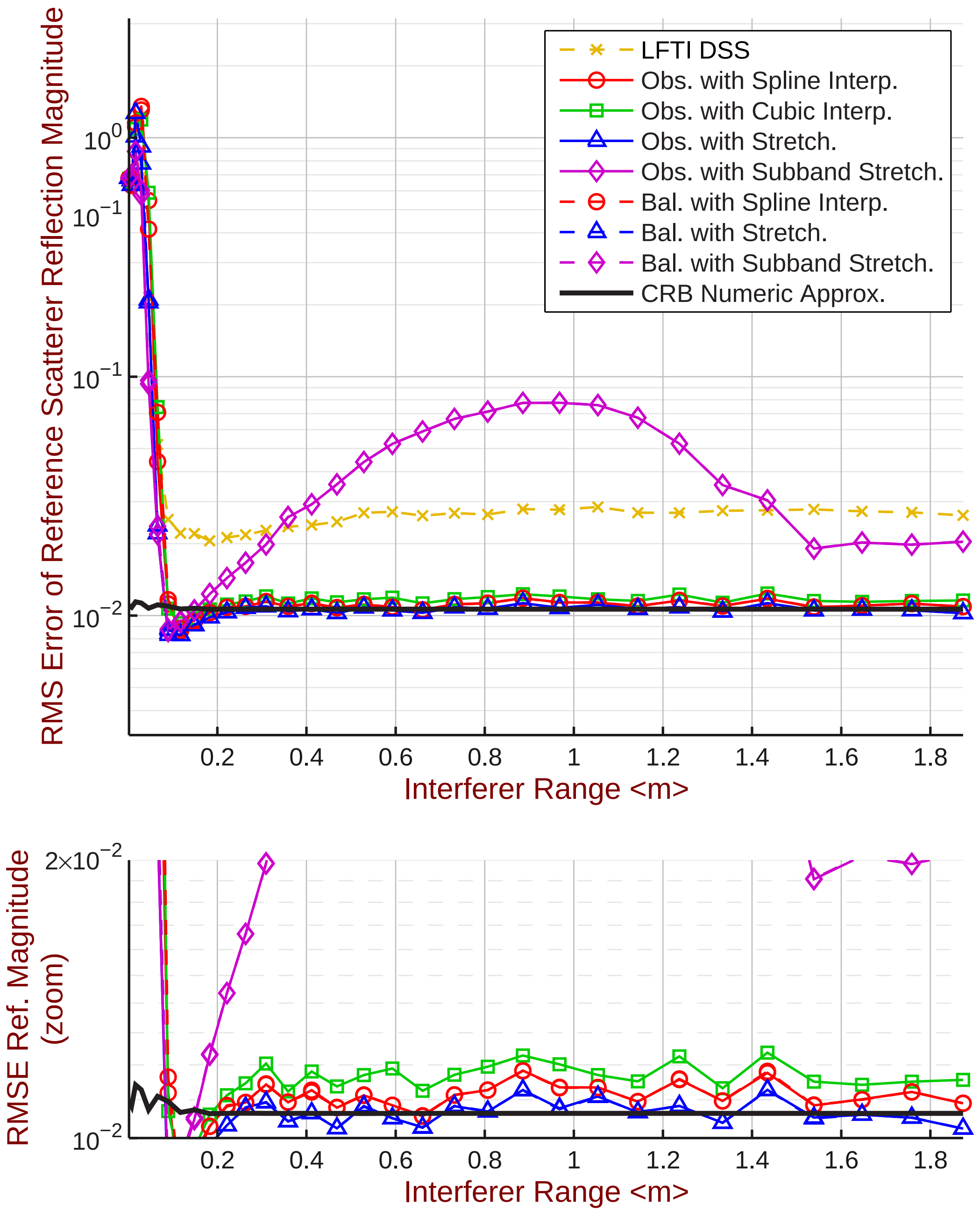


Figure 4.6: Comparison of RMS error of estimating the reference scatterer's reflection magnitude versus the range separation of the interfering scatterer for different algorithms. The lower plot is a magnified view of the upper plot.

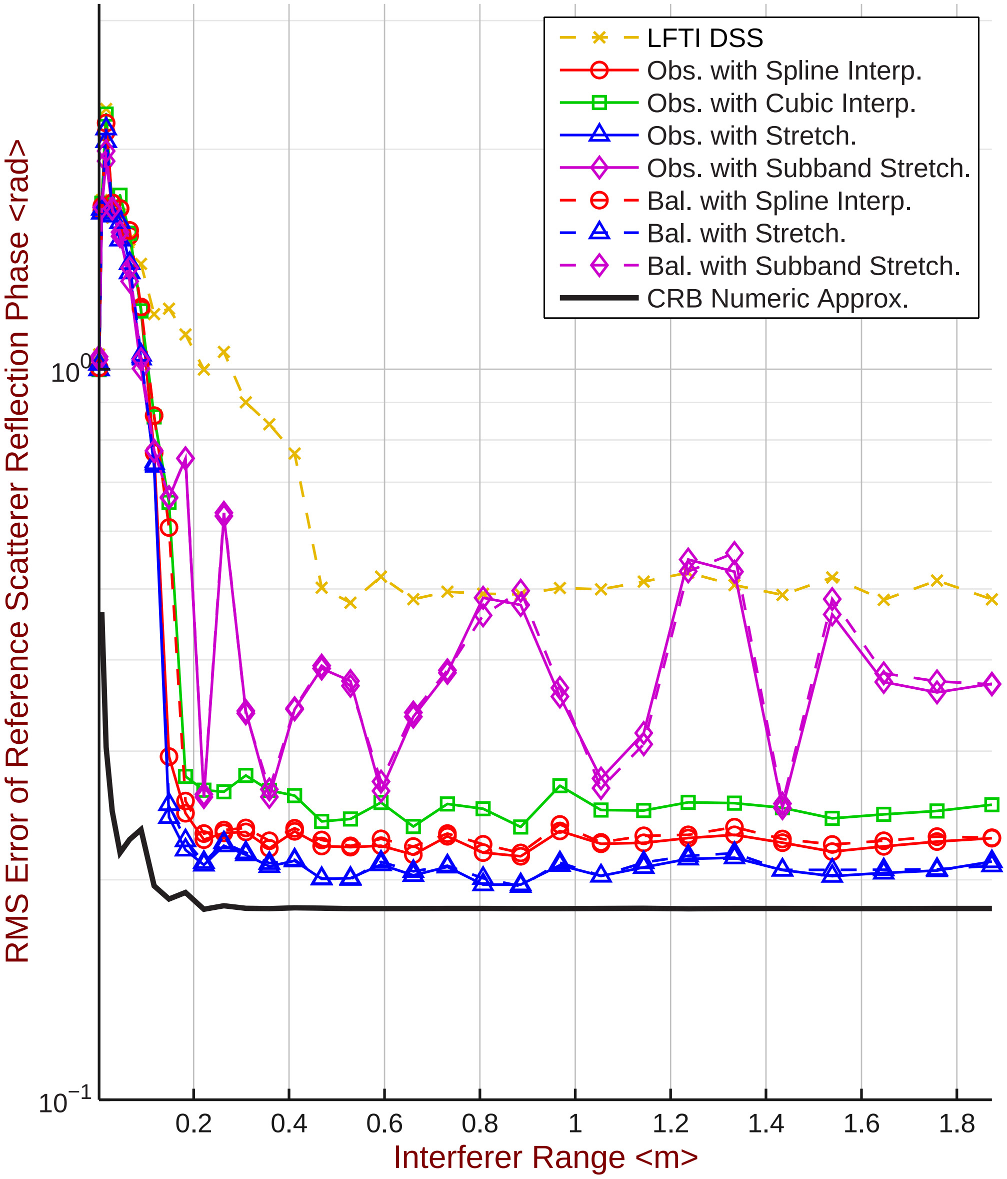


Figure 4.7: Comparison of RMS error of estimating the reference scatterer's reflection phase offset versus the range separation of the interfering scatterer for different algorithms.

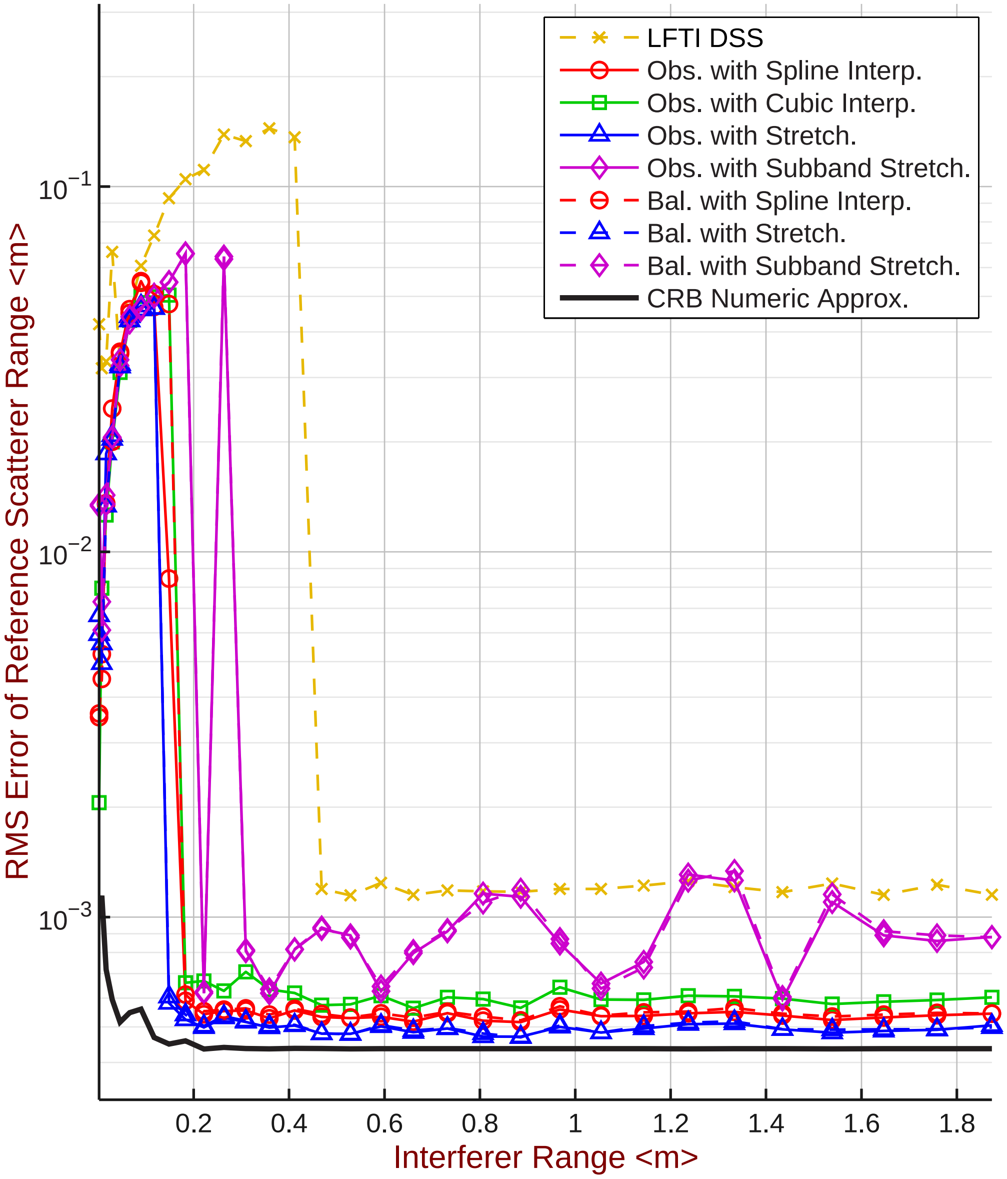


Figure 4.8: Comparison of RMS error of estimating the reference scatterer's range versus the range separation of the interfering scatterer for different algorithms.

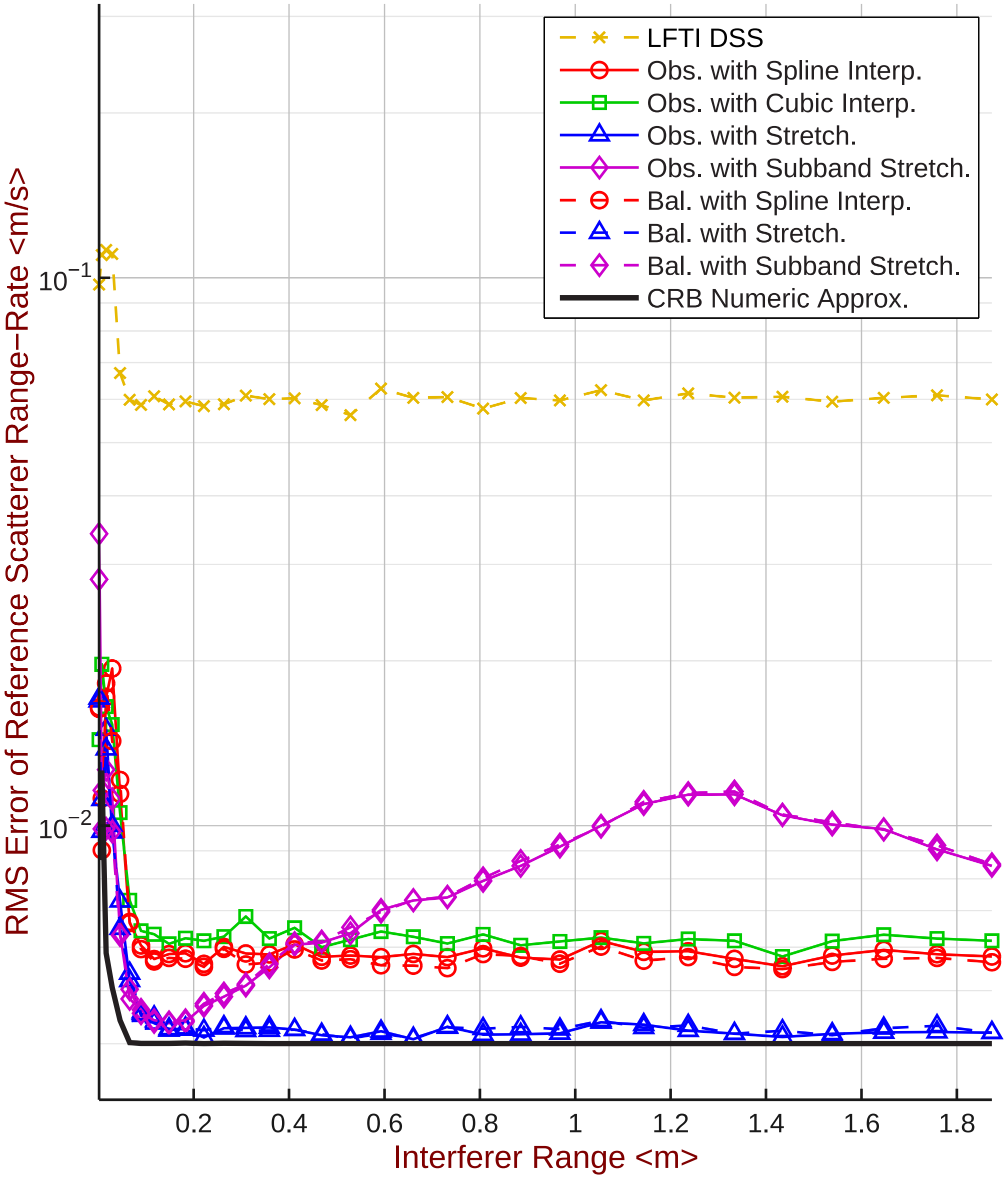


Figure 4.9: Comparison of RMS error of estimating the reference scatterer's range-rate versus the range separation of the interfering scatterer for different algorithms.

the two scatterers' range-rates were close in value, the algorithms grouped them into a single scatterer whose resulting elevated errors in reflection magnitude and phase observable in Figs. 4.10 and 4.11. Observing LFTI's performance in Fig. 4.13, we note that its estimation variance only continues to increase with the interferer's range-rate. This is due to how LFTI's Hankel matrix in Eq. 3.68 iterates over index n, which is the frequency dimension, thus setting up a system analogous to the one-dimensional range-only processing algorithms in Section 2.3.2. However in this test, both scatterers have the same range of 0 m, which forces LFTI's one-dimensional formulation to break down and always group the scatterers regardless of their separation in range-rate.

4.4 Discussion

In Section 4.1, we evaluated the performance of the state-space methods in estimating the reflection coefficient, range, and range-rate of a single unknown scatterer over varying SNR, and we found that the significant algorithmic component most benefiting the estimation variance was the usage of stretch processing instead of the traditional spline or cubic interpolations. In contrast, the subband-stretch processing degraded the estimation error the most, and we attribute this to the use of a narrow bandwidth approximation in that method's formulation. Our Linear Fast-Time Invariant solver had very low estimation error for reflection coefficient and range but the worst error for range-rate. We attribute this to LFTI's range-rate estimates being based predominantly off the state-space estimated time-derivatives formed from single frequencies at a time, and thus, those derivative estimates were created from only 11 data samples per frequency due to our narrow interrogation block size.

Then in Section 4.2, we stress tested the algorithm by supplying two scatterers, a reference having 0 m range and 0 m/s range-rate, and an interferer also having a range-rate of 0 m/s but whose range was varied from 0 to 1.8 m. In these tests, stretch processing maintained the lowest estimation errors for the reference scatterer's reflection coefficient, range, and range-rate, and observed an oscillatory error pattern in the methods relying on subband stretching.

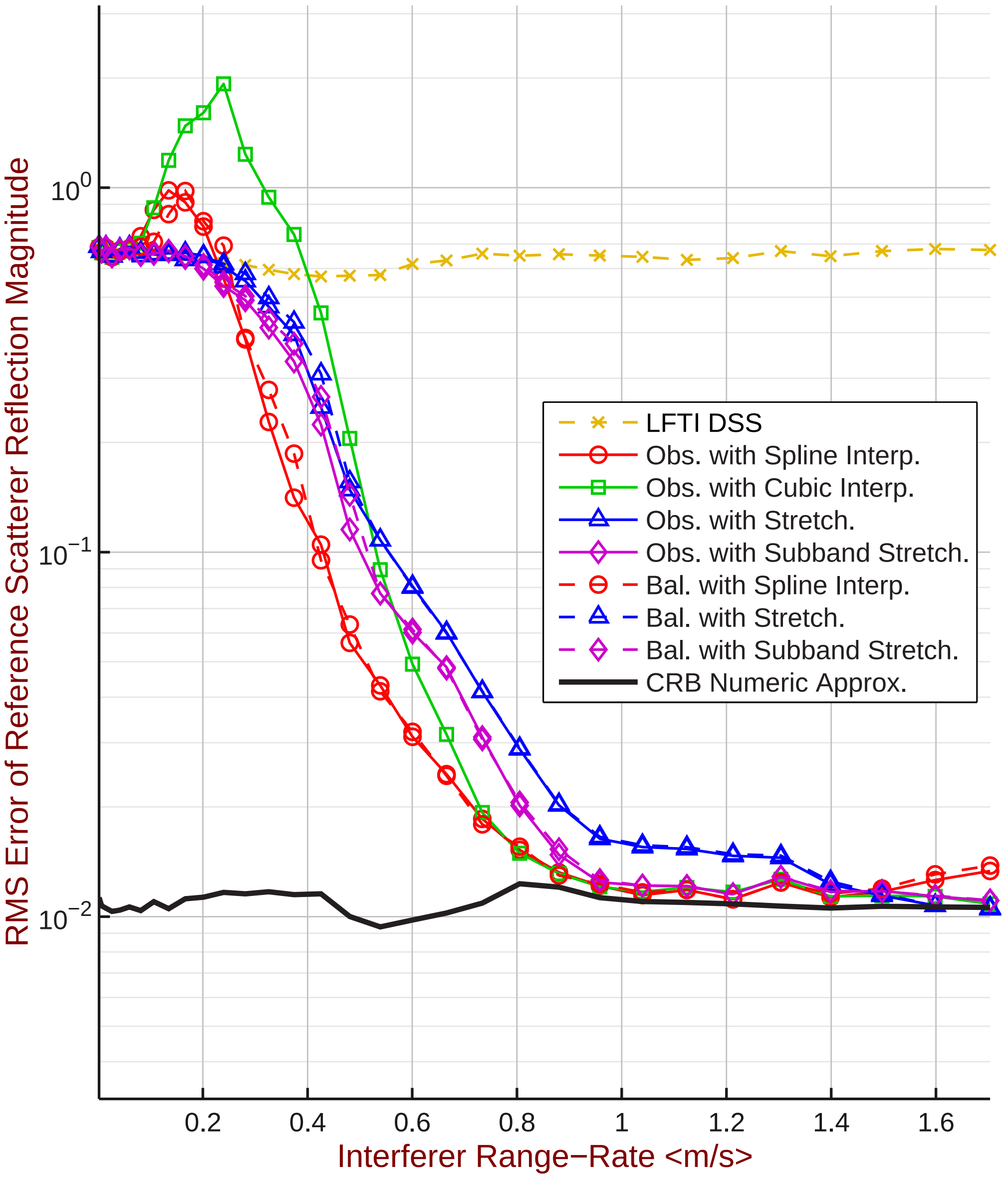


Figure 4.10: Comparison of RMS error of estimating the reference scatterer's reflection magnitude versus the range-rate separation of the interfering scatterer for different algorithms.

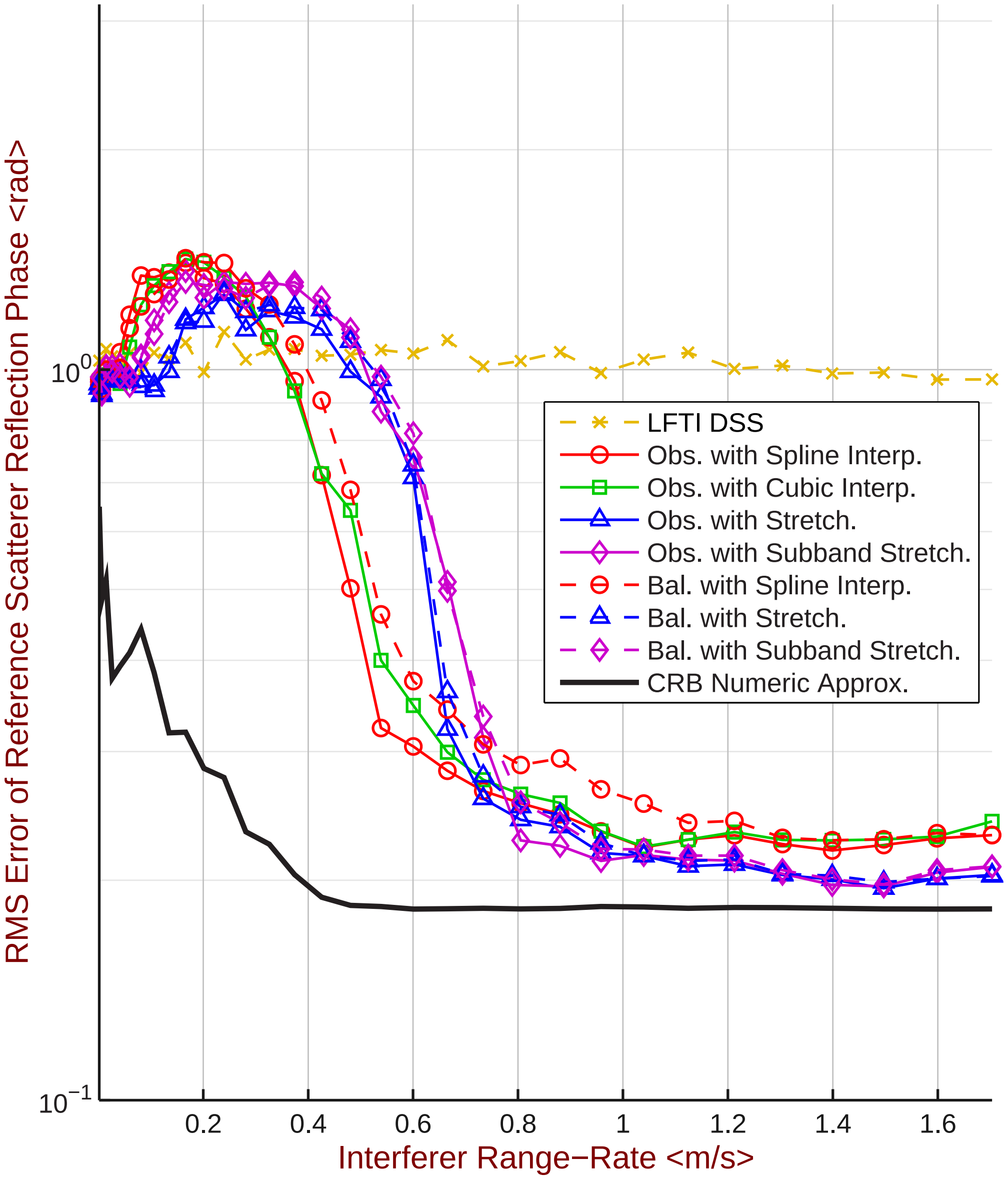


Figure 4.11: Comparison of RMS error of estimating the reference scatterer's reflection phase offset versus the range-rate separation of the interfering scatterer for different algorithms.

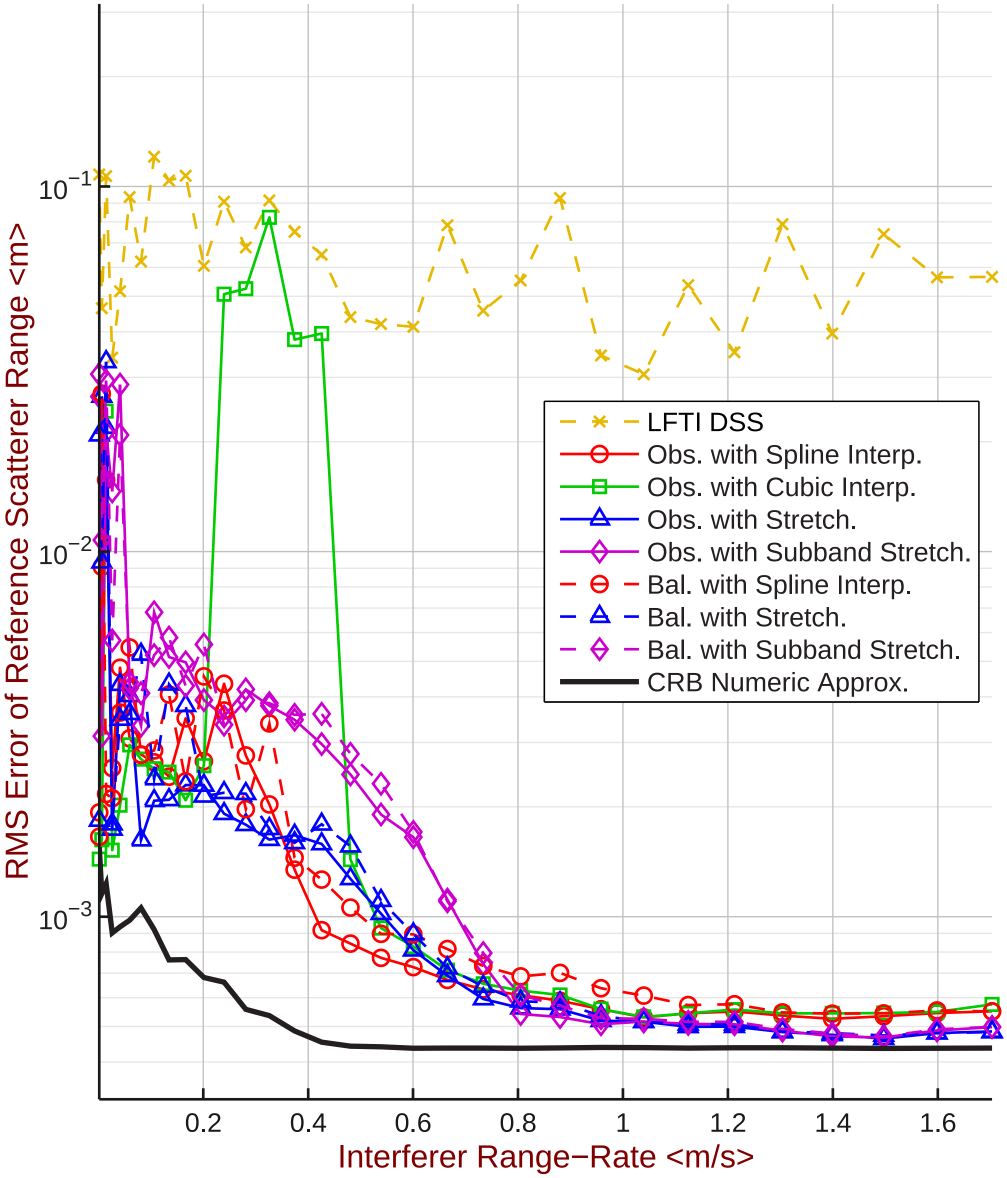


Figure 4.12: Comparison of RMS error of estimating the reference scatterer's range versus the range-rate separation of the interfering scatterer for different algorithms.

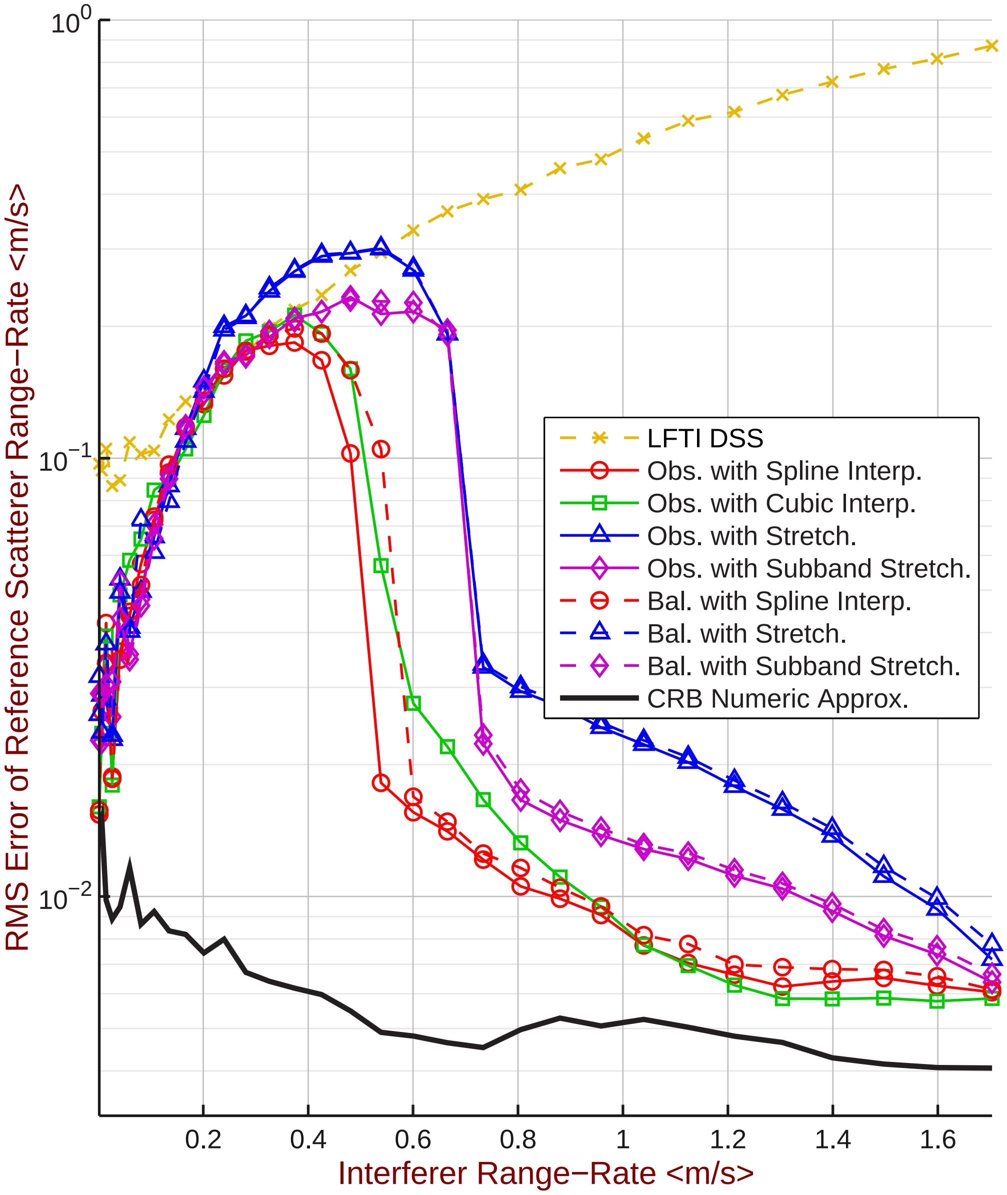


Figure 4.13: Comparison of RMS error of estimating the reference scatterer's range-rate versus the range-rate separation of the interfering scatterer for different algorithms.

Lastly in the scenario presented in Section 4.3, we again stress tested the algorithms with two scatterers, but this time, the scatterers shared the same range of 0 m, while the reference had a constant 0 m/s range-rate and the interferer's range-rate was swept from 0 m/s to 1.7 m/s. This proved to be a fatal blow to LFTI which could not separate the scatterers due to their identical ranges. In the case of stretch processing, it also demonstrated higher errors than the spline and cubic interpolation methods when estimating the reference scatterer's range-rate, but at the same time, it produced very good range estimates. As a note, we found that subband stretching, did improve upon the range-rate estimation compared to stretching, but it still did not perform as well when the interferer's range-rate was less than 0.8 m/s.

In all of the test scenarios, we found that the observability and balanced factorization approaches performed nearly identically in all of the tests. This is a counter-intuitive result as one might expect the observability factorization to yield higher errors due to how it only utilizes $\widetilde{\mathcal{O}}$ and not $\widetilde{\mathcal{C}}$ thus ignoring any information content ostensibly in the V matrix from the singular value decomposition of Eq. 3.18. In contrast, the balanced factorization uses both the $\widetilde{\mathcal{O}}$ and $\widetilde{\mathcal{C}}$ matrices thus utilizing all of the SVD output in Eq. 3.50.

Chapter 5

Conclusions

In the background material presented in Chapter 2, we reviewed state-space approaches for one-dimensional, range-only scatterer identification from radar interrogation data using various approaches—Fourier processing, observability-factorization state-space processing, and balanced-factorization state-space processing. Then in Chapter 3, we proceeded to derive the higher-dimensional observability-factorization state-space family of algorithms which includes ESPRIT [50, 35, 8], Matrix-Enhanced Matrix-Pencil [26], and C-MIMO [45]. After which in Section 3.3.2, we proposed a new multidimensional extension to balanced-factorization processing, which if we were to provide a name, would be most precisely called the Multidimensional Eigensystem Realization Algorithm to highlight its similarity with the one-dimensional Eigensystem Realization Algorithm [29].

After establishing these algorithms, we next turned our attention to the requisite interpolation stage first introduced in Section 3.1, and proposed one and two-dimensional state-space systems as replacements to interpolation, called stretch processing and subband stretch processing in Section 3.3.4. Next, we sought a more ambitious goal in Section 3.4, where we proposed Linear Fast-Time Invariant state-space processing which completely side-steps the simultaneous diagonalization normally required in the multidimensional state-space methods, by employing a time-derivative to create a one-dimensional state-space capable of solving the multidimensional problem of identifying scatterer range and range-rate.

Lastly, we measured the performance of our proposed techniques using the test scenarios presented in Chapter 4: one general test evaluating single scatterer identification, and two specialized test scenarios designed to induce duplicate eigenvalues in the state-space methods. Comparing the multidimensional balanced versus the observability factorizations, we found negligible difference in estimation performance, but when comparing the interpolation methods versus stretch processing, we found that stretch processing often produced the lowest errors of all algorithms, and that subband stretching suffered due to its reliance upon a narrow-bandwidth assumption even though it was only 1% bandwidth.

In future work, we would continue to explore the LFTI processing approach and hybrids with stretch processing, because even though LFTI appeared to break down more readily in the special case tests involving the interfering scatterer in Sections 4.2 and 4.3, it still maintained some of the lowest estimation errors for reflection magnitude and range in the single scatterer tests. Given that LFTI is formulated as a one-dimensional block Hankel, it would seem that the other algorithms should have faired better since their two-dimensional block-nested Hankel matrices should have been able to better incorporate the additional raw data for further error reduction. Augmenting the two-dimensional Hankel matrices utilized in the balanced+stretch or observability+stretch algorithms may yield further improvements.

Also, we would pursue further exploration into the processing stage common to all of the studied algorithms: factoring the Hankel matrix of the input data into the observability and controllability matrices discussed in Section 2.3.2. In this factoring stage, numerous approaches have been tried ranging from visual inspection of the singular values—which is inappropriate for applications requiring automated processing such as radar systems—to attempting to apply Akaike Information Criterion or Minimum Description Length techniques—which have been found to frequently over-estimate the number of scatterers and thus permitting too much of the noise subspace into the solution set. In this research, we employed an empirically derived heuristic wherein we automatically determine the number of scatterers based of the number of singular values (σ_s) falling within a certain distance of the largest (σ_1) as described in Eq. 2.40. Our threshold of 20 dB appeared to work well in our example plots of Chapters 2 and 3 and for the validation tests of Chapter 4. However, the threshold seems dependent upon on the number of data samples over both frequency and time selected for each processing block; the exact relationship between the threshold and data size proved elusive in a cursory exploration, because it was conducted using a singular value decomposition which does not preserve the Hankel structure when used for rank truncations. This analysis should be carried out in further detail but using Hankel-preserving methods such as the Fast-Hankel SVD [38, 53] or other structured matrix rank reduction techniques [16, 40, 36, 6].

Appendix A

State-Space Similarity Transformations

A system's internal state variables may be representative for many different forms. But suppose we were given a state-space realization,

$$\begin{aligned} \vec{x}_{n+1} &= \mathrm{A}\vec{x}_n + \mathrm{B}\vec{u}_n \\ \vec{y}_n &= \mathrm{C}\vec{x}_n + \mathrm{D}\vec{u}_n \end{aligned} \qquad n = 0, 1, 2, \ldots \tag{A.1}$$

we could change the state representation by some nonsingular transformation

$$\widetilde{\vec{x}}_n = \mathrm{T}\vec{x}_n \tag{A.2}$$

and produce a different realization:

$$\begin{aligned} \widetilde{\vec{x}}_{n+1} &= \widetilde{\mathrm{A}}\widetilde{\vec{x}}_n + \widetilde{\mathrm{B}}\vec{u}_n \\ \vec{y}_n &= \widetilde{\mathrm{C}}\widetilde{\vec{x}}_n + \widetilde{\mathrm{D}}\vec{u}_n \end{aligned} \qquad n = 0, 1, 2, \ldots \tag{A.3}$$

where

$$\widetilde{\mathrm{A}} \triangleq \mathrm{TAT}^{-1} \qquad \widetilde{\mathrm{B}} \triangleq \mathrm{TB} \qquad \widetilde{\mathrm{C}} \triangleq \mathrm{CT}^{-1} \qquad \widetilde{\mathrm{D}} \triangleq \mathrm{D} \tag{A.4}$$

Thus if we were to substitute Eq. A.2 and Eq. A.4 into Eq. A.3,

$$\begin{aligned} \mathrm{T}\vec{x}_{n+1} &= \mathrm{TAT}^{-1}\mathrm{T}\vec{x}_n + \mathrm{TB}\vec{u}_n \\ \vec{y}_n &= \mathrm{CT}^{-1}\mathrm{T}\vec{x}_n + \mathrm{D}\vec{u}_n \end{aligned} \qquad n = 0, 1, 2, \ldots \tag{A.5}$$

we would obtain our original system, Eq. A.1. This new representation (Eq. A.3) is not a new system. It still produces identical output $\vec{y}_n$ for a given input $\vec{u}_n$. It is just that the the state variables $\vec{x}_n$ have been reordered or meanings reassigned.

The transformation on A into $\widetilde{\mathrm{A}}$ (Eq. A.4) is called a *similarity transformation* [21, p. 311], and thus the state-variable transformation into Eq. A.3 is also called a similarity transformation in control engineering [30, p. 54 and p. 664]. These transformations preserve the eigenvalues of A thus enabling us to use eigen or Schur decompositions to get back to a useful canonical form.

We can perform similar operations to identify the new observability and controllability matrices, $\widetilde{\mathcal{O}}$ and $\widetilde{\mathcal{C}}$:

$$\widetilde{\mathcal{O}} \triangleq \begin{bmatrix} \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^0 \\ \widetilde{\mathrm{C}}\widetilde{\mathrm{A}}^1 \\ \vdots \end{bmatrix} \qquad \widetilde{\mathcal{C}} \triangleq \begin{bmatrix} \widetilde{\mathrm{A}}^0\widetilde{\mathrm{B}} & \widetilde{\mathrm{A}}^1\widetilde{\mathrm{B}} & \dots \end{bmatrix} \tag{A.6}$$

$$\widetilde{\mathcal{O}} \triangleq \begin{bmatrix} \mathrm{CT}^{-1}\left(\mathrm{TAT}^{-1}\right)^0 \\ \mathrm{CT}^{-1}\left(\mathrm{TAT}^{-1}\right)^1 \\ \vdots \end{bmatrix} \qquad \widetilde{\mathcal{C}} \triangleq \begin{bmatrix} \left(\mathrm{TAT}^{-1}\right)^0\mathrm{TB} & \left(\mathrm{TAT}^{-1}\right)^1\mathrm{TB} & \dots \end{bmatrix} \tag{A.7}$$

$$\widetilde{\mathcal{O}} \triangleq \begin{bmatrix} \mathrm{CT}^{-1}\mathrm{TA}^0\mathrm{T}^{-1} \\ \mathrm{CT}^{-1}\mathrm{TA}^1\mathrm{T}^{-1} \\ \vdots \end{bmatrix} \qquad \widetilde{\mathcal{C}} \triangleq \begin{bmatrix} \mathrm{TA}^0\mathrm{T}^{-1}\mathrm{TB} & \mathrm{TA}^1\mathrm{T}^{-1}\mathrm{TB} & \dots \end{bmatrix} \tag{A.8}$$

$$\widetilde{\mathcal{O}} \triangleq \begin{bmatrix} \mathrm{CA}^0 \\ \mathrm{CA}^1 \\ \vdots \end{bmatrix} \mathrm{T}^{-1} \qquad \widetilde{\mathcal{C}} \triangleq \mathrm{T}\begin{bmatrix} \mathrm{A}^0\mathrm{B} & \mathrm{A}^1\mathrm{B} & \dots \end{bmatrix} \tag{A.9}$$

$$\widetilde{\mathcal{O}} \triangleq \mathcal{O}\mathrm{T}^{-1} \qquad \widetilde{\mathcal{C}} \triangleq \mathrm{T}\mathcal{C} \tag{A.10}$$

Note that the Hankel matrices H, which factor into any $\mathcal{O}\mathcal{C}$ in our system identification discussion (Section 2.3.2), also factor into any $\widetilde{\mathcal{O}}\widetilde{\mathcal{C}}$ as well.

For further information, please see [12, p. 95,159].

Appendix B

Cramer-Rao Bound for Range and Range-Rate Estimation

This appendix presents the Cramer-Rao bound (CRB) on parameter estimation performance, which provides a bound on the covariance matrix of any *unbiased* estimate of some unknown parameters [51]. In the context of the present research, our unknown parameters are those attributes possessed by the scatterers within a target scene including complex reflection coefficient, range, and higher time-derivatives of range such as range-rate and range-acceleration. For simplicity, we only discuss the CRB derivation for the case of a single radar scatterer, and we do not include higher time-derivatives of range beyond that of range-acceleration.

Following Van Trees [51], we start the derivation by defining our signal and adding white Gaussian noise as shown in Eqs. B.1 and B.2, respectively. In one radar target interrogation, we receive a vector of reflectivity measurements, $\tilde{z}_{n,k}$, taken at each of $n\Delta f$ frequencies and $k\Delta t$ times, and consisting of the original reflected signal, $y_{n,k}$, corrupted with additive white Gaussian noise, $\tilde{x}_{n,k}$. The parameters A, ϕ, r, $\dot{r}$, and $\ddot{r}$ refer to the unknown attributes of a scatterer: reflection magnitude, reflection-phase offset, range, range-rate, and range-acceleration, respectively, and they are assumed to be real with $A > 0$. For notational convenience, we define $\theta \triangleq \begin{bmatrix} A & \phi & r & \dot{r} & \ddot{r} \end{bmatrix}^T$ to encapsulate these unknowns.

$$y_{n,k} = Ae^{-4j\pi\frac{n\Delta f}{c}\left(r+\dot{r}k\Delta t+\frac{1}{2}\ddot{r}k^2\Delta t^2\right)+\phi j} \tag{B.1}$$

$$\tilde{z}_{n,k} = y_{n,k} + \tilde{x}_{n,k} \tag{B.2}$$

We start this derivation by first treating $\tilde{z}_{n,k}$ as a proper random vector over n and k [41] and presenting its probability density function (PDF) $p_{\tilde{z}|\theta}$ in Eq. B.3 where σ^2 is the noise variance.

$$p_{\tilde{z}|\theta}(\vec{z}) = \prod_{n=n_0}^{n_0+N-1}\prod_{k=k_0}^{k_0+K-1}\frac{1}{\pi\sigma^2}e^{-\left(\tilde{z}_{n,k}-y_{n,k}\right)^H\left(\tilde{z}_{n,k}-y_{n,k}\right)/\sigma^2} \tag{B.3}$$

Next, we identify the *log-likelihood function* $L_{\vec{z}}(\theta)$ from the natural logarithm of $p_{\tilde{z}|\theta}$ in Eq. B.5 and use it to construct the $\mathrm{J}_{v,w}$ elements of the Fisher information matrix (FIM) in Eq. B.6 where $E\left[\ldots\right]$ denotes taking the expected value.

$$\begin{aligned}
L_{\vec{z}}(\theta) &\triangleq \ln p_{\tilde{z}|\theta}(\vec{z}) && \text{(B.4)}\\
&= -\ln\left(\pi\sigma^2\right) - \frac{1}{\sigma^2}\sum_{n=n_0}^{n_0+N-1}\sum_{k=k_0}^{k_0+K-1}\left(\tilde{z}_{n,k}-y_{n,k}\right)^H\left(\tilde{z}_{n,k}-y_{n,k}\right)\\
&= -\ln\left(\pi\sigma^2\right) - \frac{1}{\sigma^2}\sum_{n=n_0}^{n_0+N-1}\sum_{k=k_0}^{k_0+K-1}\Re^2\left(\tilde{z}_{n,k}-y_{n,k}\right)+\Im^2\left(\tilde{z}_{n,k}-y_{n,k}\right)\\
&= -\ln\left(\pi\sigma^2\right) - \frac{1}{\sigma^2}\sum_{k=k_0}^{k_0+K-1}\sum_{n=n_0}^{n_0+N-1}\Bigg[A^2+\Re\left(\tilde{z}_{n,k}\right)^2+\Im\left(\tilde{z}_{n,k}\right)^2 \quad \ldots\\
&\quad - 2A\Re\left(\tilde{z}_{n,k}\right)\cos\left(\frac{4\pi n\Delta f}{c}(r+\dot{r}k\Delta t+\frac{1}{2}\ddot{r}(k\Delta t)^2)-\phi\right) \quad \ldots\\
&\quad + 2A\Im\left(\tilde{z}_{n,k}\right)\sin\left(\frac{4\pi n\Delta f}{c}(r+\dot{r}k\Delta t+\frac{1}{2}\ddot{r}(k\Delta t)^2)-\phi\right)\Bigg] && \text{(B.5)}\\
\mathrm{J}_{v,w} &\triangleq -E\left[\frac{\partial L_{\vec{z}}(\theta)}{\partial\theta_v}\cdot\frac{\partial L_{\vec{z}}(\theta)}{\partial\theta_w}\right]\\
&= -E\left[\frac{\partial^2 L_{\vec{z}}(\theta)}{\partial\theta_v\partial\theta_w}\right] && \text{(B.6)}
\end{aligned}$$

$$\mathrm{J} = \begin{bmatrix} \mathrm{J}_{1,1} & 0 & 0 & 0 & 0 \\ 0 & \mathrm{J}_{2,2} & \mathrm{J}_{2,3} & \mathrm{J}_{2,4} & \mathrm{J}_{2,5} \\ 0 & \mathrm{J}_{3,2} & \mathrm{J}_{3,3} & \mathrm{J}_{3,4} & \mathrm{J}_{3,5} \\ 0 & \mathrm{J}_{4,2} & \mathrm{J}_{4,3} & \mathrm{J}_{4,4} & \mathrm{J}_{4,5} \\ 0 & \mathrm{J}_{5,2} & \mathrm{J}_{5,3} & \mathrm{J}_{5,4} & \mathrm{J}_{5,5} \end{bmatrix} \tag{B.7}$$

$$\mathrm{J}_{1,1} = \frac{2NK}{\sigma^2}$$

$$\mathrm{J}_{2,2} = \frac{2A^2NK}{\sigma^2}$$

$$\mathrm{J}_{2,3} = \frac{-4A^2\pi\Delta f N(2n_0+N-1)K}{c\sigma^2}$$

$$\mathrm{J}_{2,4} = \frac{-2A^2\pi\Delta f\Delta t N(2n_0+N-1)K(K-1+2k_0)}{c\sigma^2}$$

$$\mathrm{J}_{2,5} = \frac{-A^2\pi\Delta f\Delta t^2 N(2n_0+N-1)K\left(6{k_0}^2+6k_0K+2K^2-6k_0-3K+1\right)}{3c\sigma^2}$$

$$\mathrm{J}_{3,2} = \mathrm{J}_{2,3}$$

$$\mathrm{J}_{3,3} = \frac{16}{3}\,\frac{A^2\pi^2\Delta f^2 N\left(6{n_0}^2+6n_0N+2N^2-6n_0-3N+1\right)K}{c^2\sigma^2}$$

$$\mathrm{J}_{3,4} = \frac{8A^2\pi^2\Delta f^2\Delta t N\left(6{n_0}^2+6n_0N+2N^2-6n_0-3N+1\right)K(K-1+2k_0)}{3c^2\sigma^2}$$

$$\mathrm{J}_{3,5} = \frac{4A^2\pi^2\Delta f^2\Delta t^2 N\left(6{n_0}^2+6n_0N+2N^2-6n_0-3N+1\right)K\left(6{k_0}^2+6k_0K+2K^2-6k_0-3K+1\right)}{9c^2\sigma^2}$$

$$\mathrm{J}_{4,2} = \mathrm{J}_{2,4}$$

$$\mathrm{J}_{4,3} = \mathrm{J}_{3,4}$$

$$\mathrm{J}_{4,4} = \frac{8A^2\pi^2\Delta f^2\Delta t^2 N\left(6{n_0}^2+6n_0N+2N^2-6n_0-3N+1\right)K\left(6{k_0}^2+6k_0K+2K^2-6k_0-3K+1\right)}{9c^2\sigma^2}$$

$$\mathrm{J}_{4,5} = \frac{2A^2\pi^2\Delta f^2\Delta t^3 N\left(6{n_0}^2+6n_0N+2N^2-6n_0-3N+1\right)K(K-1+2k_0)\left(K^2-K+2k_0K+2{k_0}^2-2k_0\right)}{3c^2\sigma^2}$$

$$\mathrm{J}_{5,2} = \mathrm{J}_{2,5}$$

$$\mathrm{J}_{5,3} = \mathrm{J}_{3,5}$$

$$\mathrm{J}_{5,4} = \mathrm{J}_{4,5}$$

$$\mathrm{J}_{5,5} = \frac{2A^2\pi^2\Delta f^2\Delta t^4 N\left(6{n_0}^2+6n_0N+2N^2-6n_0-3N+1\right)K}{45c^2\sigma^2} \quad \cdots$$
$$\cdot\frac{\left(30{k_0}^4+60{k_0}^3K+60{k_0}^2K^2+30k_0K^3+6K^4-60{k_0}^3-90{k_0}^2K-60k_0K^2-15K^3+30{k_0}^2+30k_0K+10K^2-1\right)}{1}$$

Lastly, we obtain the Cramer-Rao bound $C_{CR}(\theta)$ from the inverse of J in Eq. B.8. The five diagonal elements of $C_{CR}(\theta)$ are the lower bounds on the estimation variance of the five unknown parameters listed in θ and are plotted in Fig. B.1. The off-diagonal elements of $C_{CR}(\theta)$ are the lower bounds on the estimation covariance between unknown parameters.

$$
\begin{aligned}
C_{CR}(\theta) &\triangleq \mathrm{J}^{-1} \\
&= \begin{bmatrix}
\mathrm{C}_{1,1} & 0 & 0 & 0 & 0 \\
0 & \mathrm{C}_{2,2} & \mathrm{C}_{2,3} & 0 & 0 \\
0 & \mathrm{C}_{3,2} & \mathrm{C}_{3,3} & \mathrm{C}_{3,4} & \mathrm{C}_{3,5} \\
0 & 0 & \mathrm{C}_{4,3} & \mathrm{C}_{4,4} & \mathrm{C}_{4,5} \\
0 & 0 & \mathrm{C}_{5,3} & \mathrm{C}_{5,4} & \mathrm{C}_{5,5}
\end{bmatrix} \\
\mathrm{C}_{1,1} &= \frac{\sigma^2}{2NK} \\
\mathrm{C}_{2,2} &= \frac{\left(6{n_0}^2 + 6n_0N + 2N^2 - 6n_0 - 3N + 1\right)\sigma^2}{A^2NK\left(N^2-1\right)} \\
\mathrm{C}_{2,3} &= \frac{3\left(2n_0 + N - 1\right)c\sigma^2}{4\pi\Delta f A^2NK\left(N^2-1\right)} \\
\mathrm{C}_{3,2} &= \mathrm{C}_{2,3} \\
\mathrm{C}_{3,3} &= \frac{9Qc^2\sigma^2}{8KN\Delta f^2\pi^2A^2\left(6{n_0}^2 + 6n_0N + 2N^2 - 6n_0 - 3N + 1\right)\left(N^2-1\right)\left(K^2-1\right)\left(K^2-4\right)} \\
\mathrm{C}_{3,4} &= \frac{-27\left(K - 1 + 2k_0\right)\left(10{k_0}^2 - 10k_0 + 10k_0K + 2 - 5K + 2K^2\right)c^2\sigma^2}{8KN\Delta f^2\pi^2A^2\Delta t\left(6{n_0}^2 + 6n_0N + 2N^2 - 6n_0 - 3N + 1\right)\left(K^2-1\right)\left(K^2-4\right)} \\
\mathrm{C}_{3,5} &= \frac{45\sigma^2c^2\left(6{k_0}^2 + 6k_0K + K^2 - 6k_0 - 3K + 2\right)}{4\Delta f^2NK\pi^2A^2\Delta t^2\left(6{n_0}^2 + 6n_0N + 2N^2 - 6n_0 - 3N + 1\right)\left(K^2-1\right)\left(K^2-4\right)} \\
\mathrm{C}_{4,3} &= \mathrm{C}_{3,4} \\
\mathrm{C}_{4,4} &= \frac{9\sigma^2c^2\left(16K^2 + 60k_0K - 30K - 60k_0 + 11 + 60{k_0}^2\right)}{4\Delta f^2NK\pi^2A^2\Delta t^2\left(6{n_0}^2 + 6n_0N + 2N^2 - 6n_0 - 3N + 1\right)\left(K^2-1\right)\left(K^2-4\right)} \\
\mathrm{C}_{4,5} &= \frac{-135\left(K - 1 + 2k_0\right)c^2\sigma^2}{2\Delta t^3A^2\pi^2\Delta f^2NK\left(6{n_0}^2 + 6n_0N + 2N^2 - 6n_0 - 3N + 1\right)\left(K^2-1\right)\left(K^2-4\right)} \\
\mathrm{C}_{5,3} &= \mathrm{C}_{3,5} \\
\mathrm{C}_{5,4} &= \mathrm{C}_{4,5}
\end{aligned}
\tag{B.8}
$$

$$\mathrm{C}_{5,5} = \frac{135\sigma^2 c^2}{A^2\pi^2\Delta f^2\Delta t^4 NK\left(6{n_0}^2+6n_0N+2N^2-6n_0-3N+1\right)\left(K^2-1\right)\left(K^2-4\right)}$$

$$\begin{aligned} Q =& -10n_0NK^2+42N^2k_0K+42N^2{k_0}^2K^2+2n_0NK^4+12N^2k_0K^3-42N^2k_0K^2+60N^2{k_0}^3K-90N^2{k_0}^2K \quad \ldots \\ & -4N+6K-8n_0+12k_0+8n_0N+4N^2-30{k_0}^4+60{k_0}^3-42{k_0}^2+2{n_0}^2K^4+30N^2{k_0}^4+2N^2K^4 \quad \ldots \\ & -60N^2{k_0}^3-6N^2K^3-2n_0K^4-10{n_0}^2K^2+42N^2{k_0}^2+6N^2K^2-12N^2k_0-6N^2K+10n_0K^2 \quad \ldots \\ & +5NK^2-NK^4+8{n_0}^2+6K^3-K^4-42k_0K-11K^2+90{k_0}^2K+42k_0K^2-60{k_0}^3K-42{k_0}^2K^2-12k_0K^3 \end{aligned}$$

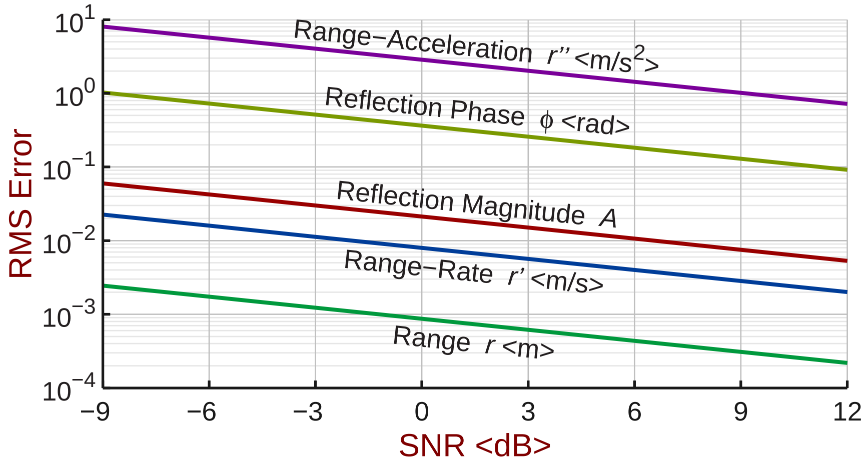


Figure B.1: Cramer-Rao lower bounds on estimation variance for identifying a scatterer's attributes: reflection magnitude, reflection phase offset, range, range-rate, and range-acceleration. These bounds are plotted for radar data having the same center frequency, bandwidth, interrogation rate, and window as used in the validation tests of Chapter 4. Though the scatterer attributes have differing units, we present them without units on the same graph for visual comparison.

The derivation for two scatterers follows naturally from this derivation by extending Eq. B.1 to

$$y_{n,k} = \sum_{s=1}^{2} A_s e^{-4j\pi\frac{n\Delta f}{c}\left(r_s+\dot{r}_sk\Delta t+\frac{1}{2}\ddot{r}_sk^2\Delta t^2\right)+\phi_s j} \tag{B.9}$$

and defining

$$\theta \triangleq \begin{bmatrix} A_1 & \phi_1 & r_1 & \dot{r}_1 & \ddot{r}_1 & A_2 & \phi_2 & r_2 & \dot{r}_2 & \ddot{r}_2 \end{bmatrix}^T . \tag{B.10}$$

However, a closed-form, analytic solution for $C_{CR}(\theta)$ proved intractable given today's available computing resources; for the two-scatterer validation tests of Chapter 4, we numerically evaluated J^{-1} in Eq. B.8 to obtain the Cramer-Rao bound.

Bibliography

[1] Abed-Meraim, K., and Hua, Y. A least-squares approach to joint schur decomposition. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing* (May 1998), vol. 4, pp. 2541–2544.

[2] Akaike, H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control 19*, 6 (Dec. 1974), 716–723.

[3] Arun, K. S. A unitarily constrained total least squares problem in signal processing. *SIAM Journal on Matrix Analysis and Applications 13*, 3 (1992), 729–745.

[4] Bayard, D. S. An algorithm for state-space frequency domain identification without windowing distortions. *IEEE Transactions on Automatic Control 39*, 9 (Sept. 1994), 1880–1885.

[5] Bazán, F. Eigensystem realization algorithm (ERA): reformulation and system pole perturbation analysis. *Journal of Sound and Vibration 274* (2004), 433–444.

[6] Botting, B. Structured total least squares for approximate polynomial operations. Master's thesis, University of Waterloo, 2004.

[7] Boutelle, J., Kau, S. P., and Marino, Jr., C. J. ICBM reentry vehicle navigation system development at Honeywell. *IEEE 1998 Position Location and Navigation Symposium* (Apr. 1998), 294–302.

[8] Burrows, M. L. Two-dimensional ESPRIT with tracking for radar imaging and feature extraction. *IEEE Transactions on Antennas and Propagation 52*, 2 (Feb. 2004), 524–532.

[9] Caicedoc, J. M., Dyke, S. J., and Johnson, E. A. Natural excitation technique and eigensystem realization algorithm for phase I of the IASC-ASCE benchmark problem: Simulated data. *Journal of Engineering Mechanics 130*, 1 (Jan. 2004), 49–60.

[10] Camp, W. W., Mayhan, J. T., and O'Donnell, R. M. Wideband radar for ballistic missile defense and range-doppler imaging of satellites. *MIT Lincoln Laboratory Journal 12*, 2 (2000), 267–280.

[11] Chang, C.-I., and Du, Q. Estimation of number of spectrally distinct signal sources in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing 42*, 3 (Mar. 2004), 608–619.

[12] Chen, C.-T. *Linear System Theory and Design*, 3rd ed. Oxford University Press, New York, Sept. 1998.

[13] Cuomo, K. M., Piou, J. E., and Mayhan, J. T. Ultrawide-band coherent processing. *IEEE Transactions on Antennas and Propagation 47*, 6 (June 1999), 1094–1107.

[14] Cuomo, K. M., Piou, J. E., and Mayhan, J. T. Ultra-wideband sensor fusion for BMD discrimination. In *The Record of the IEEE 2000 International Radar Conference* (May 2000), IEEE, pp. 31–34.

[15] Daum, F. E., and Fitzgerald, R. J. Decoupled Kalman filters for phased array radar tracking. *IEEE Transactions on Automatic Control 28*, 3 (Mar. 1983), 269–283.

[16] Demmel, J. W. The smallest perturbation of a submatrix which lowers the rank and constrained total least squares problems. *SIAM Journal of Numerical Analysis 24*, 1 (Feb. 1987), 199–206.

[17] DynCorp: Aerotherm Corporation. RV model coating RCS measurements program, RCS test plan sphere-cone target. Tech. Rep. F19628-95-C-002,CX-22751, MIT Lincoln Laboratory, PT. Mugu, Aug. 1999.

[18] Erling, J. G. *Statistical Performance Analysis of Source Localization in Multisource, Multiarray Networks*. PhD thesis, Pennsylvania State University, 2005.

[19] Ermolaev, V. T., Mal'tsev, A. A., and Rodyushkin, K. V. Statistical characteristics of the AIC and MDL criteria in the problem of estimating the number of sources of multivariate signals in the case of a short sample. *Radiophysics and Quantum Electronics 44*, 12 (Dec. 2001), 977–983.

[20] Golub, G. H., and van Loan, C. F. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis 17*, 6 (1980), 883–893.

[21] Golub, G. H., and Van Loan, C. F. *Matrix Computations*, 3rd ed. Johns Hopkins Univ. Press, 1996.

[22] Haardt, M., and Nossek, J. A. Simultaneous schur decomposition of several nonsymmetric matrices to achieve automatic pairing in multidimensional harmonic retrieval problems. *IEEE Transactions on Signal Processing 46*, 1 (Jan. 1998), 161–169.

[23] Harris, F. J. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE 66*, 1 (Jan. 1978), 51–83.

[24] Hatch, N., Holl, Jr., D., and Cyganski, D. Power-law scattering models and nonlinear parametric estimation for super-resolution radar. Tech. Rep. 1095, MIT Lincoln Laboratory, Apr. 2004.

[25] Hua, Y. *On Techniques for Estimating Parameters of Exponentially Damped / Undamped Sinusoids in Noise.* PhD thesis, Syracuse University, 1988.

[26] Hua, Y. Estimating two-dimensional frequencies by matrix enhancement and matrix pencil. *IEEE Transactions on Signal Processing 40*, 9 (Sept. 1992), 2267–2280.

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

[28] Ingwersen, P. A., and Lemnios, W. Z. Radars for ballistic missile defense research. *MIT Lincoln Laboratory Journal 12*, 2 (2000), 245–266.

[29] Juang, J. N., and Pappa, R. S. An eigensystem realization algorithm for modal parameter identification and model reduction. *Journal of Guidance, Control and Dynamics 8*, 5 (Sept. 1985), 620–627.

[30] Kailath, T. *Linear Systems.* Prentice-Hall, Inc., 1980.

[31] Kailath, T. *Lectures on Weiner and Kalman Filtering.* Springer Verlag, New York, 1981.

[32] Keller, J. Geometrical theory of diffraction. *Journal of the Optical Society of America* (1962).

[33] Kung, S. Y. A new identification and model reduction algorithm via singular value decomposition. In *Proc. Twelfth Ann. Asilomar Conference on Circuits, Systems and Computers* (Pacific Grove, CA, 1978), IEEE, pp. 705–714.

[34] Kung, S. Y., Arun, K. S., and Rao, B. D. State space and SVD based modeling methods for the harmonic retrieval problem. *Journal of the Optical Society Of America. 73*, 12 (1983), 1799–1811.

[35] Lemma, A. N., van der Veen, A.-J., and Deprettere, E. F. Analysis of joint angle-frequency estimation using esprit. *IEEE Transactions on Signal Processing 51*, 5 (May 2003), 1264–1283.

[36] Lemmerling, P., Moor, B. D., and Huffel, S. V. On the equivalence of constrained total least squares and structured total least squares. *IEEE Transactions on Signal Processing 44*, 11 (Nov. 1996), 2908–2911.

[37] Lu, B., Wei, D., Evans, B. L., and Bovik, A. C. Improved matrix pencil methods. In *Conference Record of the Thirty-Second Asilomar Conference on Signals, Systems and Computers* (Pacific Grove, CA, Nov. 1998), vol. 2, IEEE, pp. 1433–1437.

[38] Luk, F. T., and Qiao, S. A fast singular value algorithm for Hankel matrices. In *Proc. SPIE, Advanced Signal Processing Algorithms, Architectures, and Implementations XI* (Nov. 2003), vol. 4474, SPIE, pp. 254–261.

[39] Mensa, D. L. *High Resolution Radar Cross-Section Imaging*. Artech House, Inc., 1991.

[40] Moor, B. D. Total least squares for affinely structured matrices and the noisy realization problem. *IEEE Transactions on Signal Processing 42*, 11 (Nov. 1994), 3104–3113.

[41] Neeser, F. D., and Massey, J. L. Proper complex random processes with applications to information theory. *IEEE Transactions on Information Theory 39*, 4 (July 1993), 1293–1302.

[42] Paulraj, A., Roy, R., and Kailath, T. Estimation of signal parameters via rotational invariance techniques – ESPRIT. In *Proc. Nineteenth Ann. Asilomar Conference on Circuits, Systems and Computers* (Pacific Grove, CA, Nov. 1985), IEEE, pp. 83–89.

[43] Paulraj, A., Roy, R., and Kailath, T. A subspace rotation approach to signal parameter estimation. *Proceedings of the IEEE 74*, 7 (July 1986), 1044–1046.

[44] Pérez, P., Vermaak, J., and Blake, A. Data fusion for visual tracking with particles. *Proceedings of the IEEE 92*, 3 (Mar. 2004), 495–513.

[45] Piou, J. E. System realization using 2-d output measurements. In *Proceedings of the 2004 American Control Conference* (June 2004), vol. 3, pp. 2840–2845.

[46] Pridham, B. A., and Wilson, J. C. A study of damping errors in correlation-driven stochastic realizations using short data sets. *Probabilistic Engineering Mechanics 18*, 1 (Jan. 2003), 61–77.

[47] Rao, B. D., and Arun, K. S. Model based processing of signals: A state space approach. *Proceedings of the IEEE 80*, 2 (Feb. 1992), 283–309.

[48] Rissanen, J. Modeling by shortest data description. *Automatica 14* (1978), 465–471.

[49] Schwarz, G. Estimating the dimension of a model. *Annals of Statistics 6*, 2 (Mar. 1978), 461–464.

[50] van der Veen, A.-J., Vanderveen, M. C., and Paulraj, A. Joint angle and delay estimation using shift-invariance techniques. *IEEE Transactions on Signal Processing 46*, 2 (Feb. 1998), 405–418.

[51] Van Trees, H. L. *Optimum Array Processing, Part IV of Detection, Estimation, and Modulation Theory.* John Wiley & Sons, Inc., New York, 2002, pp. 925–928.

[52] Wright, C. R. *Multidimensional Direction of Arrival Estimator Performance Bounds and Optimization for Non-Stationary Noise.* PhD thesis, Worcester Polytechnic Institute, 1994.

[53] Xu, W., and Qiao, S. A divide-and-conquer method for the Takagi factorization. Technical Report CAS 05-01-SQ, Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada. L8S 4K1, Feb. 2005.

[54] Ziemer, R. E., Tranter, W. H., and Fannin, D. R. *Signals and Systems: Continuous and Discrete*, 3rd ed. Macmillan Publishing Company, 1993.