

DISSERTATION

JOINT SHAPE AND MOTION ESTIMATION FROM ECHO-BASED SENSOR DATA

Submitted by

Samuel J. Pine

Department of Mathematics

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2018

Doctoral Committee:

Advisor: Margaret Cheney

Daniel Bates  
Bailey Fosdick  
Chris Peterson

This work is licensed under a Creative Commons CC0 1.0 Universal Public Domain Dedication

License: <https://creativecommons.org/publicdomain/zero/1.0/>

## ABSTRACT

### JOINT SHAPE AND MOTION ESTIMATION FROM ECHO-BASED SENSOR DATA

Given a set of time-series data collected from echo-based ranging sensors, we study the problem of jointly estimating the shape and motion of the target under observation when the sensor positions are also unknown. Using an approach first described by Stuff et al. [1], we model the target as a point configuration in Euclidean space and estimate geometric invariants of the configuration. The geometric invariants allow us to estimate the target shape, from which we can estimate the motion of the target relative to the sensor position. This work will unify the various geometric-invariant based shape and motion estimation literature under a common framework, and extend that framework to include results for passive, bistatic sensor systems.

## ACKNOWLEDGEMENTS

In writing this thesis, I have received unwavering support from a number of sources. Most specific to the work presented here, I thank my advisor, Margaret Cheney, for her guidance toward interesting problems and her persistent advocacy for myself, her other students, and for graduate students in general. Through her, I met a new group of colleagues at Matrix Research. Thank you all for your knowledge, experience, and support, and thank you for providing the context and motivation to pursue this problem. I am grateful to the Air Force Research Lab for funding various aspects of this work.

Even when this work took a clear turn away from Dr.Cheney's field of expertise, she provided valuable advice and aided me in my search for answers. Some of that advice led me to my other committee members. Many of the ideas contained here started as a conversation with either Dan Bates or Chris Peterson, both of whom were happy to answer my questions and provide helpful suggestions as I stumbled my way through the unfamiliar territory of this problem. Thanks to both of them for fruitful discussions, as well as their excitement to share other interesting mathematics with me. Thanks also to Bailey Fosdick, for agreeing to participate on my committee despite her many new and exciting obligations as a parent.

Finally, thank you to my friends and family. I have no way to measure the laughter, the companionship, and the love they have shown me. They made no direct contributions to this thesis, but there is no doubt that I would not have completed it without their support.

## DEDICATION

*To my parents,  
who always wanted better for their children.*

## TABLE OF CONTENTS

ABSTRACT . . . . .	ii
ACKNOWLEDGEMENTS . . . . .	iii
DEDICATION . . . . .	iv
LIST OF TABLES . . . . .	vii
LIST OF FIGURES . . . . .	viii
Chapter 1    Introduction . . . . .	1
1.1        A Brief Introduction to SAR . . . . .	1
1.1.1    Electromagnetic Wave Propagation . . . . .	2
1.1.2    A Model for SAR . . . . .	3
1.1.3    An Imaging Operator . . . . .	6
1.2        A Short History of Shape and Motion Estimation . . . . .	8
1.2.1    Motion-Only Estimation . . . . .	9
1.2.2    Joint Shape and Motion Estimation . . . . .	10
1.3        Shape and Motion Estimation Techniques . . . . .	14
1.3.1    Affine Invariants Estimation . . . . .	15
1.3.2    Euclidean Invariants Estimation . . . . .	16
1.3.3    Monostatic Far-Field Case . . . . .	19
1.3.4    Monostatic Near-Field Case . . . . .	21
1.3.5    Bistatic Far-Field Case . . . . .	24
1.4        Related Techniques . . . . .	25
1.4.1    Interferometric Developments . . . . .	25
1.4.2    Image Factorization . . . . .	25
1.4.3    Low-Rank Subspace Decomposition . . . . .	26
Chapter 2    A Polynomial Systems Approach . . . . .	27
2.1        Expression as a Polynomial System . . . . .	27
2.1.1    Ellipse Constraints . . . . .	28
2.1.2    Alternative System . . . . .	29
2.1.3    Attempted Numerical Solution with Bertini . . . . .	30
Chapter 3    A Variety-Fitting Approach . . . . .	33
3.1        Monostatic, Near-Field Case . . . . .	35
3.1.1    Derivation of 2D Monostatic Near-Field Variety . . . . .	35
3.1.2    Estimation of Geometric Invariants in 2D . . . . .	37
3.2        Invariant Equation Discovery . . . . .	42
3.2.1    Recovering the Monostatic Near-Field Invariant . . . . .	42
3.3        Discovering a Bistatic Near-Field Invariant . . . . .	46
3.3.1    Elimination via Gröbner Basis Computation . . . . .	48
3.3.2    Elimination via Resultants . . . . .	52
3.4        Improvements to the Resultant Computation . . . . .	60

3.4.1	Determination of Resultant Degree . . . . .	60
3.4.2	Dixon Matrix Reduction . . . . .	63
Chapter 4	A Solution for the Passive Case . . . . .	67
4.1	An Invariant Equation for Passive SAR . . . . .	67
4.1.1	Geometric Invariant Discovery . . . . .	67
4.1.2	General Invariant Equation . . . . .	70
4.1.3	Computing Affine Invariants . . . . .	72
4.1.4	Computing Euclidean Invariants . . . . .	78
4.2	Numerical Results . . . . .	80
4.2.1	Computing Affine Invariants . . . . .	81
4.2.2	Computing Euclidean Invariants . . . . .	82
4.3	Further Work . . . . .	84
Bibliography	. . . . .	86
Appendix A	Conventions . . . . .	92
A.1	Symbol Reference . . . . .	92
A.2	Fourier Transform . . . . .	93
Appendix B	Approximations . . . . .	94
B.1	Born Approximation . . . . .	94
B.2	Far-Field Approximation . . . . .	95

## LIST OF TABLES

1.1	State of the search for invariants of radar data . . . . .	13
3.1	Dimension of BNF variety for various $N$ . . . . .	50
4.1	Convergence results for PNF affine invariant estimation . . . . .	84
4.2	Results for PNF Euclidean invariant estimation . . . . .	85

## LIST OF FIGURES

3.1	A generic 2D TDOA configuration . . . . .	34
3.2	Generic 2D monostatic configuration . . . . .	35
3.3	Generic 2D monostatic configuration, sensor contained in scatterers . . . . .	36
3.4	Image of a range map for $N = 3$ . . . . .	40
3.5	Dixon matrices for the BNF system . . . . .	59
3.6	Full-rank submatrices for the BNF system . . . . .	60
3.7	Input file for a standard run of Bertini . . . . .	62
3.8	Bertini output for projection . . . . .	62
4.1	Approximate rank objective function . . . . .	76
4.2	Illustration of rank objective function . . . . .	83
B.1	Illustration of far-field approximation . . . . .	96

# Chapter 1

## Introduction

### 1.1 A Brief Introduction to SAR

In a process similar to the echolocation of bats and dolphins, echo-based sensors can locate objects by propagating a signal through the ambient medium and listening for echoes. The time it takes the signal to travel through the medium, reflect off an object, and return to the sensor is proportional to the total distance travelled by the signal. Bats instinctively use this, for example, to estimate the relative location of flying insects. When we mimic this behavior using radio waves, we call the process RAdio Detection And Ranging, or just RADAR. This acronym has become so pervasive that we treat the acronym as a proper noun and neglect any fancy capitalization.

Though range finding remains the primary function of radar devices, the technology has taken on a number of other applications over its lifetime, including imaging [2]. Radar enjoys the capability to operate day or night, and thanks to the relatively long wavelengths of radio waves, radar systems are resistant to the effects of fog, smoke, and sand; they can even be used to image through cover including foliage and buildings. These characteristics give radar advantages over other optical imaging systems, and invite a large variety of applications. For example, radar is widely used for collision avoidance and navigation in air traffic control, as well as in the speed-monitoring radar used by police. Satellites equipped with radar systems monitor land movement, oil spills, and land use in agriculture. Military radar systems detect and recognize hostile vehicles, and assist in directing weapons to the desired target. Ground-penetrating radars are useful for guessing the composition of the Earth's crust, and we are all familiar with the weather radar that has become the primary tool of meteorologists.

### 1.1.1 Electromagnetic Wave Propagation

The electromagnetic waves used by radar systems are described by Maxwell's equations,

$$\nabla \times \mathcal{E} = -\frac{\partial \mathcal{B}}{\partial t} \quad (1.1a)$$

$$\nabla \times \mathcal{B} = \mu(\mathcal{J} + \epsilon \frac{\partial \mathcal{E}}{\partial t}) \quad (1.1b)$$

$$\nabla \cdot (\epsilon \mathcal{E}) = \rho \quad (1.1c)$$

$$\nabla \cdot (\mu \mathcal{B}) = 0, \quad (1.1d)$$

where  $\mathcal{E}(t, \mathbf{x})$  is the electric field,  $\mathcal{B}(t, \mathbf{x})$  is the magnetic field,  $\mathcal{J}(t, \mathbf{x})$  is the electric current density,  $\rho$  is the total electric charge, and the quantities  $\mu, \epsilon$  are the permeability and permittivity of the medium through which the waves propagate. If we take our ambient medium to be dry air and approximate this with the properties of free space, we can simplify Maxwell's equations with the constants  $\mathcal{J} = \rho = 0$  and set  $\mu = \mu_0, \epsilon = \epsilon_0$ . Take the curl of equation (1.1a) and combine this with equation (1.1b) to find

$$\nabla \times (\nabla \times \mathcal{E}) = -\mu_0 \epsilon_0 \frac{\partial^2 \mathcal{E}}{\partial t^2}.$$

Then, use the convenient vector identity,

$$\nabla \times (\nabla \times \mathcal{E}) = \nabla (\nabla \cdot \mathcal{E}) - \nabla^2 \mathcal{E},$$

to produce

$$\nabla^2 \mathcal{E} - \mu_0 \epsilon_0 \frac{\partial^2 \mathcal{E}}{\partial t^2} = 0. \quad (1.2)$$

The speed of light in free space is related to the permittivity and permeability by  $c_0^{-2} = \mu_0 \epsilon_0$ , so that we can see that each component of the electric field satisfies the scalar wave equation,

$$(\nabla^2 - c_0^{-2} \partial_t^2) \mathcal{E}(t, \mathbf{x}) = 0. \quad (1.3)$$

The propagation speed of an electromagnetic wave depends on the medium through which it travels, so to incorporate scattering from targets we modify the wave equation to include a spatially-dependent wave speed,

$$(\nabla^2 - c^{-2}(\mathbf{x})\partial_t^2) \mathcal{E}(t, \mathbf{x}) = 0. \quad (1.4)$$

Scattering is the result of perturbations in wave speed, so it is common to define the reflectivity function  $V(\mathbf{x}) = c_0^{-2} - c^{-2}(\mathbf{x})$ , with the understanding that this is not an entirely accurate model for electromagnetic scattering [2]. In truth, this model characterizes acoustic wave scattering. By reducing Maxwell's equations to the scalar wave equation in each component, we suppress some of the more complicated interactions of the electric field, the magnetic field, and the current density at the interface of materials.

### 1.1.2 A Model for SAR

In radar applications, it is convenient to think of the total electric field,  $\mathcal{E}^{\text{tot}}$  as the superposition of an incident field,  $\mathcal{E}^{\text{in}}$ , and the scattered field,  $\mathcal{E}^{\text{sc}}$ . The incident field is the field radiated by our antenna into free space, and so will satisfy the free space wave equation in (1.3). We expect this field to be incident on some collection of objects, thus the naming convention. Some of the energy from the incident field will scatter from those objects and return to the sensor as the scattered electric field. With the superposition property of waves we can define the scattered field as the total field minus the incident field,  $\mathcal{E}^{\text{sc}} = \mathcal{E}^{\text{tot}} - \mathcal{E}^{\text{in}}$ . As a result, the full problem to be solved is

$$(\nabla^2 - c^{-2}(\mathbf{x})\partial_t^2) \mathcal{E}^{\text{tot}}(t, \mathbf{x}) = j(t, \mathbf{x}) \quad (1.5a)$$

$$(\nabla^2 - c_0^{-2}\partial_t^2) \mathcal{E}^{\text{in}}(t, \mathbf{x}) = j(t, \mathbf{x}), \quad (1.5b)$$

where  $j(t, \mathbf{x})$  is the source term for the wave equation. We assume that the only power input comes from our antenna, so that  $\mathcal{J}(t, \mathbf{x}) = j(t, \mathbf{x})$  models the current density on the antenna.

We can introduce the reflectivity function,  $V(\mathbf{x}) = c_0^{-2} - c^{-2}(\mathbf{x})$ , into the system (1.5) by taking the difference of the two equations,

$$(\nabla^2 - c_0^{-2} \partial_t^2) \mathcal{E}^{\text{sc}}(t, \mathbf{x}) = -V(\mathbf{x}) \partial_t^2 \mathcal{E}^{\text{tot}}. \quad (1.6)$$

The goal of this radar imaging approach is to recover  $V$  from measurements of the scattered field. Objects in the scene will have different electromagnetic properties than the background, and so we can distinguish the location, shape, and orientation of these objects by forming an image of the scene's reflectivity.

We can solve the partial differential equation in (1.6) for  $\mathcal{E}^{\text{sc}}$  by convolving the right hand side with the outgoing Green's function,

$$g(t, \mathbf{x}) = \frac{\delta(t - c_0^{-1} |\mathbf{x}|)}{4\pi |\mathbf{x}|}. \quad (1.7)$$

The result is the Lippmann-Schwinger integral equation [2],

$$\mathcal{E}^{\text{sc}}(t, \mathbf{x}) = \int \int g(t - \tau, \mathbf{x} - \mathbf{z}) V(\mathbf{z}) \partial_\tau^2 \mathcal{E}^{\text{tot}}(\tau, \mathbf{z}) d\tau d\mathbf{z}. \quad (1.8)$$

The Lippmann-Schwinger equation is slightly nicer in the frequency domain, so define the Fourier-transform of  $\mathcal{E}$  to be

$$E(\omega) = \int e^{i\omega t} \mathcal{E}(t) dt. \quad (1.9)$$

Here, we use  $\nu$  to denote frequency; the abbreviation  $\omega = 2\pi\nu$  is commonly called the angular frequency. Additionally, we will let  $k = c_0^{-1}\omega$  be the wavenumber. Then (1.8) takes the form

$$E^{\text{sc}}(\omega, \mathbf{x}) = - \int G(\omega, \mathbf{x} - \mathbf{z}) V(\mathbf{z}) \omega^2 E^{\text{tot}}(\omega, \mathbf{z}) d\mathbf{z}, \quad (1.10)$$

where

$$G(\omega, \mathbf{x} - \mathbf{z}) = \frac{e^{ik|\mathbf{x}|}}{4\pi |\mathbf{x}|} \quad (1.11)$$

is the Fourier transform of the Green's function  $g$ . Both  $V$  and  $E$  are unknowns, so reconstructing  $V$  given sampled values of  $E^{\text{sc}}$  is a nonlinear inverse problem. It is common to linearize this

problem by making the *Born approximation*, which amounts to replacing the total electric field with the incident electric field. While the Born approximation greatly simplifies the problem at hand, it does not take into account the effects of multiple scattering. In some cases, this can result in poor reconstructions of the scene reflectivity. See appendix B.1 for more discussion on the Born approximation.

Assuming we can estimate the current density on our antenna, we can solve equation (1.5b) for the incident electric field by convolving with the outgoing Green's function. In the frequency domain, the PDE takes the form

$$(\nabla^2 + k^2)E^{\text{in}}(\omega, \mathbf{x}) = J(\omega, \mathbf{x} - \mathbf{x}_0), \quad (1.12)$$

when we center the antenna at  $\mathbf{x}_0$ . Then we can solve for  $E^{\text{in}}$ ,

$$E^{\text{in}}(\omega, \mathbf{x}) = \int \frac{e^{ik|\mathbf{x}-\mathbf{y}|}}{4\pi|\mathbf{x}-\mathbf{y}|} J(\omega, \mathbf{y} - \mathbf{x}_0) d\mathbf{y}. \quad (1.13)$$

Now, it follows from (1.10) that the Born-approximated scattered field (denoted by the subscript) is

$$E_B^{\text{sc}}(\omega, \mathbf{x}) = - \int \frac{e^{ik|\mathbf{x}-\mathbf{z}|}}{4\pi|\mathbf{x}-\mathbf{z}|} \omega^2 E^{\text{in}}(\omega, \mathbf{z}) V(\mathbf{z}) d\mathbf{z}. \quad (1.14)$$

If we assume for simplicity that our transmit and receive antennas are the same, then we can collect data at the location  $\mathbf{x}_0$ ,

$$E_B^{\text{sc}}(\omega, \mathbf{x}_0) = \int e^{2ik|\mathbf{x}_0-\mathbf{z}|} A(\omega, \mathbf{x}_0, \mathbf{z}) V(\mathbf{z}) d\mathbf{z}, \quad (1.15)$$

where we have grouped the effects of the incident field, the geometric spreading factor  $(4\pi|\mathbf{x}-\mathbf{z}|)^{-1}$ , and the dependence on the chosen waveform into  $A$ . If our antenna is moving, it is more appropriate to parameterize the antenna position with  $\gamma(s)$ , so we replace  $\mathbf{x}_0$ , obtaining

$$E_B^{\text{sc}}(\omega, s) = \int e^{2ik|\gamma(s)-\mathbf{z}|} A(\omega, s, \mathbf{z}) V(\mathbf{z}) d\mathbf{z}. \quad (1.16)$$

Since the speed of light is so much greater than the speed of the antenna and objects we might care to image, it is common in radar applications to use separate variables to denote *slow time* and *fast time*. For example, we think of  $t$  as a useful measure of the time it takes for an EM wave to travel to a target and back to the receiver, and  $s$  will be in units more suited to measuring the amount of time it takes our antenna to move a few meters.

### 1.1.3 An Imaging Operator

The data model we describe in (1.16) can be written as an operator on the reflectivity function,

$$\eta(\omega, s) = F[V](\omega, s). \quad (1.17)$$

Here we use  $\eta$  instead of  $E_B^{\text{sc}}$ , to note that the actual data collected is slightly different than  $E_B^{\text{sc}}$  due to matched filtering. To isolate  $V$ , we want to be able to invert the operator  $F$ . One strategy is to find the adjoint operator to  $F$ , and modify that map to approximate the inverse, in the spirit of the inverse radon transform [2]. We can define the Hermitian inner products

$$\langle g, H \rangle_{\omega, s} = \int g(\omega, s) H^*(\omega, s) d\omega ds, \quad (1.18)$$

$$\langle G, h \rangle_{\mathbf{z}} = \int G(\mathbf{z}) h^*(\mathbf{z}) d\mathbf{z}, \quad (1.19)$$

so that the adjoint we seek is the operator  $F^\dagger$  defined by

$$\langle g, Fh \rangle_{\omega, s} = \langle F^\dagger g, h \rangle_{\mathbf{z}} \quad (1.20)$$

for all  $g(\omega, s)$  and  $h(\mathbf{z})$ . Using our definition of  $F$  from (1.17), we want  $F^\dagger$  such that

$$\int f(\omega, s) \eta^*(\omega, s) d\omega ds = \int (F^\dagger f)(\mathbf{x}) V^*(\mathbf{z}) d\mathbf{z}. \quad (1.21)$$

It follows that

$$\int \left( \int f(\omega, s) e^{-2ik|\gamma(s)-\mathbf{z}|} A^*(\omega, s, \mathbf{z}) d\omega ds \right) V^*(\mathbf{z}) d\mathbf{z} = \int (F^\dagger f)(\mathbf{x}) V^*(\mathbf{z}) d\mathbf{z}, \quad (1.22)$$

so that our adjoint is

$$F^\dagger g(\mathbf{z}) = \int e^{-2ik|\gamma(s)-\mathbf{z}|} A^*(\omega, s, \mathbf{z}) g(\omega, s) d\omega ds. \quad (1.23)$$

Then we can define an approximate inverse operator of the form

$$B[\eta] = \int e^{-2ik|\gamma(s)-\mathbf{z}|} Q(\omega, s, \mathbf{z}) \eta(\omega, s) d\omega ds. \quad (1.24)$$

This operator is sometimes referred to as a filtered backprojection operator, and has a similar physical interpretation to the backprojection operator defined for the Radon transform. The exact specification of the filter  $Q$  is not important for this exposition, but we do note that the antenna position,  $\gamma(s)$ , is necessary for construction of the imaging operator in (1.24). When we know the position of the platform relative to the scene, we can reconstruct a reasonable image of the scene,  $I(\mathbf{z}) = B[\eta]$ , up to the resolution of the system.

Typically, the resolution of a radar image is restricted by the antenna aperture, which is related to the physical size of the antenna. If we return to equation (1.16), we can infer that this would be the case. Notice that we measure the scattered field at each position  $\mathbf{x}_0$ , corresponding to an element of the antenna, and that each measurement is actually the inner product of  $V(\mathbf{z})$  with some function  $f_x(\mathbf{z}) = e^{2ik|\mathbf{x}-\mathbf{z}|} A(\omega, \mathbf{x}, \mathbf{z})$ . Each distinct inner product gives us more information about the reflectivity function  $V$ . Similarly, collecting data from a moving antenna amounts to measuring the inner product of  $V$  with the functions  $f_s(\mathbf{z}) = e^{2ik|\gamma(s)-\mathbf{z}|} A(\omega, s, \mathbf{z})$ . In this case, the antenna is said to sweep out a synthetic aperture, resulting in the term *synthetic aperture radar* (SAR). As mentioned before, knowledge of the position of the antenna relative to the scene is essential for producing an image. If estimates of  $\gamma(s)$  are poor, our filter will be mismatched to the data and the resulting SAR image will be blurred [3]. Since the SAR imaging scheme depends on measuring the

scattered electric field over time, any unaccounted motion from poor GPS measurements, moving objects in the scene, etc., will degrade the image quality.

Though the SAR image reconstruction scheme depends only on the *relative* motion of the antenna to the target, the field is commonly divided into two modes. Synthetic aperture radar, or SAR, refers to a moving radar platform imaging a static target. In this case, the target is usually some region on the ground or ocean (called the *scene*) and the antenna is attached to a plane or satellite flying over the region. The other mode is *Inverse synthetic aperture radar* or ISAR. This mode refers to a stationary radar platform and a moving target; for example, ISAR is used for imaging moving airplanes, satellites, ships, and spacecraft. Observing a moving target from a stationary platform is mathematically equivalent to viewing a stationary target from a moving platform (or even a moving target *and* a moving platform), so we rarely note the distinction between the two problems for our derivations.

## 1.2 A Short History of Shape and Motion Estimation

In SAR imaging applications, any unknown motion between the radar platform and the target will blur the constructed image, analogous to the blurring we see in long-exposure cameras. The unknown motion might be due to inexact measurements of the radar platform's position by the onboard GPS and inertial measurement unit, or it could be the result of a moving vehicle in an otherwise stationary scene. While there exist specialized radar systems for separating moving targets from background clutter, these moving target indicator (MTI) systems are not adapted for operating on available SAR data [4, 5]. Further, the need to make flying radar units as light and inexpensive as possible makes it desirable to reduce the amount of specialized hardware present on each unit. Improved signal processing algorithms may allow us to estimate the relative positions of the radar platform and target without expensive hardware additions, or can aid in regularizing the noisy measurements of existing hardware [3].

### 1.2.1 Motion-Only Estimation

There are a number of techniques for forming focused images of moving targets from SAR data, all with varying levels of effectiveness. The most common approach is to form one or more initial images of the target, and then estimate the relative motion of the target from the blurred images. One then focuses the images by adjusting the imaging operator with the motion estimate.

As far back as 1967, Brown noted the complications arising from the interaction of the transmitted signal and a rotating target. The resolution of a radar system depends on a number of factors, including the bandwidth and duration of the transmitted signal. We often quantify the resolution in terms of the *range* resolution and the *angular* resolution. If we have two targets near each other in space, the range resolution is the smallest range difference such that we can separate the two objects. Similarly, the angular resolution is the minimum angle difference such that we can separate two objects at the same range. Together, the range and angular resolutions define a polar grid of *resolution cells* around the radar platform. Brown explained that the current Fourier-transform signal processing techniques result in poor image quality when we integrate over a slow time window long enough that target points move from one resolution cell to another [6]. Walker then provided a technique to partially compensate for target points that move across resolution cells with his polar storage format for range-Doppler data [7]. His approach worked well for slowly translating, rotating targets, as it was designed for imaging other planets from Earth. Since the rotation rate is both uniform and possible to estimate, this problem is less complex than for a target with erratic motion.

Unexpected movement by the target results in a mismatch between the image operator used for reconstruction and the data. The mismatch is partially due to an unexpected phase adjustment in the data, and many common approaches for focusing SAR images exploit phase adjustments to compensate for unanticipated motion. In 1995, Fienup and Kowalczyk [8] devised a technique that would detect the presence of a moving object directly from the blurring present in an image; they reasoned that the most prominent phase errors will occur in regions of the image containing a moving target. To locate regions with relatively large phase errors, they segment the image into

patches and use a phase-correction technique to automatically focus (autofocus) each region. Any region that experiences a significantly large change in phase or increase in image sharpness is likely to contain a moving object. Fienup refined this technique for his 2001 paper [4]. Barbarosa and Scaglione [9] developed a similar approach, except that they estimate the phase errors in each patch with customized version of phase-gradient autofocus. Another moving target detection approach, developed by Moreira and Keydel [10], takes multiple SAR images formed over shorter integration times (a smaller synthetic aperture), and cross-correlates the power spectra of adjacent images to estimate the phase shift in the images. By modeling this phase shift as a quadratic function of the tangential and radial velocity of the target, they can estimate the motion of the target.

In a pair of papers from the Environmental Research Institute of Michigan (ERIM), we start to see the beginnings of the modern shape and motion estimation procedure. Werness et al. [11, 12] describe a motion compensation method based on tracking prominent points in the image. First, a single prominent scatterer is located in range. The SAR data is then motion-compensated to place this point at the center of the scene, and two more prominent scatterers are extracted from the compensated data. With range estimates for the three scatterers, one can estimate the rotation rate with relatively small error.

### **1.2.2 Joint Shape and Motion Estimation**

We have briefly investigated methods that rely on some spatial invariance in the underlying phase errors in a SAR image to generate a two-dimensional (2D) motion estimate (range and cross-range) for the target. The problem is, the phase errors are actually the result of 3D motion by the target. In order to reliably produce focused moving target images, it is necessary to estimate the motion of the target in the full three-dimensional (3D) space.

In 1994, researchers from ERIM issued an outline of a signal processing approach that would make it possible to track any type of motion in two or three dimensions, requiring only the radar data [1]. Given a set of range measurements for at least four scattering centers taken at geometrically diverse viewing aspects, Stuff et al. claimed to be able to reconstruct the geometric

arrangement of the scatterers by exploiting invariants in the range data. They claimed that the ranges from a radar to scattering centers on any rigid body must be constrained to a submanifold of the space of possible range observations, and that this manifold then determines the arrangement of the scattering centers up to rotation or reflection. With this arrangement, the motion of the object can then be determined.

From this original paper, Stuff would go on to develop the approach in a series of follow-up papers. The first of these appeared in 1999, and explained that there are functions on the range data that are invariant to the motion of the object [13]. These functions map to quantities that depend on the *Euclidean invariants* of the target, such as the fixed distances between scatterers, the angles formed by the scatterer configuration, or the area contained by the scatterer configuration. Stuff referred to these quantities as Euclidean invariants because they remain unchanged when isometries of Euclidean space are applied to the scatterer configuration describing the target. As an example of one such function, Stuff derives an invariant equation for far-field range data collected from a generic target with four non-coplanar scattering centers using the properties of determinants. In a subsequent paper [14], Stuff generalizes the invariant equation for far-field data to a target with  $N$  scatterers, finding

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^N \omega_{ij} (\rho_i(t) - \rho_j(t))^2 = 1. \quad (1.25)$$

Here,  $\rho_i(t)$  denotes the range to the  $i^{th}$  scattering center at time  $t$ , and  $\omega_{ij}$  are the geometric invariants of the target. This work would ultimately lead to Stuff's dissertation [15], in which he described the geometric invariant theory for monostatic, far-field range measurements and developed statistical techniques for estimating the geometric invariants. A condensed version of the deterministic theory was published the year after, along with a numerical example with noisy range data and some comments on the performance of the algorithm and potential problems in practice [16]. It was also around this time that this shape and motion approach, along with its various subsystems, was dubbed the 3D Motion And Geometric Information (3DMAGI) system.

Stuff et al. would publish another paper in 2004, detailing the image reconstruction process for a moving vehicle [17]. In ISAR, the collected data define a surface in Fourier space, with the

motion of the object determining the size and shape of this surface. In the far-field, the imaging operator (1.24) simplifies to the Fourier transform, and so the image formation process amounts to taking a Fourier transform of the data. The data collection manifold is a 2D surface embedded in 3D Fourier space, so that a simple 2D Fast Fourier Transform (FFT) is not enough to transform the data. Instead, knowledge of the object motion allows Stuff to interpolate the data surface to a uniform grid, and then utilize the 2D FFT. The National Ground Intelligence Center data was collected from a scale model over a limited aperture, but even this indoor data set displayed the difficulty of generating 3D images of non-cooperative moving targets.

Though the 3DMAGI approach was conceived as part of an effort to track moving objects in SAR images, the approach naturally lends itself to automatic target recognition (ATR). Since the reconstructed geometric invariants are independent of the target's orientation and position, such an ATR algorithm would avoid the extremely difficult problem of recognizing the same target from different viewing angles. Additionally, the algorithm can extract geometric invariants for previously unknown target shapes, giving the potential for an ATR system to learn new target classifications. In his 2003 dissertation, Meyer explores the possibilities of a 3DMAGI-based ATR system [18].

The development of the 3DMAGI system slowed for a time, to be revived in 2008 with a joint paper from Ferrara, Jackson, and Stuff [19]. This paper used the 3DMAGI approach for motion estimation, and then introduced a variant of the CLEAN algorithm which took advantage of a better, Non-Uniform Fast Fourier Transform (NUFFT). Ferrara, Arnold, and Stuff would go on to revise and simplify Stuff's original geometric invariant approach [20]. The new approach eliminates the need for any uniqueness constraints on the invariants, and essentially reduces the shape and motion reconstruction process to two singular value decomposition (SVD) computations. In the same year, Ferrara and Arnold identified an invariant functional on near-field radar data [3]. Up to this point, all of the 3DMAGI literature was concerned with targets so far from the sensor that the curvature of electromagnetic wave fronts could be ignored in the analysis (appendix B.2 elaborates on the far-field approximation). The new invariant functional allowed the 3DMAGI system

**Table 1.1:** The current solutions and partial solutions for each case are cited in the appropriate box. It is unknown if a bistatic, near-field invariant exists.

	Near-Field	Far-Field
Monostatic	[3]	[20]
Passive	?	[21]
Bistatic	?	[21]

to accommodate near-field radar data collection, and naturally led to a new question: can we find invariant functionals on radar data collected in other sensor configurations?

For shape and motion estimation purposes, there are four types of sensor configurations. Data is either collected by a *monostatic* system, in which the signal transmitter and receiver are collocated, or by a *multistatic* system, in which there are possibly multiple transmitters and receivers, not all collocated. The simplest multistatic sensor configuration is a *bistatic* configuration, in which we have one transmitter and one receiver, in different locations. Further, the sensors are either close enough to the imaging target that we must account for the curvature of the EM waves in our range model, or far enough away that we can ignore such curvature and model the system with plane waves. The two cases are referred to as the *near-field* and the *far-field*, respectively. We can further distinguish between multistatic cases based on the movement of the sensors. In some scenarios, either the transmitter or receiver will remain fixed relative to the scene under interrogation. For instance, a receiver may use the signal emitted by a television tower to image a patch of ground. Such a data collection is referred to as *passive*, and so we will call the bistatic case with one fixed sensor the passive case. A solution for the usual bistatic case would extend naturally to the general multistatic case, and would also solve the passive bistatic case. A partial answer to our question above came in 2013, when Arnold, Ferrara, and Parker found a relation that suffices to identify the unknown motion of a target in the bistatic, far-field sensor configuration [21]. It is unknown, however, whether such an invariant exists for bistatic, near-field radar data. The current state of the search for invariants of the range data is summarized in Table 1.1.

### 1.3 Shape and Motion Estimation Techniques

Given a set of SAR data, suppose we wish to estimate the 3D shape and motion of a moving target relative to the sensor. For each EM pulse transmitted, some portions of the target, such as corners, spots with bare metal, etc., will be especially good at scattering the waves back toward the receiver. It is possible to track the range of these prominent points at each time, so that the prominent scatterers serve as a set of landmarks on the object. When these landmarks do not move relative to each other, we say that the target is *rigid*. With this in mind, we model a generic target with a point cloud of  $N$  targets, and we use *shape* to refer to the relative positions of the points. If  $d$  is the dimension of our target (2D or 3D), let  $\mathbf{s}_n \in \mathbb{R}^d$  be a column vector describing the Euclidean coordinates of the  $n^{\text{th}}$  scatterer when the target is centered at the origin. Grouping the coordinates together gives the  $N \times d$  *shape matrix*,

$$S = \begin{bmatrix} \mathbf{s}_1^T \\ \vdots \\ \mathbf{s}_N^T \end{bmatrix}, \quad (1.26)$$

encoding the shape of our point cloud. For a 3D target,  $d = 3$ , we assume that the scattering centers will be non-coplanar, so that  $S$  is full rank. Without loss of generality, we suppose that our sensor is fixed at the origin and attribute all of the relative motion to the target. For a rigid target, this means that the Euclidean coordinates of each of the scatterers at any time  $t_l$  will be some rotated and translated version of the original coordinate from  $S$ . We then model the position of the  $n^{\text{th}}$  scatterer at time  $t_l$  with

$$\mathbf{x}_{nl}^T = \mathbf{s}_n^T \mathcal{O}_l + \boldsymbol{\tau}_l^T, \quad (1.27)$$

where  $\mathcal{O}(t_l) = \mathcal{O}_l \in SO(3)$  represents a rotation and  $\boldsymbol{\tau}(t_l) = \boldsymbol{\tau}_l \in \mathbb{R}^d$  represents a translation. We can concatenate the positions from each scatterer to form the  $N \times 3$  matrix

$$X_l = S\mathcal{O}_l + \mathbf{1}\boldsymbol{\tau}_l^T, \quad (1.28)$$

where  $\mathbf{1}$  is the vector of ones. Now, suppose we measure the distance between the sensor and each of the  $N$  scatterers in the point cloud at  $L$  distinct time instants. The *range* to the  $n^{\text{th}}$  scatterer at time  $l$  will be  $\rho_{nl} = \|\mathbf{x}_{nl}\|$ , so that at the  $l^{\text{th}}$  instant, we denote the vector of range measurements to each scatterer as  $\boldsymbol{\rho}_l \in \mathbb{R}^N$ . Concatenating these range vectors gives the  $N \times L$  range data matrix,

$$R := [\boldsymbol{\rho}_1 \dots \boldsymbol{\rho}_L]. \quad (1.29)$$

The transpose,  $R^T$ , is commonly used because each column of this matrix describes the entire range history for a single scatterer. We sometimes refer to such a range history as a *track*, and call  $R^T$  the *track matrix*. Given the matrix  $R$ , our goal is to reconstruct the shape and motion of the target relative to our sensor.

### 1.3.1 Affine Invariants Estimation

Ultimately, we are interested in recovering invariants of the target, such as the pairwise distances between scatterers, the angles between scatterers, or, equivalently, the full shape matrix  $S$ . In the approaches detailed by Ferrara, Arnold, and Stuff [3, 20], however, we find an intermediate set of invariants that are sufficient for motion estimation in some cases. The matrix  $S$  can be decomposed into the product of a unitary matrix  $V$  and an invertible matrix  $M$ ,

$$S = VM, \quad (1.30)$$

where the columns of the matrix  $V$  describe an orthonormal basis for the range of  $S$ , and the invertible matrix  $M$  contains all of the scaling information from  $S$ . We'll see later that the current shape estimation method generates a factorization of the matrix  $R$  to determine  $S$  and the target motion, so that knowledge of the matrix  $V$  provides sufficient estimates of the target rotation. Since this is the case, quantities that depend only on  $V$  are called *affine invariants*, whereas quantities that depend on the full matrix  $S$  are *Euclidean invariants*.

To estimate affine invariants of the target from range data, we must first identify a mapping,  $f : \mathbb{R}^N \rightarrow \mathbb{R}^N$  on the range data such that  $f(\boldsymbol{\rho}_l) \in \text{range}(S)$ . With this mapping, we can write

$$f(\boldsymbol{\rho}_l) = S\mathbf{c}_l. \quad (1.31)$$

With some abuse of notation, let  $f(R)$  denote the function  $f$  applied to each column of  $R$ . Given the singular value decomposition of  $f(R) = VAQ^T$ , it follows that  $\text{range}(V) \subset \text{range}(S)$ , since the left singular vectors of  $f(R)$  form an orthonormal basis for the span of the columns of  $f(R)$ . If  $\text{rank}(f(R)) = d$ , then  $\text{range}(V) = \text{range}(S)$ , and we have found the full set of affine invariants. Prior to the improvements made in 2009 [20], computation of the affine invariants was an important step in enforcing uniqueness constraints on the Euclidean invariants, using the projection matrix onto the range of  $S$ ,

$$P := S(S^T S)^{-1} S^T = VV^T, \quad (1.32)$$

where the equivalence follows because  $V$  is an orthonormal basis for the columns of  $S$ .

### 1.3.2 Euclidean Invariants Estimation

Since  $S$  is full-rank, we can solve (1.31) for  $\mathbf{c}_l$ ,

$$\mathbf{c}_l = (S^T S)^{-1} S^T f(\boldsymbol{\rho}_l), \quad (1.33)$$

so that

$$\mathbf{c}_l^T \mathbf{c}_l = f(\boldsymbol{\rho}_l)^T S (S^T S)^{-2} S^T f(\boldsymbol{\rho}_l). \quad (1.34)$$

The matrix  $\Omega := S(S^T S)^{-2} S^T$  depends only on  $S$ , and we can solve (1.34) for the entries of  $\Omega$  given enough pairs of  $\|\mathbf{c}_l\|^2$  and  $f(\boldsymbol{\rho}_l)$ . Note that the matrix  $\Omega$  is unique only up to rotated versions of  $S$ , since  $\Omega = S(S^T S)^{-2} S^T = S(S^T S)^{-2} S^T$ , where  $S = S\mathcal{O}$  and  $\mathcal{O} \in O(3)$  is an arbitrary rotation or reflection. While  $P = VV^T$  is an affine invariant of the target, substituting  $S = VM$  into the definition for  $\Omega$  yields

$$\begin{aligned}
\Omega &= S(S^T S)^{-2} S^T \\
&= VM \underbrace{(M^T V^T V M)^{-2}}_I M^T V^T \\
&= V \underbrace{MM^{-1}}_I (M^T)^{-1} M^{-1} \underbrace{(M^T)^{-1} M^T}_I V^T \\
&= V(MM^T)^{-1} V^T.
\end{aligned} \tag{1.35}$$

The dependence of  $\Omega$  on  $M$  implies that  $\Omega$  is a Euclidean invariant of the target scatterer configuration.

If we suppose that the symmetric matrix  $\Omega$  has the singular value decomposition (SVD)

$$\Omega = U\Sigma U^T, \tag{1.36}$$

then the Moore-Penrose pseudo inverse of  $\Omega$  has the property

$$\Omega^\dagger = SS^T = U\Sigma^{-1}U^T. \tag{1.37}$$

This suggests the factorization  $S = U\Sigma^{-\frac{1}{2}}$ . Inputting this factorization to the definition of the projection matrix (1.32) shows that

$$P = UU^T, \tag{1.38}$$

which implies  $UU^T = VV^T$ , or that  $U = V\mathcal{O}$ ,  $\mathcal{O} \in O(3)$ . Writing  $S = V\mathcal{O}\Sigma^{-\frac{1}{2}}$  and substituting this into (1.34) gives

$$\begin{aligned}
\mathbf{c}_l^T \mathbf{c}_l &= f(\boldsymbol{\rho}_l)^T V \mathcal{O} \Sigma \mathcal{O}^T V^T f(\boldsymbol{\rho}_l) \\
&= \mathbf{b}_l^T \mathcal{O} \Sigma \mathcal{O}^T \mathbf{b}_l \\
&= \mathbf{b}_l^T W \mathbf{b}_l,
\end{aligned} \tag{1.39}$$

where  $\mathbf{b}_l = V^T f(\boldsymbol{\rho}_l)$  and  $W = \mathcal{O} \Sigma \mathcal{O}^T$ . The  $\mathbf{b}_l$  are known, since we have  $V$  from the SVD of  $f(R)$ . There remain only six unknowns left to determine in the symmetric matrix  $W$ . Vectorizing

the matrix as  $\mathbf{w} = [w_{11}, w_{12}, w_{13}, w_{22}, w_{23}, w_{33}]^T$ , we can rewrite (1.39) as

$$\|\mathbf{c}_l\|^2 = \begin{bmatrix} b_1^2 & 2b_1b_2 & 2b_1b_3 & b_2^2 & 2b_2b_3 & b_3^2 \end{bmatrix} \mathbf{w}. \quad (1.40)$$

Collecting these equations across all times  $t_l$ , we can create the system

$$Z\mathbf{w} = \boldsymbol{\gamma}, \quad (1.41)$$

where

$$Z = \begin{bmatrix} B_1 \odot B_1 \\ 2B_1 \odot B_2 \\ 2B_1 \odot B_3 \\ B_2 \odot B_2 \\ 2B_2 \odot B_3 \\ B_3 \odot B_3 \end{bmatrix}^T, \boldsymbol{\gamma} = \begin{bmatrix} \|\mathbf{c}_1\|^2 \\ \vdots \\ \|\mathbf{c}_L\|^2 \end{bmatrix}. \quad (1.42)$$

$B_1, B_2$ , and  $B_3$  are the rows of the matrix  $B = V^T f(R)$ , and  $\odot$  denotes element-wise multiplication (the Hadamard product). Note that if we write the SVD of  $f(R) = VAQ^T$ , then  $B = AQ^T$  so that we can avoid an extra matrix multiplication in our computation by recycling the results of the first SVD computation.

After solving for  $W$ , simply multiply to find  $\Omega = VWV^T$ . For our computations above, we assumed that  $\Omega = U\Sigma U^T$ ; one recovers this form by taking the SVD of  $VWV^T$ , and generates a shape exemplar with

$$\tilde{S} = U\Sigma^{-\frac{1}{2}}, \quad (1.43)$$

along with its corresponding motion,

$$\tilde{C} = \Sigma^{\frac{1}{2}} U^T f(R). \quad (1.44)$$

The matrix  $\tilde{C}$  is composed of the columns  $c_l$ . The specific interpretation of these vectors depends on the radar configuration, but in general the matrix  $\tilde{C}$  characterizes the motion of the target over the viewing interval. In some cases, estimation of the scaling parameters in  $\Sigma$  is unstable, and so we could instead consider a scaled, rotated shape exemplar consisting of only the affine invariants,

$$\tilde{S}_a = V, \quad (1.45)$$

along with its corresponding motion,

$$\tilde{C}_a = V^T f(R). \quad (1.46)$$

At this point, it is clear that the revised method presented by Ferrara, Arnold, and Stuff [20] is a matrix factorization method, where the derivation of the function  $f$  and the estimation of Euclidean invariants is the key to expanding this method to other radar configurations. We see that generating our affine shape and motion estimates in (1.45), (1.46), amounts to taking the SVD of  $f(R)$  and grouping the factors,

$$f(R) = \underbrace{V}_{\tilde{S}_a} \underbrace{AQ^T}_{\tilde{C}_a} = \underbrace{V\mathcal{O}\Sigma^{-\frac{1}{2}}}_{\tilde{S}} \underbrace{\Sigma^{\frac{1}{2}}\mathcal{O}^T AQ^T}_{\tilde{C}}, \quad (1.47)$$

and that the Euclidean shape and motion estimates in (1.43), (1.44), are rotated and scaled versions of the affine shape and motion estimates. With this general outline of shape and motion estimation, we can now define the specific mapping  $f$  for each radar configuration.

### 1.3.3 Monostatic Far-Field Case

In the monostatic, far-field case, we assume that the target is so far from the platform that EM wavefronts are nearly planar in the region of the target. Given the scatterer positions at time  $l$  from (1.28), the range to the  $n^{\text{th}}$  scattering center is

$$\rho_{nl} = \|\mathbf{x}_{nl}\| = \sqrt{(\mathbf{s}_n^T \mathcal{O}_l + \boldsymbol{\tau}^T)(\mathcal{O}_l^T \mathbf{s}_n + \boldsymbol{\tau})}. \quad (1.48)$$

Accepting the far-field approximation simplifies the model for the range to the  $n^{\text{th}}$  scatterer at time  $l$  to

$$\rho_{nl} \approx \mathbf{s}_n^T \mathcal{O}_l \hat{\boldsymbol{\tau}}_l + \|\boldsymbol{\tau}_l\|, \quad (1.49)$$

so that when we ignore the far-field approximation error, the full vector of range measurements to each scatterer at time  $l$  is

$$\boldsymbol{\rho}_l = S \mathcal{O}_l \hat{\boldsymbol{\tau}}_l + \|\boldsymbol{\tau}_l\| \mathbf{1}. \quad (1.50)$$

$\hat{\boldsymbol{\tau}}_l$  denotes the unit vector in the direction of  $\boldsymbol{\tau}_l$ . See appendix B.2 for more explanation of the far-field approximation. Notice that  $\|\boldsymbol{\tau}_l\|$  is independent of the scatterer number, so that subtracting the mean of each range vector will remove the second term from (1.50). One way to implement this subtraction is with the centering matrix,

$$C = I_N - \frac{1}{N} \mathbf{1}\mathbf{1}^T. \quad (1.51)$$

The column sums of  $S$  are zero by definition, so  $\mathbf{1}^T S = 0$ , and the first term of (1.50) is unaffected by the centering,

$$f(\boldsymbol{\rho}_l) := C \boldsymbol{\rho}_l = S \mathcal{O}_l \hat{\boldsymbol{\tau}}_l. \quad (1.52)$$

Notice that  $C \boldsymbol{\rho}_l \in \text{range}(S)$ , so we have found the desired mapping of the range data into  $\text{range}(S)$ . This mapping is advantageous in that it specifies the values of  $\|\mathbf{c}_l\|^2$  from (1.34), since it is natural to take  $\mathbf{c}_l = \mathcal{O}_l \hat{\boldsymbol{\tau}}_l$ , and

$$\begin{aligned} \mathbf{c}_l^T \mathbf{c}_l &= \hat{\boldsymbol{\tau}}_l^T \mathcal{O}_l^T \mathcal{O}_l \hat{\boldsymbol{\tau}}_l \\ &= \hat{\boldsymbol{\tau}}_l^T \hat{\boldsymbol{\tau}}_l \\ &= 1. \end{aligned} \quad (1.53)$$

### 1.3.4 Monostatic Near-Field Case

In the monostatic, near-field case, the mapping into  $range(S)$  is more complicated. Ferrara et al. [3] found that column-wise differences of the squared range data fall in  $range(S)$  and described the technique for solving for the invariants in  $W$ . Without the far-field assumption to simplify our model for the range measurements, we have

$$\rho_{nl}^2 = \mathbf{x}_{nl}^T \mathbf{x}_{nl} = \|\mathbf{s}_n^T\|^2 + 2\mathbf{s}_n^T \mathcal{O}_l \boldsymbol{\tau}_l + \|\boldsymbol{\tau}_l\|^2. \quad (1.54)$$

Again, concatenating the range measurements over the  $N$  scatterers gives the vector

$$\boldsymbol{\rho}_l^2 = \begin{bmatrix} \|\mathbf{s}_1^T\|^2 \\ \vdots \\ \|\mathbf{s}_N^T\|^2 \end{bmatrix} + 2S\mathcal{O}_l \boldsymbol{\tau}_l + \|\boldsymbol{\tau}_l\|^2 \mathbf{1}. \quad (1.55)$$

In this case,  $\boldsymbol{\rho}_l^2$  denotes that each element of  $\boldsymbol{\rho}_l$  is squared. Notice that the last term is once again independent of scatterer number, and so can be removed by subtracting the mean from each pulse. The result of applying the centering matrix,  $C$ , is

$$C\boldsymbol{\rho}_l^2 = \begin{bmatrix} \|\mathbf{s}_1^T\|^2 - \frac{1}{N} \sum \|s_i\|^2 \\ \vdots \\ \|\mathbf{s}_N^T\|^2 - \frac{1}{N} \sum \|s_i\|^2 \end{bmatrix} + 2S\mathcal{O}_l \boldsymbol{\tau}_l. \quad (1.56)$$

Now, the first term of (1.56) is independent of time,  $t_l$ , and can be removed by taking an inter-pulse difference,

$$C(\boldsymbol{\rho}_l^2 - \boldsymbol{\rho}_k^2) = 2S(\mathcal{O}_l \boldsymbol{\tau}_l - \mathcal{O}_k \boldsymbol{\tau}_k). \quad (1.57)$$

The vector  $C(\boldsymbol{\rho}_l^2 - \boldsymbol{\rho}_k^2) \in range(S)$  for any pair  $(l, k)$ , and so we can define a mapping on  $R$  such that the columns of  $g(R)$  are in the range of  $S$ ,

$$g(R) = C(R \odot R)D. \quad (1.58)$$

$D \in \mathbb{R}^{N \times \binom{L}{2}}$  is a matrix encoding column-wise differences, e.g.

$$D = \begin{bmatrix} 1 & 0 & -1 & 1 & \dots & 0 \\ -1 & 1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 1 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -1 \\ 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix}. \quad (1.59)$$

The mapping given in (1.58) is sufficient for identifying the left singular vector matrix,  $V$ , from our procedure for estimating invariants, but it is not clear how to compute the magnitude of  $(\mathcal{O}_l \boldsymbol{\tau}_l - \mathcal{O}_k \boldsymbol{\tau}_k)$  so that we can solve for  $W$  in (1.39). To alleviate this difficulty, we define

$$\mathbf{m} = \begin{bmatrix} \|\mathbf{s}_1^T\|^2 - \frac{1}{N} \sum \|\mathbf{s}_i\|^2 \\ \vdots \\ \|\mathbf{s}_N^T\|^2 - \frac{1}{N} \sum \|\mathbf{s}_i\|^2 \end{bmatrix} \quad (1.60)$$

and rewrite (1.56) as

$$f(\boldsymbol{\rho}_l) := \frac{1}{2} (C \boldsymbol{\rho}_l^2 - \mathbf{m}) = S \mathcal{O}_l \boldsymbol{\tau}_l. \quad (1.61)$$

When we choose  $\mathbf{c}_l = \mathcal{O}_l \boldsymbol{\tau}_l$  and insert these expressions into (1.39), we have

$$\begin{aligned} \|\boldsymbol{\tau}_l\|^2 &= \frac{1}{4} (C \boldsymbol{\rho}_l^2 - \mathbf{m})^T V W V^T (C \boldsymbol{\rho}_l^2 - \mathbf{m}) \\ &= \frac{1}{4} (\mathbf{b}_l - \mathbf{u})^T W (\mathbf{b}_l - \mathbf{u}) \\ &= \frac{1}{4} (\mathbf{b}_l^T W \mathbf{b}_l - 2 \mathbf{u}^T W \mathbf{b}_l + \mathbf{u}^T W \mathbf{u}), \end{aligned} \quad (1.62)$$

where  $\mathbf{b}_l = V^T C \boldsymbol{\rho}_l^2$  and  $\mathbf{u} = V^T \mathbf{m}$ . We have no way of estimating  $\|\boldsymbol{\tau}_l\|^2$ , but we do know

$$\frac{1}{N} \mathbf{1}^T \boldsymbol{\rho}_l^2 = \|\boldsymbol{\tau}_l\|^2 + \frac{1}{N} \sum \|\mathbf{s}_i\|^2, \quad (1.63)$$

so that

$$\|\boldsymbol{\tau}_l\|^2 - \|\boldsymbol{\tau}_k\|^2 = \frac{1}{N} \mathbf{1}^T (\boldsymbol{\rho}_l^2 - \boldsymbol{\rho}_k^2). \quad (1.64)$$

Define  $\boldsymbol{\eta}_{lk} := \frac{1}{N} \mathbf{1}^T (\boldsymbol{\rho}_l^2 - \boldsymbol{\rho}_k^2)$ , and plug equation (1.62) into (1.64) to find

$$4\boldsymbol{\eta}_{lk} = \mathbf{b}_l^T W \mathbf{b}_l - \mathbf{b}_k^T W \mathbf{b}_k - 2(\mathbf{b}_l - \mathbf{b}_k)^T W \mathbf{u}. \quad (1.65)$$

As before, we will expand the symmetric matrix  $W$  as  $\mathbf{w} = [w_{11}, w_{12}, w_{13}, w_{22}, w_{23}, w_{33}]^T$ . In this case, however, we must also solve for the invariants in  $\mathbf{z} := W \mathbf{u}$ , and so we combine the equations for each of the  $\binom{L}{2}$  combinations of pulses from (1.65) into the system

$$A \begin{bmatrix} \mathbf{w} \\ \mathbf{z} \end{bmatrix} = 4 \begin{bmatrix} \boldsymbol{\eta}_{12} \\ \vdots \\ \boldsymbol{\eta}_{lk} \\ \vdots \\ \boldsymbol{\eta}_{(L-1)L} \end{bmatrix}. \quad (1.66)$$

Each row of the matrix  $A$  corresponds to one pair of pulses, so we define the rows to be

$$A^{lk} = \begin{bmatrix} a_{11}^{lk} & a_{12}^{lk} & a_{13}^{lk} & a_{22}^{lk} & a_{23}^{lk} & a_{33}^{lk} & a_1^{lk} & a_2^{lk} & a_3^{lk} \end{bmatrix}, \quad (1.67)$$

where

$$\begin{aligned} a_{11}^{lk} &= b_1^2(t_l) - b_1^2(t_k) & a_{12}^{lk} &= 2(b_1(t_l)b_2(t_l) - b_1(t_k)b_2(t_k)) & a_1^{lk} &= 2(b_1(t_k) - b_1(t_l)) \\ a_{22}^{lk} &= b_2^2(t_l) - b_2^2(t_k) & a_{13}^{lk} &= 2(b_1(t_l)b_3(t_l) - b_1(t_k)b_3(t_k)) & a_2^{lk} &= 2(b_2(t_k) - b_2(t_l)) \\ a_{33}^{lk} &= b_3^2(t_l) - b_3^2(t_k) & a_{23}^{lk} &= 2(b_2(t_l)b_3(t_l) - b_2(t_k)b_3(t_k)) & a_3^{lk} &= 2(b_3(t_k) - b_3(t_l)) \end{aligned}$$

and we have used the notation  $\mathbf{b}_l = [b_1(t_l), b_2(t_l), b_3(t_l)]^T$ . Once the invariants in  $\mathbf{w}$  are known from solving (1.66), we can solve for a shape representative as described in Section 1.3.2.

### 1.3.5 Bistatic Far-Field Case

Invariants for bistatic sensor systems are not as well understood; in fact, it is not known whether any geometric invariant equations exist for bistatic range measurements. For the far-field case, however, Arnold et al. [21] describe a technique for identifying the affine invariants.

In this scenario, we have one transmitter and one receiver. Without loss of generality, we can orient our coordinate system so that the transmitter and receiver both lie on the  $x$ -axis, at positions  $-\alpha$  and  $\alpha$ , and the target position is given by (1.28), as before. With this setup, the bistatic range to the  $n^{\text{th}}$  scatterer at time  $l$  is

$$\rho_{nl} = \|\mathbf{x}_{ln} + \alpha_l\| + \|\mathbf{x}_{ln} - \alpha_l\|. \quad (1.68)$$

To apply the far-field approximation, we first substitute our model for  $\mathbf{x}_{nl}$  from (1.27), and define  $\gamma_{tl} = \tau_l + \alpha_l$ ,  $\gamma_{rl} = \tau_l - \alpha_l$ . Note that, in the far-field, the distance between the scattering centers is much smaller than the distance from the scattering centers to the sensors, so that  $\|\mathcal{O}_l \mathbf{s}_n\| \ll \|\gamma_{tl}\|, \|\mathcal{O}_l \mathbf{s}_n\| \ll \|\gamma_{rl}\|$ . Our approximation follows,

$$\begin{aligned} \rho_{nl} &= \|\mathcal{O}_l \mathbf{s}_n + \tau_l + \alpha_l\| + \|\mathcal{O}_l \mathbf{s}_n + \tau_l - \alpha_l\| \\ &= \|\mathcal{O}_l \mathbf{s}_n + \gamma_{tl}\| + \|\mathcal{O}_l \mathbf{s}_n + \gamma_{rl}\| \\ &\approx \mathbf{s}_n^T \mathcal{O}_l^T (\hat{\gamma}_{tl} + \hat{\gamma}_{rl}) + \|\gamma_{tl}\| + \|\gamma_{rl}\|. \end{aligned} \quad (1.69)$$

As in the other cases, we can concatenate the range measurements together to form a vector of ranges at time  $l$ ,

$$\boldsymbol{\rho}_l = S \mathcal{O}^T (\hat{\gamma}_{tl} + \hat{\gamma}_{rl}) + (\|\gamma_{tl}\| + \|\gamma_{rl}\|) \mathbf{1}, \quad (1.70)$$

and apply the centering matrix,  $C$ ,

$$f(\boldsymbol{\rho}_l) := C \boldsymbol{\rho}_l = S \mathcal{O}^T (\hat{\gamma}_{tl} + \hat{\gamma}_{rl}). \quad (1.71)$$

$f(\boldsymbol{\rho}_l)$  is in  $range(S)$ , so that we can approximate the affine invariants of  $S$ . If we have estimates for  $\mathbf{c}_l = \|\hat{\boldsymbol{\gamma}}_{tl} + \hat{\boldsymbol{\gamma}}_{rl}\|$ , we can compute Euclidean invariants of the target just as we did for the monostatic, far-field case. Without these estimates, however, this method will not suffice to find the Euclidean invariants of the target.

## 1.4 Related Techniques

### 1.4.1 Interferometric Developments

Following the approaches discussed in section 1.2, some radar engineers recognized that 2D imaging is insufficient, and instead exploited the phase of the returning wave to identify the 3D shape and motion of the target. In 2009, Mayhan [22] described a technique that allowed him to develop 3D images of a target by comparing the phase differences between two nearly-identical 2D images. The difference in phase between the two "snapshots" gives enough information to estimate the height of the scattering centers out of the range-doppler plane. Then in 2014, Martorella et al. [23] used the multi-channel CLEAN technique to extract phase measurements relative to two orthogonal baselines to jointly estimate the target rotation and the height of the scattering centers out of the imaging plane.

### 1.4.2 Image Factorization

With the improvements made to the 3DMAGI approach in 2009 [20], Ferrara et al. noticed an interesting similarity to an approach from electro-optical imaging. Given a sequence of images from a traditional camera, Tomasi and Kanade [24] describe a factorization method for recovering the scene geometry and the camera motion. Aside from the apparent difference in sensor type, the approach utilizes more constraints than are possible in the radar case. Each image produced by a camera is a 2D orthographic projection of the 3D scene, and naturally contains more information than the 1D range measurements produced by a radar; this allows the optical factorization algorithm to operate with few images and fewer tracked features than in the radar case. Otherwise, the

two approaches similarly compute the shape and motion of a target by factorizing a data matrix with respect to constraints on the distribution of scaling factors.

Within the field of computer vision, the problem of reconstructing the 3D shape of a stationary scene from a set of projective measurements is called the Structure from Motion (SfM) problem [25]. The majority of the SfM literature considers only the 2D projective measurements produced by optical cameras, but our 1D range measurements are also projective measurements with appropriate model assumptions (see section 1.3.3). In this way, our problem is a close analogue to the SfM problem.

### 1.4.3 Low-Rank Subspace Decomposition

In real applications, the collected range data will be noisy and often incomplete. Naturally, we cannot view a 3D target from every angle with a single sensor. Scattering centers on the side of the target opposite from the sensor will be invisible to the radar system, so that at any time we only see a fraction of the scattering centers on an object. This self-shadowing, along with other undesirable effects, mean that the range data matrix,  $R$ , will often have empty or wildly inaccurate entries. Additionally, the sensor cannot distinguish between scattering centers that belong to target and those that do not, so that some of the tracks in  $R$  are range measurements to *other* targets in the scene. Arnold et al. [21] address this problem with the Low-rank Affinity Matrix Estimation (LAME) algorithm, which separates the scatterers by assuming that the motion of each target is necessarily low-rank. Following the example of other Low-Rank Representation (LRR) approaches, LAME factors the data into a clean dictionary matrix,  $A$ , and a low-rank coefficient matrix,  $C$ , so that  $R^T \approx A = AC$ . The entries of  $C$  are then used to group the tracks into probable targets, and the matrix  $A$  serves as a cleaner, complete version of the track matrix to be used in shape and motion estimation.

# Chapter 2

## A Polynomial Systems Approach

In section 1.3, we outlined the current best practice in joint shape and motion estimation. The approaches there identified transformations on the data that simplified the shape and motion reconstruction process to a pair of matrix factorizations. If we could find a similar transformation on the bistatic, near-field (BNF) data to linearize the shape and motion estimation problem, we would be able to replicate the outlined approaches. Finding such a transformation has proven non-trivial. Instead of identifying such a transformation, we can describe a polynomial system relating the bistatic range measurements, the target shape, and the target motion, and attempt to solve the non-linear system of equations directly.

### 2.1 Expression as a Polynomial System

Suppose we have a rigid target, defined by a point cloud with  $N$  points. As in section 1.3, we encode the shape of the point cloud as a centered configuration matrix,

$$S = \begin{bmatrix} \mathbf{s}_1^T \\ \vdots \\ \mathbf{s}_N^T \end{bmatrix}. \quad (2.1)$$

In this format,  $S \in \mathbb{R}^{N \times 3}$ , and each row of  $S$  is the position vector for a single point in Euclidean space. Since we have assumed that the point cloud is centered, we have  $\mathbf{1}^T S = \mathbf{0}^T$ .

At each time  $t_l$ , the target's position is a rotated and translated version of  $S$ ,

$$X_l = S\mathcal{O}_l + \mathbf{1}\boldsymbol{\tau}_l^T. \quad (2.2)$$

$\mathcal{O}_l \in SO(3)$  is a time-dependent rotation, and  $\boldsymbol{\tau}_l \in \mathbb{R}^3$  represents a time-dependent translation. Suppose that there are two sensors, fixed at the positions  $\boldsymbol{\alpha}_l = [\alpha_l, 0, 0]$  and  $-\boldsymbol{\alpha}_l$ . For bistatic

sensor arrangements, we only measure the sum of ranges

$$\rho_{nl} = \underbrace{\|\mathbf{x}_{nl} + \boldsymbol{\alpha}_l\|}_{\text{transmitter to scatterer}} + \underbrace{\|\mathbf{x}_{nl} - \boldsymbol{\alpha}_l\|}_{\text{scatterer to receiver}}, \quad (2.3)$$

and possibly the distance between the sensors,  $2\alpha_l$ .

### 2.1.1 Ellipse Constraints

For each point  $n$  and time  $l$ , suppose we know the distance between the sensors,  $2\alpha_l$ , and the bistatic range to the scatterers,  $\rho_{nl}$ . This knowledge uniquely defines an ellipsoid on which the scatterers must lie. The foci of the ellipsoid are  $\boldsymbol{\alpha}_l$  and  $-\boldsymbol{\alpha}_l$ ; the major axis ( $x$ -axis) of the ellipse has length  $2a$ , and the remaining axes have length  $2b$ , where

$$\begin{aligned} a_{nl}^2 &= \frac{1}{4}\rho_{nl}^2, \\ b_{nl}^2 &= a_{nl}^2 - \alpha_l^2. \end{aligned} \quad (2.4)$$

To see this, take an arbitrary point  $\mathbf{x} = (x_1, x_2, x_3)$ . The sum of distances from  $\boldsymbol{\alpha}_l$  to  $\mathbf{x}$  and from  $\mathbf{x}$  to  $-\boldsymbol{\alpha}_l$  is the bistatic range to  $x$ ,

$$\rho = \sqrt{(x_1 - \alpha)^2 + x_2^2 + x_3^2} + \sqrt{(x_1 + \alpha)^2 + x_2^2 + x_3^2} \quad (2.5)$$

Substitute the relation  $\rho = 2a$ , and we can manipulate the equation to find

$$1 = \frac{x_1^2}{a^2} + \frac{x_2^2}{a^2 - \alpha^2} + \frac{x_3^2}{a^2 - \alpha^2}. \quad (2.6)$$

This holds for each scatterer at each time instant, so we have the system of equations

$$1 = \mathbf{x}_{nl}^T \begin{bmatrix} \frac{1}{a_{nl}^2} & 0 & 0 \\ 0 & \frac{1}{b_{nl}^2} & 0 \\ 0 & 0 & \frac{1}{b_{nl}^2} \end{bmatrix} \mathbf{x}_{nl} \quad \forall n = 1 \dots N; l = 1 \dots L. \quad (2.7)$$

Now, substitute  $x_{nl}^T = \mathbf{s}_n^T \mathcal{O}_l + \boldsymbol{\tau}_l^T$ , and let

$$\Sigma_{nl} = \begin{bmatrix} a_{nl} & 0 & 0 \\ 0 & b_{nl} & 0 \\ 0 & 0 & b_{nl} \end{bmatrix}, \quad (2.8)$$

So that we have the system for all  $n$  and  $l$ ,

$$1 = (\mathcal{O}_l^T \mathbf{s}_n + \boldsymbol{\tau}_l)^T \Sigma_{nl}^{-2} (\mathcal{O}_l^T \mathbf{s}_n + \boldsymbol{\tau}_l), \quad (2.9)$$

with the additional constraints,

$$\begin{aligned} \mathcal{O}_l^T \mathcal{O}_l &= I \quad \forall l = 1 \dots L, \\ \mathbf{1}^T S &= \mathbf{0}^T. \end{aligned} \quad (2.10)$$

### 2.1.2 Alternative System

Starting with (2.9), we can define

$$\mathbf{c}_{nl} \equiv \Sigma_{nl}^{-1} (\mathcal{O}_l^T \mathbf{s}_n + \boldsymbol{\tau}_l), \quad (2.11)$$

so that  $1 = \mathbf{c}_{nl}^T \mathbf{c}_{nl}$ . Then

$$\begin{aligned} \mathcal{O}_l^T \mathbf{s}_n + \boldsymbol{\tau}_l &= \Sigma_{nl} \mathbf{c}_{nl} \\ \implies \mathbf{s}_n &= \mathcal{O}_l \Sigma_{nl} \mathbf{c}_{nl} - \mathcal{O}_l \boldsymbol{\tau}_l \\ \implies \mathbf{s}_m - \mathbf{s}_n &= \mathcal{O}_l (\Sigma_{ml} \mathbf{c}_{ml} - \Sigma_{nl} \mathbf{c}_{nl}) \\ \implies \underbrace{(\mathbf{s}_m - \mathbf{s}_n)^T (\mathbf{s}_m - \mathbf{s}_n)}_{\text{independent of } l} &= \mathbf{c}_{ml}^T \Sigma_{ml}^2 \mathbf{c}_{ml} - 2 \mathbf{c}_{nl}^T \Sigma_{nl} \Sigma_{ml} \mathbf{c}_{ml} + \mathbf{c}_{nl}^T \Sigma_{nl}^2 \mathbf{c}_{nl}. \end{aligned} \quad (2.12)$$

The vector  $(\mathbf{s}_m - \mathbf{s}_n)^T (\mathbf{s}_m - \mathbf{s}_n)$  is the squared distance between scatterer  $m$  and scatterer  $n$ , so this quantity is fixed (but still unknown). We introduce the notation,

$$d_{mn}^2 = (\mathbf{s}_m - \mathbf{s}_n)^T (\mathbf{s}_m - \mathbf{s}_n), \quad (2.13)$$

and note that the angles formed by any three scatterers are also fixed, so that we could generate  $N \binom{N-1}{2}$  more equations for each unknown of the form  $(\mathbf{s}_m - \mathbf{s}_n)^T (\mathbf{s}_p - \mathbf{s}_n)$ , and then add further constraints between the unknowns based on geometric relations between the angles defined by the scatterers and the squared distances between scatterers.

If we write  $c_{nl} = [x_{nl}, y_{nl}, z_{nl}]^T$ , then we can expand and simplify (2.12) as

$$(a_{ml}x_{ml} - a_{nl}x_{nl})^2 + (b_{ml}y_{ml} - b_{nl}y_{nl})^2 + (b_{ml}z_{ml} - b_{nl}z_{nl})^2 - d_{mn}^2 = 0 \quad \forall m \neq n, \quad (2.14)$$

Along with the unit-length constraints,  $\|c_{nl}\|^2 = 1$ , these equations describe a system of  $L \binom{N}{2} + NL$  equations and  $\binom{N}{2} + 3NL$  variables for the 3D case. If we instead consider the 2D case, we can drop the third coordinate from all of our computations so that we only have  $\binom{N}{2} + 2NL$  variables. In either case, we would like to avoid positive-dimensional solution components, and so we do not want our system to be underdetermined. To have more equations than variables for the 2D and 3D case, we require,

$$L \binom{N}{2} + NL \geq \binom{N}{2} + 2NL \implies (N-3)(L-1) \geq 2 \quad (2.15)$$

$$L \binom{N}{2} + NL \geq \binom{N}{2} + 3NL \implies (N-5)(L-1) \geq 4, \quad (2.16)$$

respectively. It follows that we must track  $N \geq 4$  scatterers in the 2D case, and  $N \geq 6$  scatterers in the 3D case.

### 2.1.3 Attempted Numerical Solution with Bertini

Now, we can attempt to directly solve the system

$$\begin{aligned} 0 &= (a_{ml}x_{ml} - a_{nl}x_{nl})^2 + (b_{ml}y_{ml} - b_{nl}y_{nl})^2 + (b_{ml}z_{ml} - b_{nl}z_{nl})^2 - d_{mn}^2 & \forall m \neq n \\ 1 &= x_{nl}^2 + y_{nl}^2 + z_{nl}^2 & \forall n, l. \end{aligned} \quad (2.17)$$

Instead of solving the system exactly, we will use the Numerical Algebraic Geometry (NAG) software program, Bertini [26].

Bertini uses a process called *homotopy continuation* to identify the solutions to our input system. Suppose our system is defined by

$$f(\mathbf{z}) := \begin{bmatrix} f_1(z_1, \dots, z_N) \\ \vdots \\ f_N(z_1, \dots, z_N) \end{bmatrix} = 0, \quad (2.18)$$

and suppose we know the solutions to another system,  $g(\mathbf{z}) = 0$ . Then we can define a homotopy between the two systems,

$$h(\mathbf{z}, s) = sg(\mathbf{z}) + (1 - s)f(\mathbf{z}). \quad (2.19)$$

For  $s = 1$ ,  $h(\mathbf{z}, 1) = g(\mathbf{z})$ , and for  $s = 0$ ,  $h(\mathbf{z}, 0) = f(\mathbf{z})$ . The key to this process is that the solutions of  $h(\mathbf{z}, s)$  vary continuously in  $s$ , so that we have paths connecting the solutions of  $f$  to the solutions of  $g$ . Additionally, the paths will not intersect with probability one [27]. The combination of these facts allows Bertini to track the solutions of  $g$  (the start points) to the solutions of  $f$  (the endpoints) using a combination of predictor-corrector methods. Put over-simply, we use Euler's method to move along the path from  $s = 1$  to  $s = 0$ , alternated with Newton's method steps included to correct the predictions back to the path. Some problems arise with solutions coinciding or even diverging to infinity, but Bertini handles these with specialized approaches, called endgames, that are managed adaptively.

Since we must have a known set of start points, Bertini's first task is to estimate the number of paths necessary to capture all of the solutions of the input system, and generate a system with at least that many solutions. The simplest method, total degree homotopy, simply uses the roots of unity as starting points (keep in mind that Bertini solves polynomials over  $\mathbb{C}$ ). If the input system  $f$  has degree  $d_i$  for  $f_i$ , then the total degree of  $f$  is  $D = d_1 \cdots d_N$ , and so Bertini uses the solutions to the system  $g_i(\mathbf{z}) = z_i^{d_i} - 1$  as the start points. As an example, suppose we run a 2D example with  $N = 4, L = 3$ . We have 30 quadratic equations in 30 variables, so that

running a total degree homotopy would involve tracking  $2^{30} = 1,073,741,824$  paths. This makes the total degree homotopy infeasible given our resources. Other homotopy methods are available in Bertini, including the multihomogeneous homotopy and equation-by-equation solution methods like regeneration. The multihomogeneous homotopy allows us to decrease the necessary size of our start system, saving on computation time by taking advantage of sparsity in our polynomial system. Regeneration allows us to solve the full system of equations in intermediate steps, essentially allowing us to introduce one polynomial at a time. In some cases, this can automatically reveal structure in the system of equations, and greatly increase the computation speed [27].

To date, we are unable to compute the solution to (2.17). Even with our simplest possible example, with  $N = 4, L = 3$  in 2D, the computational resources necessary are infeasible. There is still some hope, however, that we may be able to reformulate the polynomial system to run in a reasonable amount of time. With one proof-of-concept run, we should be able to extend the results to any new system with the same  $N$  and  $L$ . Sometimes normally occurring systems have fewer solutions than are predicted by the start system of our homotopy. If we can solve one example of this system, Bertini has another tool, called parameter homotopy, which will allow us to deform the solutions for the old set of parameters into the solution set for the new set of parameters. Suppose we have the set of solutions  $Z_i$ , to the system

$$f(\mathbf{z}) := \begin{bmatrix} f_1(\mathbf{z}; \mathbf{a}_i) \\ \vdots \\ f_N(\mathbf{z}; \mathbf{a}_i) \end{bmatrix} = 0, \quad (2.20)$$

with the vector of parameters  $\mathbf{a}_i$ . Then we can track paths from the solution  $Z_i$  to the solutions to the same system with parameters  $\mathbf{a}_j$ . The parameter homotopy works in much the same way as the conventional homotopy, with some additional guarantees. If we originally solved our system for general parameter values  $\mathbf{a}_i$ , then the number of solutions in any  $Z_j$  must necessarily be less than or equal to the number of solutions in  $Z_i$ . This means we need only track one path for each solution in  $Z_i$ , which greatly reduces computation time.

# Chapter 3

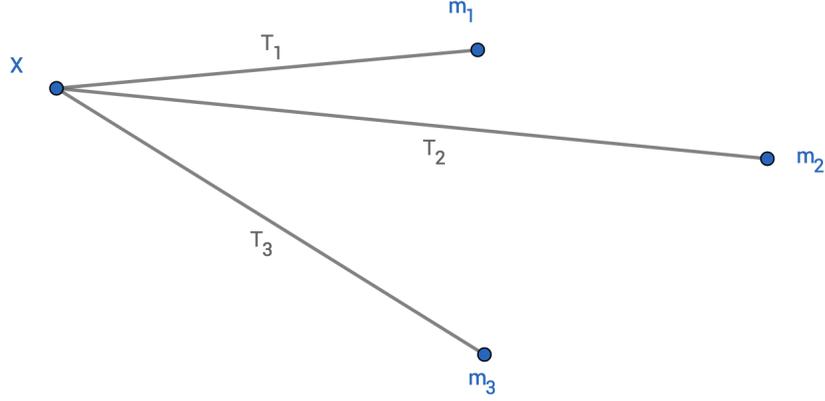
## A Variety-Fitting Approach

At the conception of 3DMAGI, Stuff et al. [1] realized that the far-field approximated range data from a sensor falls on a submanifold of the space of possible range observations, and in subsequent papers [13–15] they would even go on to write invariant equations that the range data must satisfy. Though it was not framed in the language of algebraic geometry at the time, the monostatic, far-field invariant equations define an algebraic variety in the range space. Unwittingly, the monostatic, near-field invariant equation for 3 scatterers in 2D would appear in a paper [28] motivated by the Time Of Arrival (TOA) problem.

In their 2017 paper, Campagnoni et al. consider the algebraic structure of monostatic range measurements from a transmitter to three fixed receivers. Given three receivers with known positions  $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3 \in \mathbb{R}^2$ , and a transmitter, with unknown position,  $\mathbf{x} \in \mathbb{R}^2$ , they call the displacement vector from receiver to transmitter  $\mathbf{d}_i(\mathbf{x}) := \mathbf{x} - \mathbf{m}_i$ , so that the Euclidean distance from  $\mathbf{x}$  to  $\mathbf{m}_i$  is  $d_i(\mathbf{x}) := \|\mathbf{d}_i(\mathbf{x})\|$ . Then  $\mathbf{d}_{ij} := \mathbf{m}_i - \mathbf{m}_j$  is the displacement vector from  $\mathbf{m}_i$  to  $\mathbf{m}_j$ . The system

$$\begin{cases} d_1(\mathbf{x}) = \mathcal{T}_1 \\ d_2(\mathbf{x}) = \mathcal{T}_2 \\ d_3(\mathbf{x}) = \mathcal{T}_3 \end{cases} \quad (3.1)$$

defines a map,  $\mathcal{T}_3 : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ , which takes the transmitter position as input and outputs the range measurements to each of the receivers. It follows that the image of  $\mathcal{T}_3$  contains every possible combination of the range measurements  $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$ . To reflect this, we will call (3.1) the *range map* of the scatterer configuration. Campagnoni et al. show that the image of  $\mathcal{T}_3$  is contained in a real algebraic surface. Further, they show that this variety is actually a known degree-four surface, called a Kummer surface, with parameters directly related to the configuration of receivers. They expressed the surface as

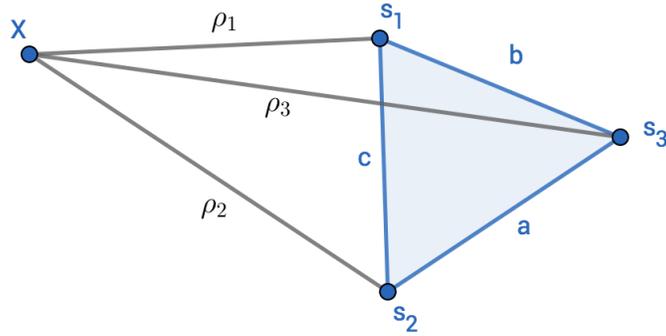


**Figure 3.1:** A generic configuration of receivers  $\mathbf{m}_1$ ,  $\mathbf{m}_2$ ,  $\mathbf{m}_3$  and transmitter  $\mathbf{x}$ .

$$\begin{aligned}
0 = & d_{32}^2 \mathcal{T}_1^4 + d_{31}^2 \mathcal{T}_2^4 + d_{21}^2 \mathcal{T}_3^4 \\
& - 2\mathbf{d}_{32}^T \mathbf{d}_{31} \mathcal{T}_1^2 \mathcal{T}_2^2 - 2\mathbf{d}_{32}^T \mathbf{d}_{21} \mathcal{T}_1^2 \mathcal{T}_3^2 - 2\mathbf{d}_{31}^T \mathbf{d}_{21} \mathcal{T}_2^2 \mathcal{T}_3^2 \\
& - 2\mathbf{d}_{32}^T \mathbf{d}_{31} d_{21}^2 \mathcal{T}_3^2 - 2\mathbf{d}_{32}^T \mathbf{d}_{21} d_{31}^2 \mathcal{T}_2^2 - 2\mathbf{d}_{31}^T \mathbf{d}_{21} d_{32}^2 \mathcal{T}_1^2 \\
& + d_{21}^2 d_{31}^2 d_{32}^2,
\end{aligned} \tag{3.2}$$

from which we can see that the possible set of range measurements is determined by the squared distances between the receivers and the inner products of the displacement vectors. From this observation, it is clear that the image of  $\mathcal{T}_3$  is determined by the configuration of receivers. This connection between the image of  $\mathcal{T}_3$  and the configuration of receivers is a multiple of the invariant equation given in (1.62), section 1.3.4.

In this chapter, we will reframe the earlier method as a technique for estimating geometric invariants of a scatterer configuration by finding the variety of best fit that describes a sample of range data. We begin with a simpler presentation of the derivation for the monostatic, near-field variety given in (3.2), and then describe how knowledge of this variety allows us to recover geometric invariants from sampled range data. We then describe techniques for discovering geometric invariant equations for other scatterer configurations, including the bistatic, near-field case.



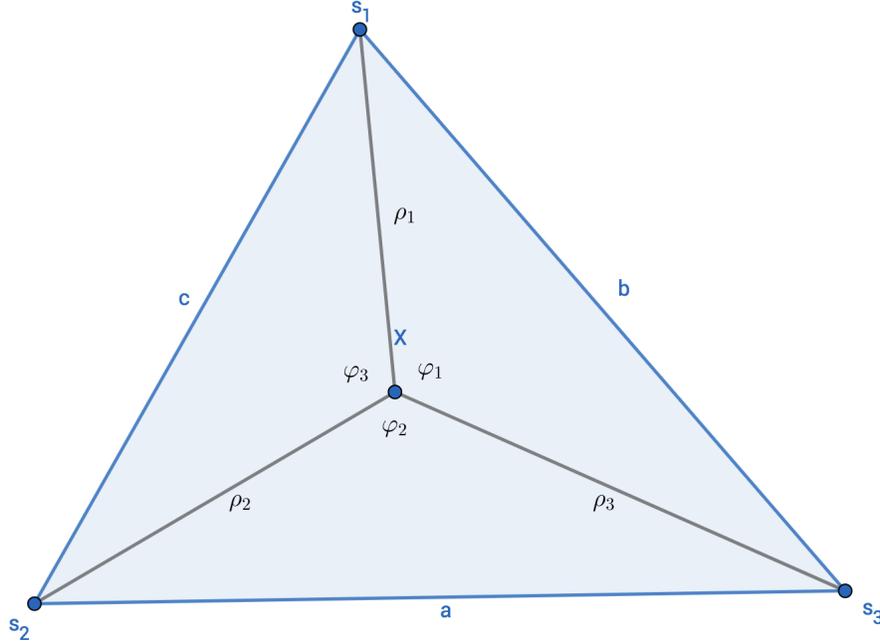
**Figure 3.2:** A generic configuration of scattering centers  $s_1, s_2, s_3$  and radar platform,  $x$ .

### 3.1 Monostatic, Near-Field Case

In their papers, Compagnoni et al. consider the problem of locating a transmitter, given the positions of multiple receivers and the range from each receiver to the transmitter. The problem we have in mind is more complicated, but with some re-labelling we can see how the two are connected. Instead of considering receivers  $m_1, m_2, m_3$ , suppose we have scattering centers at positions  $s_1, s_2, s_3 \in \mathbb{R}^2$  and a radar platform at  $x \in \mathbb{R}^2$ . In our problem, we can still measure the range from  $x$  to each of the scatterers, but the positions of each of the scatterers, as well as the position of the radar platform, are unknown. We first must estimate the geometric invariant parameters in (3.2), after which we may either use the approach explained by Compagnoni [28], or the original 3DMAGI approach to estimate the transmitter position.

#### 3.1.1 Derivation of 2D Monostatic Near-Field Variety

For this derivation, we assume that the three scattering centers  $s_1, s_2, s_3$  are not collinear. For convenience, we place the receiver on the interior of the triangle formed by the scatterers, and label the angles  $\varphi_1, \varphi_2$ , and  $\varphi_3$ , and side lengths  $a, b, c$ , as shown in Figure 3.3. The following approach is only one of many possible constructions, as this figure has been studied in many different contexts. In his paper studying the set of points at rational distance from the vertices of the triangle, Berry



**Figure 3.3:** A generic configuration of scattering centers  $s_1, s_2, s_3$ , but with the radar platform contained within the triangle formed by the three receivers.

notes that the relation in (3.5) was discovered sometime in the last century [29]. We will follow the geometric approach presented by Bottema [30], for its cleanliness and accessibility.

First, note that  $\varphi_1 + \varphi_2 + \varphi_3 = 2\pi$ , so that  $\cos(\varphi_1 + \varphi_2 + \varphi_3) = 1$ . Using angle addition formulas, we have

$$\cos(\varphi_1 + \varphi_2 + \varphi_3) = \cos^2(\varphi_1) + \cos^2(\varphi_2) + \cos^2(\varphi_3) - 2 \cos(\varphi_1) \cos(\varphi_2) \cos(\varphi_3), \quad (3.3)$$

so that

$$0 = 1 + 2 \cos(\varphi_1) \cos(\varphi_2) \cos(\varphi_3) - \cos^2(\varphi_1) - \cos^2(\varphi_2) - \cos^2(\varphi_3). \quad (3.4)$$

By the law of cosines, we have

$$\cos(\varphi_1) = \frac{-b^2 + \rho_1^2 + \rho_3^2}{2\rho_1\rho_3}, \quad \cos(\varphi_2) = \frac{-a^2 + \rho_2^2 + \rho_3^2}{2\rho_2\rho_3}, \quad \cos(\varphi_3) = \frac{-c^2 + \rho_1^2 + \rho_2^2}{2\rho_1\rho_2}.$$

Inserting these relations into (3.4) and reducing the resulting right hand side leads to

$$\begin{aligned}
0 = & a^2 \rho_1^4 + b^2 \rho_2^4 + c^2 \rho_3^4 + a^2 b^2 c^2 \\
& + (-a^2 - b^2 + c^2) \rho_1^2 \rho_2^2 + (-a^2 + b^2 - c^2) \rho_1^2 \rho_3^2 + (a^2 - b^2 - c^2) \rho_2^2 \rho_3^2 \\
& + a^2 (a^2 - b^2 - c^2) \rho_1^2 + b^2 (-a^2 + b^2 - c^2) \rho_2^2 + c^2 (-a^2 - b^2 + c^2) \rho_3^2.
\end{aligned} \tag{3.5}$$

Note that since  $\cos(2\pi - \varphi) = \cos(\varphi)$ , equation (3.5) is true even when the radar platform is not within the triangle defined by the scatterer positions.

Assuming  $a, b, c > 0$ , (3.5) describes an algebraic variety ( see section 3.2 ) which contains the image of the range map from (3.1). We can show that this variety is the “smallest” such variety, in the sense that any other variety containing the image of the range map will also contain the variety defined by (3.5). Before we go into detail on this, we will describe how one might use (3.5) to estimate the geometric invariants  $a, b$ , and  $c$ .

### 3.1.2 Estimation of Geometric Invariants in 2D

In general, we will want to estimate geometric invariants for targets with  $N$  scattering centers, where it’s very likely that  $N > 3$ . In this section, we will develop the procedure for estimating geometric invariants for triangular targets ( $N = 3$ ), and then argue that this approach is fundamentally the same as that detailed in section 1.3.4.

Suppose that we have range measurements taken from  $L$  distinct radar platform positions,  $\{\mathbf{x}_l\}_{l=1}^L$ . This amounts to sampling  $L$  points from the image of the map  $\mathcal{T}_3$ , where we denote the  $l^{th}$  point as  $\boldsymbol{\rho}_l = [\rho_1(\mathbf{x}_l) \ \rho_2(\mathbf{x}_l) \ \rho_3(\mathbf{x}_l)]^T$ .

Before proceeding, we will rescale the parameters in (3.5) for stability by dividing through by  $a^2 b^2 c^2$ . The resulting equation is

$$\begin{aligned}
-1 = & b^{-2} c^{-2} \rho_1^4 + a^{-2} c^{-2} \rho_2^4 + a^{-2} b^{-2} \rho_3^4 \\
& + (abc)^{-2} (-a^2 - b^2 + c^2) \rho_1^2 \rho_2^2 + (abc)^{-2} (-a^2 + b^2 - c^2) \rho_1^2 \rho_3^2 \\
& + (abc)^{-2} (a^2 - b^2 - c^2) \rho_2^2 \rho_3^2 + (bc)^{-2} (a^2 - b^2 - c^2) \rho_1^2 \\
& + (ac)^{-2} (-a^2 + b^2 - c^2) \rho_2^2 + (ab)^{-2} (-a^2 - b^2 + c^2) \rho_3^2.
\end{aligned} \tag{3.6}$$

From the  $l^{th}$  set of range measurements, form the vector

$$\mathbf{r}_l^T = \left[ \rho_1^4 \quad \rho_2^4 \quad \rho_3^4 \quad \rho_1^2 \rho_2^2 \quad \rho_1^2 \rho_3^2 \quad \rho_2^2 \rho_3^2 \quad \rho_1^2 \quad \rho_2^2 \quad \rho_3^2 \right], \quad (3.7)$$

where we have suppressed the range measurements' dependence on  $\mathbf{x}_l$ . By the relation (3.6), we know that  $\mathbf{r}_l$  satisfies

$$\mathbf{r}_l^T \mathbf{q} = -1, \quad (3.8)$$

where

$$\mathbf{q} = \begin{bmatrix} b^{-2}c^{-2} \\ a^{-2}c^{-2} \\ a^{-2}b^{-2} \\ (abc)^{-2}(-a^2 - b^2 + c^2) \\ (abc)^{-2}(-a^2 + b^2 - c^2) \\ (abc)^{-2}(a^2 - b^2 - c^2) \\ (bc)^{-2}(a^2 - b^2 - c^2) \\ (ac)^{-2}(-a^2 + b^2 - c^2) \\ (ab)^{-2}(-a^2 - b^2 + c^2) \end{bmatrix}, \quad (3.9)$$

is the vector of coefficients which correspond to unknown, fixed lengths in the scatterer configuration. Concatenate the vectors  $\mathbf{r}_l^T$  to form the matrix

$$R = \begin{bmatrix} \mathbf{r}_1^T \\ \vdots \\ \mathbf{r}_L^T \end{bmatrix}, \quad (3.10)$$

so that the solution to

$$R\mathbf{q} = -\mathbf{1} \quad (3.11)$$

contains geometric information about the scatterer configuration. From the solution  $\mathbf{q}$ , we can extract the squared distances between the sides of the receivers according to the relations

$$a^4 = \frac{q_1}{q_2 q_3}, \quad b^4 = \frac{q_2}{q_1 q_3}, \quad c^4 = \frac{q_3}{q_1 q_2}. \quad (3.12)$$

In practice,  $L \gg 8$ , so that the system (3.11) is overdetermined. Unfortunately, the range measurements will be noisy and this system usually will not have a solution. Instead, we will find the coefficients  $\mathbf{q}$  that best fit the system of equations in the least-squares sense. More formally, we will take  $\mathbf{q}^*$  to be our vector of invariants, where

$$\mathbf{q}^* = \arg \min_{\mathbf{q}} \|\mathbf{1} - R\mathbf{q}\|^2. \quad (3.13)$$

The unique minimizer to this problem is well-known,

$$\mathbf{q}^* = -(R^T R)^{-1} R^T \mathbf{1}. \quad (3.14)$$

In this way, our approach is analogous to the classic linear least squares problem, except that we are estimating the parameters of a variety rather than the coefficients of a function, as suggested in Figure 3.4. Thus, we name this approach the variety-fitting technique.

To improve the solution quality, we can augment the matrix  $R$  with some extra constraints describing the relationships between physical parameters. The parameters must satisfy the relationships

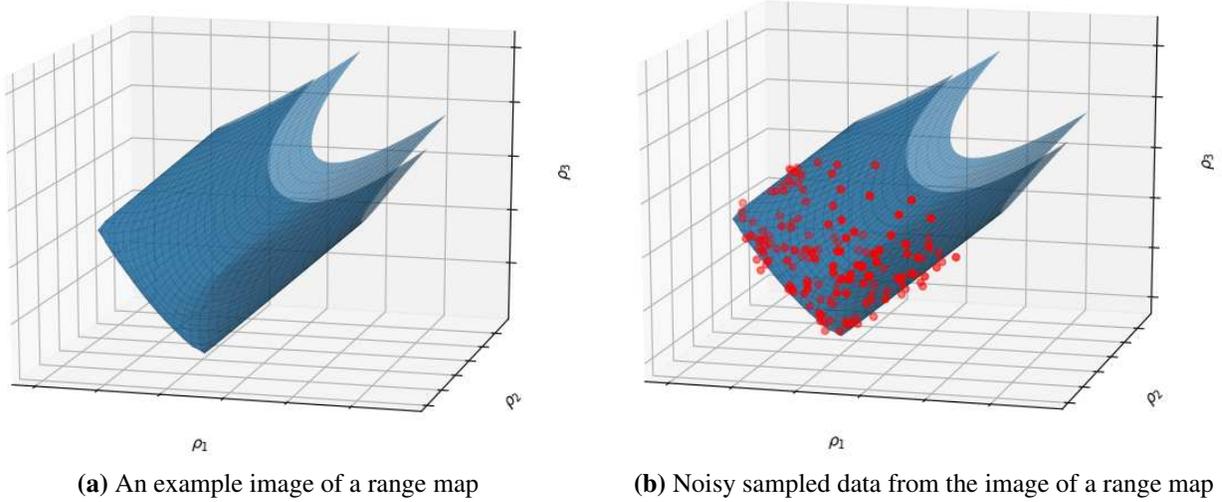
$$q_1 - q_2 - q_3 - q_6 = 0 \quad (3.15)$$

$$-q_1 + q_2 - q_3 - q_5 = 0 \quad (3.16)$$

$$-q_1 - q_2 + q_3 - q_4 = 0 \quad (3.17)$$

Including these constraints in  $R$  amounts to enforcing the law of cosines on each angle in the triangle defined by the scattering centers.

To see how this technique is equivalent to the approach from section 1.3.4, recall the original form of (3.5). We can write this in matrix form as



**Figure 3.4:** (a) The set of points in the image of the range map associated with  $N = 3$ ,  $\mathbf{s}_1 = [1, 0]$ ,  $\mathbf{s}_2 = [0, 1]$ ,  $\mathbf{s}_3 = [0, 0]$ . In (b), we have the image of the same range map overlaid with noisy range data sampled from the target. By tuning the parameters  $a, b, c$  to fit (3.5) to the noisy data, we can estimate the target shape.

$$0 = (\boldsymbol{\rho}^2)^T A \boldsymbol{\rho}^2 + \frac{1}{2} \mathbf{b}^T \boldsymbol{\rho}^2 + a^2 b^2 c^2, \quad (3.18)$$

where we define

$$A = \begin{bmatrix} a^2 & -\frac{1}{2}(c^2 - a^2 - b^2) & -\frac{1}{2}(b^2 - a^2 - c^2) \\ -\frac{1}{2}(c^2 - a^2 - b^2) & b^2 & -\frac{1}{2}(a^2 - b^2 - c^2) \\ -\frac{1}{2}(b^2 - a^2 - c^2) & -\frac{1}{2}(a^2 - b^2 - c^2) & c^2 \end{bmatrix}, \quad (3.19)$$

$$\mathbf{b} = \begin{bmatrix} a^2(a^2 - b^2 - c^2) \\ b^2(b^2 - a^2 - c^2) \\ c^2(c^2 - a^2 - b^2) \end{bmatrix}.$$

Define the differencing matrix  $D$  as

$$D = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}, \quad (3.20)$$

and recall that  $S$  denotes the matrix of scatterer positions. We can then write

$$A = DSS^T D^T.$$

From here, it is easy to show with a computer algebra system that

$$A = DSS^T D^T = 3[\det(S^T S)]S(S^T S)^{-2}S^T = 3[\det(S^T S)]\Omega, \quad (3.21)$$

and that the other terms in (3.18) are multiples of terms from (1.62), our original geometric invariant equation for the monostatic, near-field case, section 1.3.4. Further, the relations in (3.15) are equivalent to previous conditions,

$$\frac{1}{3\det(S^T S)}A\mathbf{1} = \Omega\mathbf{1} = \mathbf{0}. \quad (3.22)$$

Compared to the two-step process for estimating geometric invariants in section 1.3.4, the potentially large number of unknowns in our variety-fitting approach makes it less stable than the original technique. In this case, however, we see that we can recover the original invariant equation with alternative means.

Previous attempts at defining geometric invariant equations for the bistatic, near-field case have involved searching over equations defined in terms of matrices, either by hand manipulation or by least-squares fitting to synthetic data over various non-linear terms. In the following sections, we will attempt to search for a polynomial geometric invariant equation using the tools of elimination theory. The framework developed there will give us a means to identify polynomial invariant equations for the BNF case, even if the resulting equation may not be convenient to express with matrices.

## 3.2 Invariant Equation Discovery

To use the approach in chapter 3.1 for other radar scenarios, we must discover new geometric invariant equations for the bistatic cases. In this chapter, we will investigate the existence of geometric invariant equations from an algebro-geometric point of view, and attempt to adapt the techniques of elimination theory to create an effective means for generating new geometric invariant equations. To introduce this approach, we first show that we can recover the invariant equation described by Compagnoni et al. [28] for the monostatic, near-field case.

### 3.2.1 Recovering the Monostatic Near-Field Invariant

For simplicity, first suppose we have a target with  $N = 3$  scattering centers in 2D, so that the configuration matrix of scatterers is  $S \in \mathbb{R}^{3 \times 2}$ . Here, we attribute all motion to our receiver and assume that the scattering centers are fixed. Let  $\mathbf{x}(t_i) = (x, y)$  be the time-dependent position of the radar platform. Without loss of generality, we assume that the centroid of the 3 scatterers is the origin of our coordinate system. Let

$$S = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ -(x_1 + x_2) & -(y_1 + y_2) \end{bmatrix}. \quad (3.23)$$

This model for  $S$  contains an unnecessary degree of freedom, in that we have fixed the center of our coordinate system with respect to  $S$ , but we have not picked a standard orientation. We could do so by allowing  $s_1$  to be the origin and forcing  $s_2$  to be along the  $x$ -axis. For now, we leave the extra rotation in the system because it is easier to recognize the relationship between the final result and our previous expressions for the MNF variety. We can write a system of polynomials that encodes the geometric relationships in our model,

$$\begin{aligned}
f_1 &:= (x - x_1)^2 + (y - y_1)^2 - \rho_1^2 \\
f_2 &:= (x - x_2)^2 + (y - y_2)^2 - \rho_2^2 \\
f_3 &:= (x + (x_1 + x_2))^2 + (y + (y_1 + y_2))^2 - \rho_3^2
\end{aligned} \tag{3.24}$$

where  $\rho_i$  is the monostatic distance to the  $i^{\text{th}}$  scatterer. Each of these polynomials is an element of the ring

$$\mathbb{C}[x, y, x_1, x_2, y_1, y_2, \rho_1, \rho_2, \rho_3], \tag{3.25}$$

and the *affine variety* associated with this set of polynomials is the set

$$V(f_1, f_2, f_3) = \{(x, \dots, \rho_3) : f_i(x, \dots, \rho_3) = 0 \forall i\}$$

In other words,  $V(f_1, f_2, f_3)$  is the set of solutions to the system in (3.24) when we set each  $f_i = 0$ . If we correctly specified the polynomials in our system, then the affine variety will contain all of the potential combinations of sensor positions and target geometries. For our particular inverse problem, we can measure the ranges,  $\rho_1, \rho_2, \rho_3$ , and we would like to determine geometric information about the target configuration matrix variables,  $x_1, x_2, y_1, y_2$ . To find an implicit relationship between only these variables, we can take the system defined in (3.24), and attempt to eliminate variables until only  $\rho_1, \rho_2, \rho_3$ , the  $x$ 's and the  $y$ 's remain. Geometrically, this means we want to project the set  $V(f_1, f_2, f_3)$  down to a smaller space with variables,

$$\begin{bmatrix} x_1 & x_2 & y_1 & y_2 & \rho_1 & \rho_2 & \rho_3 \end{bmatrix} \in \mathbb{R}^7,$$

and find polynomials that describe the smallest set containing the projected variety.

To accomplish this, we first define the *ideal* generated by the polynomials in (3.24) to be

$$I = \langle f_1, f_2, f_3 \rangle := \left\{ \sum_{i=1}^3 h_i f_i : h_i \in \mathbb{C}[x, \dots, \rho_3] \right\}. \tag{3.26}$$

In some sense, the ideal is the natural generalization of a subspace; instead of restricting ourselves to linear combinations of the generator polynomials,  $\{f_1, f_2, f_3\}$ , the ideal contains polynomial combinations of the generators. Sometimes we may refer to  $I \subset \mathbb{C}[x, \dots, \rho_3]$  as the set of polynomial consequences of the polynomials in (3.24). Ideals and varieties are closely connected in that, if an ideal is generated by two distinct sets of polynomials, then the varieties associated with each of the generating sets are the same [31]. In other words, if

$$\langle f_1, \dots, f_s \rangle = \langle g_1, \dots, g_t \rangle$$

then

$$V(f_1, \dots, f_s) = V(g_1, \dots, g_t).$$

If we think of the sets  $\{f_1, \dots, f_s\}$ ,  $\{g_1, \dots, g_t\}$  as bases of an ideal, then this proposition says that the variety associated with an ideal is independent of the chosen basis. We can use this to our advantage by finding a basis for our ideal,  $I$ , that is convenient for carrying out variable elimination. The most commonly used bases are called Gröbner bases.

We make no attempt to describe all the known properties of Gröbner bases, and instead we note that Gröbner bases are computed with Buchberger's algorithm, the generalization of Gaussian elimination to polynomial systems. In Gaussian elimination, we choose a term ordering for our linear expressions and perform a pivoting operation to eliminate terms according to that ordering. The result is a triangular basis for the set of linear expressions. Buchberger's algorithm applies to general polynomials, in which the terms are monomials. As such, the term ordering is a more subtle issue. For the purposes of variable elimination, it is necessary to compute a basis with respect to a chosen *lexicographic ordering*(lex). The resulting Gröbner basis will reflect the choice of ordering. To see this, consider the following example.

**Example.** Let  $I_{ex}$  be the ideal generated by

$$\begin{aligned}
&x^2 + y + z - 1, \\
&x + y^2 + z - 1, \\
&x + y + z^2 - 1.
\end{aligned}$$

Then a Gröbner basis for  $I_{ex}$  with lexicographic order  $x > y > z$  is

$$\begin{aligned}
g_1 &= x + y + z^2 - 1, \\
g_2 &= y^2 - y - z^2 + z, \\
g_3 &= 2yz^2 + z^4 - z^2, \\
g_4 &= z^6 - 4z^4 + 4z^3 - z^2.
\end{aligned}$$

Notice that  $g_1$  includes all three variables,  $g_2$  and  $g_3$  include only  $y$  and  $z$ , and  $g_4$  depends only on  $z$ . The elimination of variables from successive basis polynomials is not a coincidence, and we can use this property to compute our polynomial elimination.

**Proposition 1.** *Let  $I \subset k[x_1, \dots, x_n]$  be an ideal and let  $G$  be a Gröbner basis of  $I$  with respect to lex order where  $x_1 > \dots > x_n$ . Then the set*

$$G_l = G \cap k[x_{l+1}, \dots, x_n]$$

*is a Gröbner basis of the  $l^{\text{th}}$  elimination ideal,  $I_l$ .*

This reduces the problem of eliminating variables from (3.24) to the problem of computing a Gröbner basis. Once we have a Gröbner basis for (3.24), the subset of polynomials in the basis that depend only on  $\rho_1, \rho_2, \rho_3$ , the  $x_i$ 's and the  $y_i$ 's form a Gröbner basis for the set of geometric invariant equations.

Using the software `Singular` [32] to compute a Gröbner basis for (3.24), we find that the desired elimination ideal is generated by a single polynomial,

$$G \cap \mathbb{C}[x_1, x_2, y_1, y_2, \rho_1, \rho_2, \rho_3] = \langle g_1 \rangle, \quad (3.27)$$

where the exact expression for  $g_1$  is suppressed due to its length.

If we included all possible geometric relations in system (3.24), we can conclude that the elimination ideal contains all possible polynomial invariant equations for the monostatic, near-field case. We could solve for the squared pair-wise differences between scatterers, given data for  $\rho_1, \rho_2$ , and  $\rho_3$ , as we did in section 3.1.2. This approach will be insufficient for cases with more scatterers ( $N > 3$ ), as we will be forced to generate new invariant polynomials for each case, and the complexity of the polynomial invariants increases with the number of scatterers. Instead, the polynomial  $g_1$  can be rewritten in the quadratic form,

$$3(\boldsymbol{\rho}^2 - \mathbf{m})^T D S S^T D^T (\boldsymbol{\rho}^2 - \mathbf{m}) - 4 \det(S^T S) \mathbf{1}^T (\boldsymbol{\rho}^2 - \mathbf{m}) = 0, \quad (3.28)$$

where  $D$  is a column-differencing matrix,

$$D = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}. \quad (3.29)$$

From this, we can see that (3.28) is a multiple of (1.62), the equation found by Ferrara et al. [3], for  $N = 3$ , and is equivalent to (3.5) found in section 3.1.1.

### 3.3 Discovering a Bistatic Near-Field Invariant

Now, suppose we have a target with  $N$  scattering centers in dimension  $d$ , so that the configuration matrix is  $S \in \mathbb{R}^{N \times d}$ . Again, we attribute all motion to our receiver and assume that the scattering centers are fixed. Let  $\mathbf{x}_r(t_l) = (x_r, y_r)$  be the time-dependent position of the receiver, and let  $\mathbf{x}_t(t_l) = (x_t, y_t)$  be the position of the transmitter. Without loss of generality, we can fix the  $\mathbf{s}_1$  to be the origin, and let  $\mathbf{s}_2$  be on the first coordinate axis. Then

$$S = \begin{bmatrix} 0 & 0 \\ x_2 & 0 \\ x_3 & y_3 \\ \vdots & \vdots \\ x_N & y_N \end{bmatrix}. \quad (3.30)$$

Again, we can write a system of polynomials that encodes the geometric relationships in our model,

$$\begin{aligned} f_1 &:= t_1 + r_1 - \rho_1 \\ &\vdots \\ f_N &:= t_N + r_N - \rho_N \\ f_{N+1} &:= x_t^2 + y_t^2 - t_1^2 & f_{2N+1} &:= x_r^2 + y_r^2 - r_1^2 \\ f_{N+2} &:= (x_t - x_2)^2 + y_t^2 - t_2^2 & f_{2N+2} &:= (x_r - x_2)^2 + y_r^2 - r_2^2 \\ f_{N+3} &:= (x_t - x_3)^2 + (y_t - y_3)^2 - t_3^2 & f_{2N+3} &:= (x_r - x_3)^2 + (y_r - y_3)^2 - r_3^2 \\ &\vdots \\ f_{2N} &:= (x_t - x_N)^2 + (y_t - y_N)^2 - t_N^2 & f_{3N} &:= (x_r - x_N)^2 + (y_r - y_N)^2 - r_N^2, \end{aligned} \quad (3.31)$$

where  $r_i$  is the one-way distance from the receiver to the  $i^{\text{th}}$  scatterer,  $t_i$  is the one-way distance from the transmitter to the  $i^{\text{th}}$  scatterer, and  $\rho_i$  is the bistatic distance to the  $i^{\text{th}}$  scatterer. Each of these polynomials is an element of the ring

$$\mathbb{C}[x_r, y_r, x_t, y_t, x_2, \dots, x_N, y_3, \dots, y_N, t_1, \dots, t_N, r_1, \dots, r_N, \rho_1, \dots, \rho_N], \quad (3.32)$$

and the affine variety associated with this set of polynomials is the set

$$V(f_1, \dots, f_{3N}) = \{(x_r, \dots, \rho_N) : f_i(x_r, \dots, \rho_N) = 0 \forall i\}.$$

### 3.3.1 Elimination via Gröbner Basis Computation

As before, we attempted to compute a Gröbner basis of the ideal,  $I$ , generated by this set of polynomials. Buchberger's algorithm, however, is computationally expensive, and dependent on the ordering we impose on the monomial terms of our polynomials. A well-known heuristic is that *degree reverse lexicographic ordering* (degrevlex) is typically the fastest; for this example, we can compute a basis almost instantly with this monomial ordering. Unfortunately, computing a basis with respect to this ordering does not necessarily eliminate variables from the polynomial basis. Direct computation of the Gröbner basis with Buchberger's algorithm in `Singular` is so slow that our initial attempt with  $N = 5$  took more than four weeks before we were forced terminate the computation without success.

Even if our computation had terminated in a reasonable amount of time, it may have returned a trivial answer. For example, compute a Gröbner basis,  $G$ , for  $I$  with  $N = 3$ , and you find that the elimination ideal is not helpful,

$$G \cap \mathbb{C}[x_2, \dots, x_N, y_3, \dots, y_N, r_1, \dots, r_N, \rho_1, \dots, \rho_N] = \langle 0 \rangle. \quad (3.33)$$

In this case, the only polynomial in our ideal is the zero polynomial. Before we commit time and resources to an expensive elimination ideal computation, it would be helpful to know if the computation will result in the zero ideal. As such, we will first investigate the existence of a BNF invariant equation with Gröbner bases.

To address the problem of existence, we must consider whether the subset of variables,  $U = \{x_2, \dots, x_N, y_3, \dots, y_N, \rho_1, \dots, \rho_N\}$ , is an independent set. In other words, we want to know if

$$I \cap \mathbb{C}[U] = \langle 0 \rangle. \quad (3.34)$$

This question is closely tied to the dimension of the variety,  $V(f_1, \dots, f_{3N})$ , by the following corollaries from Cox et al. [31],

**Corollary 1.** *Let  $V \subset k^n$  be an affine variety. Then the dimension of  $V$  is equal to the largest integer  $r$  for which there exist  $r$  variables  $x_{i_1}, \dots, x_{i_r}$  such that  $I(V) \cap k[x_{i_1}, \dots, x_{i_r}] = \langle 0 \rangle$ .*

**Corollary 2.** *Let  $k$  be an algebraically closed field and let  $V \subset k^n$  be an affine variety. Then the dimension of  $V$  is the largest dimension of a subspace  $H \subset k^n$  for which a projection of  $V$  onto  $H$  is Zariski dense.*

The first corollary tells us that the dimension of our variety gives us a constraint on how many variables we can remove from the system; if the dimension of our affine variety is  $r$ , then any subset of variables  $U$  with size  $|U| \geq r + 1$  will necessarily produce an elimination ideal strictly smaller than the zero ideal. The second corollary gives us a way to think about the elimination process from a geometric point of view. Elimination of variables is equivalent to finding an ideal corresponding to a projected version of our variety,  $V$ . For example, suppose we have  $V(h_1, \dots, h_M) \subset \mathbb{C}^N$  and we wish to eliminate the last  $N - r$  variables.  $V$  is the solution set of  $h_i(x_1, \dots, x_N) = 0$  for all  $i$ , and we can project these points down to a subspace spanned by the first  $r$  variables via a mapping  $\pi(x_1, \dots, x_r, x_{r+1}, \dots, x_N) = (x_1, \dots, x_r)$ . The resulting set of points is contained in an affine variety,  $\pi(V) \subset V_r \subset \mathbb{C}^r$ , which has an associated ideal,  $I(V_r) \subset \mathbb{C}[x_1, \dots, x_r]$ . If the smallest variety containing  $\pi(V)$  is the entire space  $\mathbb{C}^r$ , then we say that  $\pi(V)$  is *Zariski dense* in  $\mathbb{C}^r$ . Note that the ideal corresponding to  $V_r = \mathbb{C}^r$  is the zero ideal. It follows from the second corollary that we will find a non-trivial ideal if we project  $V$  onto a subspace with dimension  $r + 1$ , where  $r$  is the dimension of  $V$ .

Fortunately, we can compute the dimension of our variety associated with (3.31) as long as we can compute *any* Gröbner basis of the ideal  $I = \langle f_1, \dots, f_{3N} \rangle$ . `Singular` contains a fast implementation of this algorithm. For various  $N$ , we compute a Gröbner basis of  $I$  using degree reverse lexicographic ordering, and then compute the dimension of the variety  $V(f_1, \dots, f_{3N})$ . A summary of the results are in Table 3.1.

From Table 3.1, we can see that the smallest  $N$  for which  $\dim(V) < |U|$  is  $N = 5$  for the 2D case, and  $N = 7$  for the 3D case. From this, we can conclude that the elimination ideal satisfies

**Table 3.1:** Results of Singular computations for various numbers of scatterers,  $N$ .  $U$  is the desired set of variables remaining after elimination and  $V$  denotes the variety defined by (3.31); for a non-trivial invariant to exist, we require  $\dim(V) < |U|$ .

$d = 2$	Total # Vars	$\dim(V)$	Desired $ U $
$N = 3$	16	7	6
$N = 4$	21	9	9
$N = 5$	26	11	12

$d = 3$	Total # Vars	$\dim(V)$	Desired $ U $
$N = 5$	30	15	14
$N = 6$	36	18	18
$N = 7$	42	21	22

$$I_U := I \cap \mathbb{C}[U] \subset \langle 0 \rangle \tag{3.35}$$

for  $N = 5, d = 2$  and  $N = 7, d = 3$ . Since the elimination ideal is a strict subset of  $\langle 0 \rangle$ , we can conclude that  $I_U$  contains a non-trivial polynomial relationship in the variables from  $U$ .

Now that we know  $N = 5, d = 2$  is the smallest non-trivial example, we transition to the task of computing a polynomial in the elimination ideal. Computing the elimination ideal may be more efficient if we first reduce the system by hand. One easy simplification is to note that

$$0 = \rho_n^4 - 2\rho_n^2(t_n^2 + r_n^2) + (t_n^2 - r_n^2)^2, \tag{3.36}$$

and substitute the equations for  $t_n$  and  $r_n$  into this modified expression to find the reduced system,

$$\begin{aligned}
f_1 &= \rho_1^4 - 2\rho_1^2(x_r^2 + y_r^2 + x_t^2 + y_t^2) + (x_r^2 + y_r^2 - x_t^2 - y_t^2)^2 \\
f_2 &= \rho_2^4 - 2\rho_2^2((x_r - x_2)^2 + y_r^2 + (x_t - x_2)^2 + y_t^2) + ((x_r - x_2)^2 + y_r^2 - (x_t - x_2)^2 - y_t^2)^2 \\
f_3 &= \rho_3^4 - 2\rho_3^2((x_r - x_3)^2 + (y_r - y_3)^2 + (x_t - x_3)^2 + (y_t - y_3)^2) \\
&\quad + ((x_r - x_3)^2 + (y_r - y_3)^2 - (x_t - x_3)^2 - (y_t - y_3)^2)^2 \\
&\quad \vdots \\
f_N &= \rho_N^4 - 2\rho_N^2((x_r - x_N)^2 + (y_r - y_N)^2 + (x_t - x_N)^2 + (y_t - y_N)^2) \\
&\quad + ((x_r - x_N)^2 + (y_r - y_N)^2 - (x_t - x_N)^2 - (y_t - y_N)^2)^2.
\end{aligned} \tag{3.37}$$

Even though we have eliminated the  $t_i$  and  $r_i$  variables in (3.37), it may still be less efficient to use this system. To investigate this, we will name (3.31) the “full” system, and (3.37) the “reduced” system, and apply our methods to both systems.

### Improved Gröbner Basis Computations

The most user-friendly approach to eliminating variables from our systems is to use the built in `eliminate` command in `Singular`. The command computes a Gröbner basis with respect to lexicographic order, and then removes polynomials from the basis that depend on variables not contained in the desired set,  $U$ .

When a Gröbner basis is easy to compute with respect to one ordering, as in our problem, we gain access to other techniques for accelerating the computation of the elimination ideal. In our case, we can compute  $dG$ , a basis with respect to *degrevlex*, relatively quickly. From here, we have options. Given  $dG$ , we can compute the Hilbert function of our ideal and use this to perform Hilbert-driven elimination, or we can attempt to convert  $dG$  to a lexicographic basis using a Gröbner basis conversion algorithm. The Hilbert function of an ideal encodes important information about the ideal, and in this case it provides a set of variable weights that we can use to reorder our monomial terms to speed up the Gröbner basis computation. This process is usually referred to as Hilbert-driven elimination. The second technique involves iteratively modifying the monomial term ordering of a known Gröbner basis to translate the basis into a lexicographic basis. The

process "walks" a path from the known ordering to the desired ordering, and so the algorithm is generally called a Gröber walk. We will not attempt to describe these techniques in detail here, but instead refer the reader to Cox et al. [31] for more background on the Hilbert function and further references for the Gröbner walk algorithm [33–35]. `Singular` provides implementations of both algorithms. In the end, both methods fail to compute the desired lexicographic Gröbner basis for either start system within a reasonable amount of time.

One final technique, called Faugère’s F4 algorithm, computes a Gröbner basis using fast linear algebra techniques to improve on Buchberger’s algorithm. This algorithm is not as well supported as the currently implemented versions of the Buchberger algorithm, but is known to be extremely fast for some problems [36]. The software program `MAGMA` [37] contains an efficient implementation of this algorithm by Allan Steel. Unfortunately, a trial run with this algorithm also failed after two weeks.

### 3.3.2 Elimination via Resultants

While a lexicographic basis for the ideal defined by (3.31) would make it possible for us to characterize all the geometric invariant equations of our system, we may be able to find a non-trivial invariant equation by generating a single polynomial in the desired elimination ideal. With this in mind, we can avoid the expensive polynomial divisions required by the Gröbner basis computation algorithms by computing resultants of the polynomials in (3.31), instead.

Resultants are an efficient way to decide whether two polynomials share a common factor, but they have the added benefit that the resultant of two polynomials is always an element of the first elimination ideal [31]. For example, if we have two polynomials  $f, g \in k[x, y]$ , then the resultant of  $f(x, y)$  and  $g(x, y)$ , denoted  $\text{Res}(f, g, x)$ , will always be in  $\langle f, g \rangle \cap k[y]$ . As a consequence, we can use resultants to eliminate variables from a system without performing any polynomial division. The interested reader will find a good introduction to resultants and their theoretical implications for elimination in Cox et al. [31].

Unfortunately, pair-wise resultants can also be extremely expensive to compute if the number of monomial terms in our polynomials is large. To see why this might be, note that resultants are usually computed as the determinant of the *Sylvester matrix* of two polynomials. Given

$$\begin{aligned} f &= a_0x^l + \cdots + a_l & a_0 &\neq 0 \\ g &= b_0x^m + \cdots + b_m & b_0 &\neq 0, \end{aligned}$$

the Sylvester matrix of  $f$  and  $g$  with respect to  $x$  is the  $(l + m) \times (l + m)$  matrix

$$\text{Syl}(f, g, x) = \begin{bmatrix} a_0 & & & b_0 & & & \\ & a_1 & \cdots & & b_1 & \cdots & \\ & \vdots & & a_0 & \vdots & & b_0 \\ a_l & & & a_1 & b_m & & b_1 \\ & & & \cdots & \vdots & \cdots & \vdots \\ & & & & a_l & & b_m \end{bmatrix}, \quad (3.38)$$

and

$$\text{Res}(f, g, x) = \det(\text{Syl}(f, g, x)). \quad (3.39)$$

The computational complexity of the determinant scales poorly ( $\mathcal{O}((l + m)^3)$  in the number of operations for the straight-forward approach) with the size of the matrix, so if there are a large number of monomial terms, the determinant computation will be expensive. Furthermore, eliminating multiple variables from our system will require many resultant computations. Suppose we want to eliminate  $x$  and  $y$  from the MNF polynomials,

$$\begin{aligned} f_1 &:= x^2 + y^2 - \rho_1^2 \\ f_2 &:= (x - x_2)^2 + y^2 - \rho_2^2 \\ f_3 &:= (x - x_3)^2 + (y - y_3)^2 - \rho_3^2. \end{aligned} \quad (3.40)$$

To first eliminate  $x$ , we would compute

$$\begin{aligned}g_1 &:= \text{Res}(f_1, f_2, x) \\g_2 &:= \text{Res}(f_1, f_3, x),\end{aligned}\tag{3.41}$$

and then eliminate  $y$  by computing

$$h := \text{Res}(g_1, g_2, y).\tag{3.42}$$

In this case, we had 3 equations and 2 variables to eliminate, for a total of 3 resultant computations. For the BNF problem, we have, at best, 5 equations with 4 variables to eliminate for a total of  $4 + 3 + 2 + 1 = 10$  resultant computations. This does not seem to be that bad, until we consider that the number of monomials explodes as the computations proceed. For example,  $f_1, f_2$  and  $f_3$  have 3, 5, and 7 distinct terms.  $g_1$  and  $g_2$  have 7 and 16 terms. The final polynomial,  $h$ , has 125 distinct monomial terms. Attempts to apply this naive method of round-robin elimination with resultants failed for both the reduced and full BNF system.

### **Extended Dixon Resultant (Projector)**

In 1909, Dixon reported a method [38] for simultaneously eliminating two variables from three generic, bidegree polynomials based on Cayley's formulation [39] of Bezout's efficient method for computing the resultant for two univariate polynomials. Unfortunately, most algebraic and geometric problems include polynomials that are not generic, and so the original method was not widely applicable. Dixon mentioned that the technique generalized to  $n + 1, n$ -degree polynomials in  $n$  variables, but in practice, the *Dixon matrix* is often singular, and sometimes is not even square. In 1994, Kapur et al. resolved this issue by proving that it is sufficient to consider a full-rank submatrix of the original Dixon matrix [40].

Cayley's method for computing the resultant of two polynomials,  $f_1, f_2$  in one variable,  $x$ , involves computing what is now called the *Dixon polynomial*,

$$\delta(x, \alpha) := \frac{1}{(x - \alpha)} \begin{vmatrix} f_1(x) & f_2(x) \\ f_1(\alpha) & f_2(\alpha) \end{vmatrix}, \quad (3.43)$$

where  $\alpha$  is a dummy variable. Setting  $x = \alpha$  would make the determinant in (3.43) vanish, so we have included  $(x - \alpha)^{-1}$  to cancel that factor. Notice that every common zero of  $f_1(x)$  and  $f_2(x)$  is a zero of  $\delta(x, \alpha)$ , regardless of the value of  $\alpha$ . This means that, at a common zero of  $f_1$  and  $f_2$ , all the coefficients of the various powers of  $\alpha$  must be zero. This gives a set of equations,

$$D \begin{bmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^{d-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (3.44)$$

where  $D$  is a matrix of coefficients, and  $d$  is the maximum power of  $x$  appearing in  $f_1, f_2$ . If a common solution to  $f_1, f_2$  exists, there must be a non-trivial solution to (3.44). This implies that the vanishing of the determinant of  $D$  is a necessary condition for  $f_1$  and  $f_2$  to share a common solution. Cayley was the first to show that the vanishing of the determinant of  $D$  is a necessary condition for  $f_1, f_2$  to share a nontrivial common zero. The matrix  $D$  is called the *Dixon matrix*, and its determinant is the *Dixon resultant*.

**Example.** Let

$$\begin{aligned} f(x) &:= (x - 1)(x - b) &= x^2 - (b + 1)x + b \\ g(x) &:= (x - 2)(x - 3)(x - 4) &= x^3 - 9x^2 + 26x - 24 \end{aligned} \quad (3.45)$$

Then the Dixon polynomial is

$$\begin{aligned}
\delta(x, \alpha) &= \frac{1}{x - \alpha} \begin{vmatrix} x^2 - (b+1)x + b & x^3 - 9x^2 + 26x - 24 \\ \alpha^2 - (b+1)\alpha + b & \alpha^3 - 9\alpha^2 + 26\alpha - 24 \end{vmatrix} \\
&= -bx^2 + (9b - 24)x - 2b + 24 \\
&\quad + ((b+1)x^2 + (17 - 10b)x + (9b - 24)) \alpha \\
&\quad + (-x^2 + (b+1)x - b) \alpha^2
\end{aligned}$$

From this, we know that each of the coefficients of the powers of  $\alpha$  must be zero,

$$\begin{aligned}
\alpha^0 : \quad & -2b + 24 + (9b - 24)x - bx^2 = 0 \\
\alpha^1 : \quad & (9b - 24) + (17 - 10b)x + (b+1)x^2 = 0 \\
\alpha^2 : \quad & -b + (b+1)x - x^2 = 0,
\end{aligned}$$

and so we can construct the Dixon matrix,

$$D = \begin{bmatrix} 24 - 2b & 9b - 24 & -b \\ 9b - 24 & 17 - 10b & b + 1 \\ -b & b + 1 & -1 \end{bmatrix},$$

and find that the Dixon resultant is

$$\det(D) = 144 - 156b + 54b^2 - 6b^3 = -6(b-2)(b-3)(b-4). \quad (3.46)$$

Notice that the Dixon resultant captured the fact that the system (3.45) can have a common zero if and only if  $b = 2, 3, 4$ . In this case, we have an extraneous factor of  $-6$  in addition to the usual resultant of  $f$  and  $g$ . For the more general Dixon resultant with  $n$  polynomials, these extra factors can be expressions in terms of the parameters, which makes them a nuisance. This is because the Dixon resultant actually belongs to a more general set of objects known as *projection operators*. Let  $A = \mathbb{Q}[a_1, \dots, a_m]$ , where the  $a_i$  are parameters, and construct a set of  $n + 1$  polynomials,  $\mathcal{P}$ , in the ring with coefficients from  $A$ ,  $A[x_1, \dots, x_n]$ . Any non-zero polynomial in

A that vanishes for all the values of the parameters  $a_1, \dots, a_m$  for which the polynomials in  $\mathcal{P}$  must have a common zero is a projection operator. The resultant of a polynomial system  $\mathcal{P}$  must divide all of the projection operators, and so in that sense the resultant is the minimal projection operator [41]. To make this distinction, from now on we will refer to the Dixon resultant as the Dixon projector, and reserve the word resultant for the minimal projection operator. Any projection operator would be suitable for the least-squares fitting approach we have in mind, but the resultant is the simplest (lowest degree) polynomial relation among the parameters and so will likely be the optimal choice for computational efficiency and numerical stability as we proceed.

Dixon's generalization to the case with  $n + 1$  polynomials in  $n$  variables is straightforward. For a system  $\mathcal{P} = f_1, \dots, f_{n+1}$  in the ring  $k[x_1, \dots, x_n]$ , define

$$\nabla(x_1, \dots, x_n, \alpha_1, \dots, \alpha_n) = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n) & \dots & f_{n+1}(x_1, x_2, \dots, x_n) \\ f_1(\alpha_1, x_2, \dots, x_n) & \dots & f_{n+1}(\alpha_1, x_2, \dots, x_n) \\ f_1(\alpha_1, \alpha_2, \dots, x_n) & \dots & f_{n+1}(\alpha_1, \alpha_2, \dots, x_n) \\ \vdots & \ddots & \vdots \\ f_1(\alpha_1, \alpha_2, \dots, \alpha_n) & \dots & f_{n+1}(\alpha_1, \alpha_2, \dots, \alpha_n) \end{bmatrix}. \quad (3.47)$$

$\nabla$  is divisible by each  $(x_i - \alpha_i)$ , so we define the multivariate Dixon polynomial to be

$$\delta(x_1, \dots, x_n, \alpha_1, \dots, \alpha_n) = \frac{\nabla(x_1, \dots, x_n, \alpha_1, \dots, \alpha_n)}{(x_1 - \alpha_1) \cdots (x_n - \alpha_n)}. \quad (3.48)$$

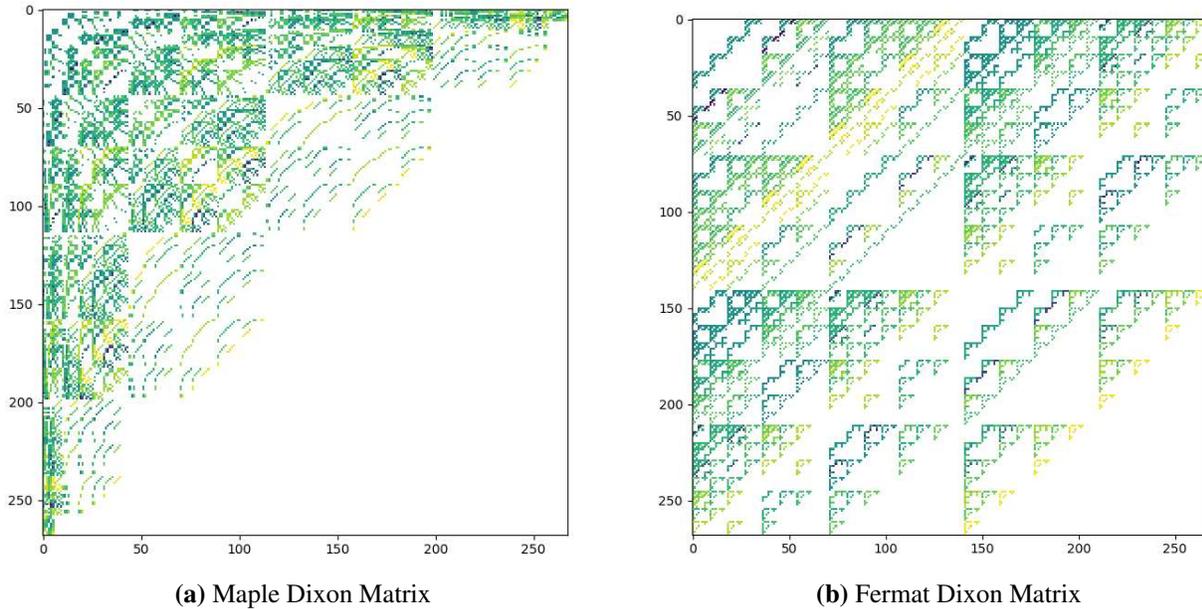
Again, any common zero of  $\mathcal{P}$  will make  $\delta = 0$ , regardless of the values of the  $\alpha_i$ , and so the coefficients of the monomials in  $\alpha_i \cdots \alpha_n$  will all vanish. This creates a system of  $s$  equations, with  $s$  equal to the number of monomials in the  $\alpha$  variables present. We can express the system as

$$D \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_n \\ x_1^2 \\ x_1x_2 \\ x_2^2 \\ \vdots \\ \prod_i x_i^{d(i)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (3.49)$$

where  $D$  is a matrix of coefficients. Any non-trivial solution to the system  $\mathcal{P}$  implies that (3.49) has a non-trivial solution, so that the determinant of  $D$  must vanish. As before,  $D$  is the Dixon matrix and its determinant is the Dixon polynomial.

Dixon's original derivation accounted for generic,  $n$ -degree polynomials, for which this method was guaranteed to succeed. For general polynomial systems, however, the Dixon matrix may be singular for any choice of parameters. In fact, it may not even be square. In these cases, the determinant provides no information about the solutions of the original system of equations. This was the primary shortcoming of Dixon's method, until Kapur et al. resolved the issue by showing that it is sufficient to consider a full-rank submatrix of the Dixon matrix [40]. In further works, they make progress towards characterizing the extraneous factors in the Dixon resultant and compare the computation of the Dixon resultant to other popular elimination techniques [41–45]. Currently, we are using an implementation of these improved resultant methods by Manfred Minimair [46] in the software program `Maple` [47] to compute the Dixon resultant of both the full and reduced BNF systems.

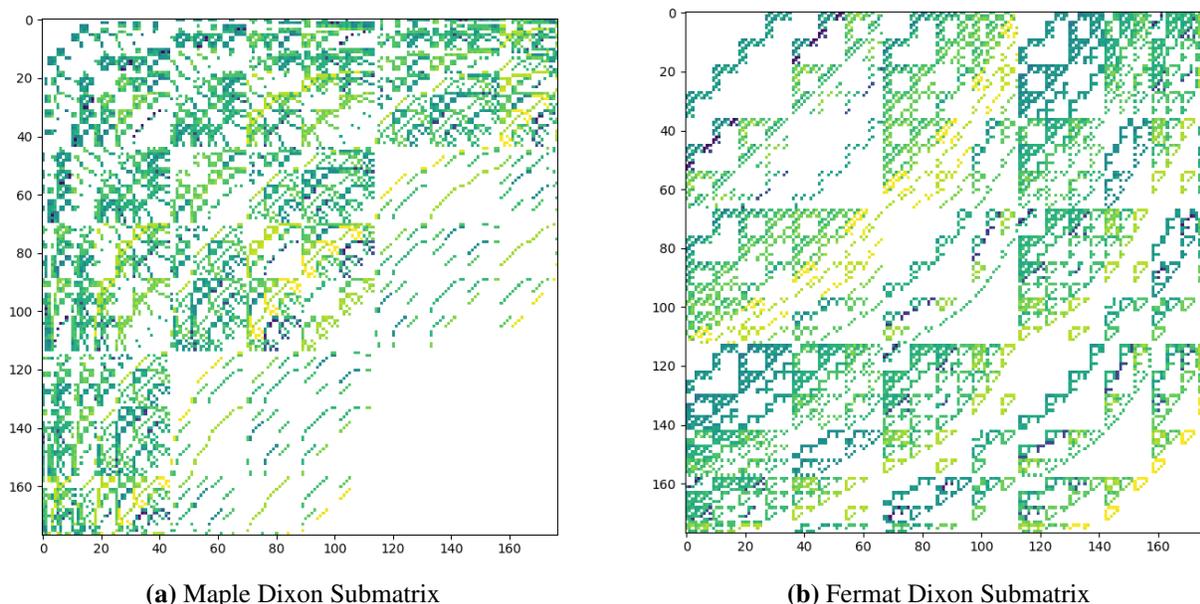
For the system (3.31), the resulting Dixon matrix has dimension  $268 \times 268$ , and has rank 177. Further, there exists a symmetric  $177 \times 177$  full-rank submatrix with some apparent block structure. In Figure 3.5, we include a graphical representation of the Dixon matrices generated by Minimair's



**Figure 3.5:** The Dixon matrices generated by Maple (left) and Fermat (right). Here, we have substituted values for each of the free variables and plotted the resulting matrix of numerical values. The matrix generated by Fermat is equivalent to a permutation of the rows and columns of the Maple-generated matrix. Both matrices are symmetric and hollow (the diagonal entries are all zero).

DR package for Maple [46], and Lewis’ EDF code for his language, Fermat [48]. We should note that the matrix generated by Fermat is equivalent to a permutation of the rows and columns of the Maple-generated matrix.

Despite the sparsity of these matrices and the exploitation of clever early detection of factors (EDF) heuristics [46, 49], the computation of the determinant of the  $177 \times 177$  submatrix was still too expensive to compute on the 512 GB of shared memory available (CSU math department’s tx1), even when we substitute numerical values for most of the parameters. It is clear that further simplifications of the system, or a more efficient technique, are required to extract the resultant of (3.31).



**Figure 3.6:** Full-rank submatrices extracted from the Dixon matrices generated by Maple (left) and Fermat (right). Here, we have substituted values for each of the free variables and plotted the resulting matrix of numerical values. In this case, we are able to extract submatrices which are symmetric.

### 3.4 Improvements to the Resultant Computation

Given the difficulty of computing the desired resultant of (3.31), it is clear that further simplifications are necessary to reduce the time and memory required. In this section, we explore methods for reducing the overall computational expense of obtaining the desired resultant.

#### 3.4.1 Determination of Resultant Degree

With the large number of extraneous terms that could potentially appear in the Dixon projector, it is useful to have some criteria for distinguishing the true resultant from extraneous terms in the projector. One helpful piece of information is the degree of the resultant polynomial.

Two important pieces of information will allow us to determine the degree of the resultant from the system (3.31). First, we know from Table 3.1 that the variety associated to our elimination ideal is 11-dimensional, in the 12 dimensional space. If we choose a particular target shape, then the 7 shape parameters  $x_2, \dots, x_5, y_3, \dots, y_5$  are fixed and we are really considering a 4D variety in 5D space (the remaining variables are  $\rho_1, \dots, \rho_5$ ). A manifold or algebraic variety with dimension

$n - 1$  in an  $n$ -dimensional space is a *hypersurface*, and these objects have special properties. In particular, every hypersurface is (at least locally) defined by a single implicit equation, so that the ideal associated to our variety is generated by a single polynomial (a principal ideal). Since the resultant divides every projection operator, the resultant must be the generator of our ideal. Next, the degree of a hypersurface is the same as the degree of its defining equation [31], so that we can determine the degree of the resultant by computing the degree of the associated variety.

Recall that, given the variety,  $V$ , defined by (3.31), the variety defined by the elimination ideal,  $V_U$  is the Zariski closure of the projection of  $V$  to the subspace defined by the remaining variables in the set  $U = \{x_2, \dots, x_N, y_3, \dots, y_N, \rho_1, \dots, \rho_N\}$ . Bertini includes a specialized homotopy type for tracking points through these types of projections, and the result is that Bertini can compute a number of useful quantities related to the problem of elimination of variables. We will not cover the details of such procedures here, but the interested reader should consult the companion book for Bertini [27] and the literature cited therein. To determine the degree of the variety  $V_U$ , we first initiate a standard run of Bertini with the input file in Figure 3.7. The Bertini run successfully completes, and confirms that  $\dim V = 4$ . The output also informs us that  $V$  has one irreducible component, of degree 1024. To compute the projection to  $V_U$ , we change the track type in the input file to 5 and define the desired projection in a supplementary file, `projection`. Running Bertini again outputs the information in Figure 3.8. We can see from the output that the projected set is still a hypersurface; it has dimension 4 in a 5D space. The *fiber* of a point  $y$  under the map  $f$  is the set of points  $f^{-1}(y)$ . In this case, `Fiber` refers to the set of points on the original variety which map to a generic point in the projected set. Notice that the dimension of the fiber is 0, so that only isolated points map to the same point under the projection. This is useful because, though Bertini states that the degree of  $V_U$  is 80, the degree of each fiber is 2. This tells us that exactly two isolated points mapped to the same point, so that every root in the polynomial associated to this variety is of multiplicity 2. This suggests that the resultant polynomial is actually a degree 40 polynomial, squared. In other words, the ideal associated to  $V_U$  is  $\langle r(\rho_1, \dots, \rho_5)^2 \rangle$ , where  $r$

```

% bnfDegree.input
CONFIG
  TrackType:1;
END;
INPUT
  function f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12,f13,f14,f15;
  variable_group p1,p2,p3,p4,p5,t1,t2,t3,t4,t5,r1,r2,r3,r4,r5,xt,yt,x,y;
  constant x2,x3,x4,x5,y3,y4,y5;

  x2 = 2; x3 = 3; x4 = 1; x5 = -1;
  y3 = 1; y4 = 2; y5 = 1.5;

  f1 = xt^2 + yt^2 - t1^2;
  f2 = (xt-x2)^2 + yt^2 - t2^2;
  f3 = (xt-x3)^2 + (yt-y3)^2 - t3^2;
  f4 = (xt-x4)^2 + (yt-y4)^2 - t4^2;
  f5 = (xt-x5)^2 + (yt-y5)^2 - t5^2;
  f6 = x^2 + y^2 - r1^2;
  f7 = (x-x2)^2 + y^2 - r2^2;
  f8 = (x-x3)^2 + (y-y3)^2 - r3^2;
  f9 = (x-x4)^2 + (y-y4)^2 - r4^2;
  f10 = (x-x5)^2 + (y-y5)^2 - r5^2;
  f11 = t1 + r1 - p1;
  f12 = t2 + r2 - p2;
  f13 = t3 + r3 - p3;
  f14 = t4 + r4 - p4;
  f15 = t5 + r5 - p5;
END;

```

**Figure 3.7:** Input file for a standard run of Bertini. This file contains the system from (3.31) with “random” parameters chosen for the shape variables.

```

Dimensions
  Projection: 4
  Fiber: 0

Degrees
  Projection: 80
  Fiber: 2

```

**Figure 3.8:** The output from our Bertini run projecting the variety  $V$  down to  $V_U$ . Notice that the fiber of the projection map has degree 2.

is degree 40. The ideal  $\langle r(\rho_1, \dots, \rho_5) \rangle$  corresponds to the same variety, so that knowledge of the degree 40 polynomial is sufficient for our purposes.

### 3.4.2 Dixon Matrix Reduction

For some implicitization problems, it is possible to further reduce the size of the Dixon matrix prior to computing the Dixon projector. Consider the system,

$$\begin{aligned} x &= t_3 + t_4 & t_3^2 &= t_1^2 + t_2^2 \\ y &= t_1 t_2 & t_4^2 &= (t_1 - 1)^2 + (t_2 - 1)^2 \\ z &= t_3 t_4, \end{aligned} \tag{3.50}$$

where we have the free parameters  $t_1, t_2$ , and we would like to eliminate  $t_1, t_2, t_3$ , and  $t_4$  to find an implicit representation in  $x, y$ , and  $z$ . A full-rank submatrix of the Dixon matrix generated by Fermat for this system is

$$\begin{bmatrix} -2y & 0 & 0 & 0 & 0 & 0 & 0 & -2z + 2y & 0 & 0 & 2x & 0 & 0 & z & z - 2 & -x \\ 0 & 2y & z + 2 & -2y & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & -x & 0 & 1 \\ 0 & 0 & -x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & z & -2z & 0 & 0 & 0 & 2x & 0 & 0 & -2 & 0 & 0 & 0 & -x & 1 \\ -z + 2 & 0 & 0 & 0 & -z & 0 & x & 0 & -2x & 0 & 0 & 2z - 2y & 0 & 0 & 2 & 0 \\ 0 & -z - 2 & -2 & 0 & x & 2y & -1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & x & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x & -z & 0 & 0 & 0 & 2z & -1 & 0 & 2 & 0 & 0 & -2x & 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & x & -z & 0 & -z + 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 2 & 0 & -z - 2 & 0 & 0 & -1 & x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -z & 0 & 0 & -1 & 0 & 0 & x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & z - 2 & 0 & 0 & -x & 2y & z & 0 & 2 & 0 \\ 0 & 0 & -2 & z + 2 & 0 & -2y & 0 & 0 & 0 & 0 & 1 & 0 & -x & 0 & 0 & 0 \\ 0 & 0 & 0 & -x & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & z & 0 & 0 & 0 & -x & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.51}$$

This matrix is  $16 \times 16$ , with 74 non-zero entries. The associated Dixon resultant is

$$\begin{aligned} r(x, y, z) &= 16x^4 z^2 - 32x^2 z^2 + 16z^2 + 64x^2 y z - 64y z - 8x^6 z + 24x^4 z \\ &\quad - 32x^2 z + 32z + 64y^2 - 16x^4 y + 32x^2 y - 64y + x^8 - 4x^6 \\ &\quad + 8x^4 - 16x^2 + 16, \end{aligned} \tag{3.52}$$

but the Dixon projector  $d(x, y, z)$  contains extraneous terms,

$$d(x, y, z) = Cr(x, y, z)^2, \quad (3.53)$$

where  $C$  is a scalar. Following the results by Chtcherba et al. [50], we are guaranteed that the resultant appears as a factor in the determinant of the chosen full-rank submatrix, (3.51). In this case, however, an even smaller submatrix of the Dixon matrix contains the resultant as a factor. If we keep only rows 5, 6, 9, 10, 11, 12, 13, 14, 15, and 16, along with columns 3, 4, 6, 8, 9, 10, 11, 12, 13, and 15 from (3.51), then the resulting submatrix is only  $10 \times 10$ , with 36 non-zero entries,

$$\begin{bmatrix} 0 & 0 & 0 & 0 & -2 * x & 0 & 0 & 2 * z - 2 * y & 0 & 2 \\ -2 & 0 & 2 * y & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & x & -z & 0 & -z + 2 & 0 & 0 \\ 0 & 2 & -z - 2 & 0 & -1 & x & 0 & 0 & 0 & 0 \\ 0 & 0 & x & 0 & 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -z & 0 & -1 & 0 & 0 & x & 0 & 0 \\ 0 & 0 & 0 & z - 2 & 0 & 0 & -x & 2 * y & z & 2 \\ -2 & z + 2 & -2 * y & 0 & 0 & 0 & 1 & 0 & -x & 0 \\ 0 & -x & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & z & 0 & -x & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \quad (3.54)$$

The determinant of this submatrix is exactly  $4r(x, y, z)$ . Solving a linear system with Gaussian elimination is  $O(n^3)$  even for matrices with floating point entries; for matrices with symbolic entries, exact division will not be possible at every stage, and so the intermediate expressions can grow very large. Gaussian elimination over the integers experiences a similar phenomenon, for which the worst-case complexity is exponential [51]. We can safely assume that any reduction in the dimension or number of non-zero entries in our essential submatrix will greatly reduce the

effort necessary to extract the resultant. We now investigate the possibility of identifying such a submatrix of an arbitrary Dixon matrix.

Suppose that we have a polynomial system with  $m+1$  equations,  $\mathcal{P} \subset \mathbb{Q}[p_1, \dots, p_m][x_1, \dots, x_n]$ , with parameters  $p_i$  to be eliminated. For convenience, we will write a point in the domain of the polynomials in  $\mathcal{P}$  as  $\mathbf{z} = (\mathbf{p}, \mathbf{x}) = (p_1, \dots, p_m, x_1, \dots, x_n)$ . Further suppose that we have successfully generated a Dixon matrix and identified the usual full-rank submatrix as described by Kapur, Saxena, and Yang (KSY) [40]. Call this submatrix  $M$ . The entries of  $M$  are polynomials in  $\mathbb{Q}[x_1, \dots, x_n]$ , so denote the evaluation of  $M$  at a point  $\mathbf{x}$  as  $M(\mathbf{x})$ . As described in section 3.3.2, the determinant of  $M$  contains the resultant,  $r(\mathbf{x})$ , as a factor.

**Lemma 1.** *Given a point  $\mathbf{z}^* = (\mathbf{p}^*, \mathbf{x}^*) \in V(\mathcal{P})$ ,  $M(\mathbf{x}^*)$  is singular.*

*Proof.* Let  $\mathbf{z}^* \in V(\mathcal{P})$  so that  $r(\mathbf{x}^*) = 0$ . The determinant of  $M$  necessarily contains the resultant of  $\mathcal{P}$  as a factor, so that  $\det(M(\mathbf{x}^*)) = \sum_k q_k(\mathbf{x}^*)r(\mathbf{x}^*) = 0$ , where the  $q_k$  are extraneous polynomial factors.  $\square$

Similarly, any matrix containing the resultant as a factor of its determinant will be singular when evaluated at  $\mathbf{x}^*$ . On the other hand, for a randomly selected  $\mathbf{z} = (\mathbf{p}, \mathbf{x}) \notin V(\mathcal{P})$ ,  $M(\mathbf{x})$  will almost certainly be full rank. To see this, consider that  $\det(M(\mathbf{x})) = \prod_{k=1}^N g_k(\mathbf{x})$ , where each  $g_k$  represents one of  $N$  factors in the Dixon projector. We have chosen  $M$  to be a full-rank submatrix with the KSY approach [40], so that we are guaranteed  $g_k \neq 0 \forall k$ . The set of solutions for each  $g_k(\mathbf{x}) = 0$ ,  $V(g_k)$ , has codimension at least 1, and so is a set of measure zero with respect to the usual Lebesgue measure on  $\mathbb{C}^n$ . Finite unions of sets of measure zero are still of measure zero, so that  $\cup_{k=1}^N V(g_k)$  is of measure zero.  $M(\mathbf{x})$  is rank deficient for exactly the points in  $\cup_{k=1}^N V(g_k)$ , so that the probability of choosing  $\mathbf{x} \in \mathbb{C}^n$  such that  $M(\mathbf{x})$  is full rank is 1. This suggests criteria by which we can identify a submatrix of  $M$  that almost certainly contains the desired resultant as a factor of its determinant.

**Lemma 2.** *Let  $\mathbf{x}^* \in V(\mathcal{P})$ , let  $\mathbf{x} \in \mathbb{C}^n$  be selected at random, and let  $M_k$  be a submatrix of  $M$ . If  $\text{rank}(M_k(\mathbf{x})) > \text{rank}(M_k(\mathbf{x}^*))$ , then  $\det(M_k)$  contains the desired resultant as a factor.*

To find the smallest such submatrix, one would apply the criteria of Lemma 2 to every possible submatrix of  $M$ . In practice, the rank computation is expensive enough that it is not feasible to investigate every possible submatrix. Instead, we settle for an iterative procedure by which we investigate only submatrices with one fewer column and row than  $M$ , and repeat until no smaller submatrix can be found. It was by this procedure that we discovered the submatrix (3.54). An alternate version of this process would iterate through the entries of  $M$ , setting each to 0, checking the rank condition of Lemma 2, and restoring the entry if the condition is not met.

For the Dixon matrix associated to the bistatic, near-field system (3.31), the full-rank submatrix,  $M$ , is  $177 \times 177$ , with 7692 non-zero entries. After running many trials of our Dixon matrix reduction procedure, the smallest submatrix we could extract was  $173 \times 173$ , with 7176 non-zero entries. This reduction, while helpful, is not enough to make the computation of the Dixon projector feasible.

# Chapter 4

## A Solution for the Passive Case

### 4.1 An Invariant Equation for Passive SAR

Some radar systems do not require the user to actively transmit a signal. These passive radar systems process reflected energy from non-cooperative sources of illumination, such as the signals broadcast by television and radio towers. Since the transmitter and receiver are not colocated, this type of passive radar system is a specific case of bistatic radar. For ground-imaging applications using the signals from a stationary transmitter such as a radio tower, we can model the scenario as a bistatic, near-field radar with the transmitter location fixed relative to the scattering centers in the scene.

#### 4.1.1 Geometric Invariant Discovery

For simplicity, first suppose we have a target with  $N = 3$  scattering centers in 2D, so that the configuration matrix is  $S \in \mathbb{R}^{3 \times 2}$ . In contrast to our exposition in section 1.3, we attribute all motion to our receiver and assume that the scattering centers and the transmitter are fixed. Let  $\mathbf{x}_r(t_l) = (x_r, y_r)$  be the time-dependent position of the receiver, and let  $\mathbf{x}_t = (x_t, y_t)$  be the fixed position of the transmitter. Without loss of generality, we can fix the  $\mathbf{s}_1$  to be the origin, and let  $\mathbf{s}_2$  be on the first coordinate axis. Then

$$S = \begin{bmatrix} 0 & 0 \\ x_2 & 0 \\ x_3 & y_3 \end{bmatrix}. \quad (4.1)$$

Finally, let  $a, b$ , and  $c$  be the distances between the scatterers in  $S$ . With these, we can write a system of polynomials that encodes the geometric relationships in our model,

$$\begin{aligned}
f_1 &:= t_1 + r_1 - \rho_1 & f_7 &:= (x_3 - x_2)^2 + y_3^2 - a^2 \\
f_2 &:= t_2 + r_2 - \rho_2 & f_8 &:= x_3^2 + y_3^2 - b^2 \\
f_3 &:= t_3 + r_3 - \rho_3 & f_9 &:= x_2^2 - c^2 \\
f_4 &:= x_t^2 + y_t^2 - t_1^2 & f_{10} &:= x_r^2 + y_r^2 - r_1^2 \\
f_5 &:= (x_t - x_2)^2 + y_t^2 - t_2^2 & f_{11} &:= (x_r - x_2)^2 + y_r^2 - r_2^2 \\
f_6 &:= (x_t - x_3)^2 + (y_t - y_3)^2 - t_3^2 & f_{12} &:= (x_r - x_3)^2 + (y_r - y_3)^2 - r_3^2,
\end{aligned} \tag{4.2}$$

where  $r_i$  is the one-way distance from the receiver to the  $i^{\text{th}}$  scatterer,  $t_i$  is the one-way distance from the transmitter to the  $i^{\text{th}}$  scatterer, and  $\rho_i$  is the bistatic distance to the  $i^{\text{th}}$  scatterer. Each of these polynomials is an element of the ring

$$\mathbb{R}[x_r, y_r, x_t, y_t, x_2, x_3, y_3, t_1, t_2, t_3, r_1, r_2, r_3, a, b, c, \rho_1, \rho_2, \rho_3], \tag{4.3}$$

and the affine variety associated with this set of polynomials is the set

$$V(f_1, \dots, f_{12}) = \{(x_r, \dots, \rho_3) : f_i(x_r, \dots, \rho_3) = 0 \forall i\}$$

In other words,  $V(f_1, \dots, f_{12})$  is the set of solutions to the system in (4.2) when we set each  $f_i = 0$ . If we correctly specified the polynomials in our system, then the affine variety will contain all of the potential combinations of sensor positions and target geometries. For our particular inverse problem, we can measure the bistatic ranges,  $\rho_1, \rho_2, \rho_3$ , and we would like to determine geometric information about the target configuration matrix,  $a, b$ , and  $c$ . To find an implicit relationship between only these variables, we would take the system defined in (4.2), and attempt to eliminate variables until only  $\rho_1, \rho_2, \rho_3, a, b$ , and  $c$  remain. As noted in section 3.2, this elimination will result in the ideal generated by 0. With this in mind, we return to our assumption that the transmitter location is fixed relative to the scattering center locations. This implies that the variables  $t_1, t_2$ , and  $t_3$  are all constant, even though they are unknown. If we consider these values as geometric invari-

ants to be estimated, we are interested in finding a polynomial relationship between the variables  $\rho_1, \rho_2, \rho_3, a, b, c, t_1, t_2,$  and  $t_3$ . Following the same elimination procedure, we find

$$G \cap \mathbb{R}[t_1, t_2, t_3, a, b, c, \rho_1, \rho_2, \rho_3] = \langle g_1 \rangle, \quad (4.4)$$

where

$$\begin{aligned}
g_1 := & a^2\rho_1^4 + b^2\rho_2^4 + c^2\rho_3^4 - 4a^2t_1\rho_1^3 - 4b^2t_2\rho_2^3 - 4c^2t_3\rho_3^3 \\
& - (a^2 + b^2 - c^2)\rho_1^2\rho_2^2 - (a^2 - b^2 + c^2)\rho_1^2\rho_3^2 + (a^2 - b^2 - c^2)\rho_2^2\rho_3^2 \\
& + 2(a^2 + b^2 - c^2)t_1\rho_1\rho_2^2 + 2(a^2 - b^2 + c^2)t_1\rho_1\rho_3^2 \\
& + 2(a^2 + b^2 - c^2)t_2\rho_1^2\rho_2 - 2(a^2 - b^2 - c^2)t_2\rho_2\rho_3^2 \\
& + 2(a^2 - b^2 + c^2)t_3\rho_1^2\rho_3 - 2(a^2 - b^2 - c^2)t_3\rho_2^2\rho_3 \\
& + (a^2(a^2 - b^2 - c^2) - (a^2 + b^2 + c^2)t_2^2 - (a^2 - b^2 + c^2)t_3^2 + 6a^2t_1^2)\rho_1^2 \\
& + (b^2(-a^2 + b^2 - c^2) - (a^2 + b^2 - c^2)t_1^2 + (a^2 - b^2 + c^2)t_3^2 + 6b^2t_2^2)\rho_2^2 \\
& + (c^2(-a^2 - b^2 + c^2) - (a^2 - b^2 + c^2)t_1^2 + (a^2 - b^2 - c^2)t_2^2 + 6c^2t_3^2)\rho_3^2 \\
& - 4(a^2 + b^2 - c^2)t_1t_2\rho_1\rho_2 - 4(a^2 - b^2 + c^2)t_1t_3\rho_1\rho_3 + 4(a^2 - b^2 - c^2)t_2t_3\rho_2\rho_3 \\
& + 2(a^2(-a^2 + b^2 + c^2) + (a^2 + b^2 - c^2)t_2^2 + (a^2 - b^2 + c^2)t_3^2 - 2a^2t_1^2)t_1\rho_1 \\
& + 2(b^2(a^2 - b^2 + c^2) + (a^2 + b^2 - c^2)t_1^2 - (a^2 - b^2 - c^2)t_3^2 - 2b^2t_2^2)t_2\rho_2 \\
& + 2(c^2(a^2 + b^2 - c^2) + (a^2 - b^2 + c^2)t_1^2 - (a^2 - b^2 - c^2)t_2^2 - 2c^2t_3^2)t_3\rho_3.
\end{aligned} \quad (4.5)$$

If we included all possible geometric relations in system (4.2), we can conclude that  $g_1$  generates the ideal containing all possible polynomial invariant equations for the bistatic, near-field case in which the transmitter location is fixed. We could solve for the pairwise squared distances,  $a^2, b^2$  and  $c^2$ , given data for  $\rho_1, \rho_2,$  and  $\rho_3$ , as we did in section 3.1.2. This approach will be insufficient for cases with more scatterers ( $N > 3$ ), as we will be forced to generate new polynomials for each case, and the complexity of the polynomial invariants increases with the number of scatterers. Additionally, the polynomial invariants are distinct for the 2D and 3D cases. In the 3D case, it is

unlikely that we will be able to extract pairwise distances between scattering centers without an additional non-linear solve, and so an approach in which we construct a shape representative as we did in section 3.1.2 is undesirable. Instead, we will take the form of this invariant as inspiration and attempt to find a matrix expression analogous to (1.62), the MNF invariant equation described by Ferrara et al. [3].

### 4.1.2 General Invariant Equation

As before, suppose we encode the positions of our scattering centers in the centered configuration matrix,  $S \in \mathbb{R}^{N \times d}$ , where  $N$  is the number of scatterers and  $d$  is the dimension of the target (either 2 or 3). Let  $\mathbf{t} \in \mathbb{R}^N$  be the vector of fixed distances between the transmitter and each of the scattering centers, and let  $\boldsymbol{\rho}_l$  be the vector of bistatic ranges to each of the scatterers measured by the receiver. Model the relative motion of the target to the receiver as before, with time-dependent rotation  $\mathcal{O}_l$  and time-dependent translation  $\boldsymbol{\tau}$ . Then the one-way distance to the  $n^{\text{th}}$  scatterer from the receiver at time  $l$  is

$$\rho_{nl} - t_n = \|\mathbf{s}_n \mathcal{O}_l + \boldsymbol{\tau}_l \mathbf{1}^T\|. \quad (4.6)$$

If we square each entry and concatenate the results, we have

$$(\boldsymbol{\rho}_l - \mathbf{t})^2 = \underbrace{\begin{bmatrix} \|s_1^T\|^2 \\ \vdots \\ \|s_N^T\|^2 \end{bmatrix}}_{\mathbf{m}} + 2S\mathcal{O}_l\boldsymbol{\tau}_l + \|\boldsymbol{\tau}_l\|^2\mathbf{1}, \quad (4.7)$$

where  $(\boldsymbol{\rho}_l - \mathbf{t})^2 = (\boldsymbol{\rho}_l - \mathbf{t}) \odot (\boldsymbol{\rho}_l - \mathbf{t})$  denotes the entry-wise squaring operation, and  $\odot$  is the entry-wise product. As we did in section 1.3.4, we can apply the centering matrix,

$$C = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T, \quad (4.8)$$

and expand the squared quantity to find

$$C(\boldsymbol{\rho}_l^2 - 2\mathbf{t} \odot \boldsymbol{\rho}_l + \mathbf{t}^2 - \mathbf{m}) = 2S\mathcal{O}_l\boldsymbol{\tau}_l. \quad (4.9)$$

Now, we can take the pulse-wise difference,

$$C(\boldsymbol{\rho}_l^2 - \boldsymbol{\rho}_k^2) - 2C\mathbf{t} \odot (\boldsymbol{\rho}_l - \boldsymbol{\rho}_k) = 2S(\mathcal{O}_l\boldsymbol{\tau}_l - \mathcal{O}_k\boldsymbol{\tau}_k). \quad (4.10)$$

The product  $\mathbf{t} \odot (\boldsymbol{\rho}_l - \boldsymbol{\rho}_k) = T(\boldsymbol{\rho}_l - \boldsymbol{\rho}_k)$ , where

$$T = \begin{bmatrix} t_1 & 0 & 0 \\ 0 & t_2 & 0 \\ 0 & 0 & t_3 \end{bmatrix}, \quad (4.11)$$

so that, if we can determine the matrix  $T$ , we can determine the affine invariants of  $S$  with the relation

$$C(\boldsymbol{\rho}_l^2 - \boldsymbol{\rho}_k^2) - 2CT(\boldsymbol{\rho}_l - \boldsymbol{\rho}_k) = 2S(\mathcal{O}_l\boldsymbol{\tau}_l - \mathcal{O}_k\boldsymbol{\tau}_k). \quad (4.12)$$

Returning to (4.9), we can multiply by  $S^T$  on both sides,

$$S^TC(\boldsymbol{\rho}_l^2 - 2T\boldsymbol{\rho}_l + \mathbf{t}^2 - \mathbf{m}) = 2S^TS\mathcal{O}_l\boldsymbol{\tau}_l. \quad (4.13)$$

Since  $S$  is rank  $d$ , the matrix  $S^TS \in \mathbb{R}^{d \times d}$  is full rank, and we can invert the matrix to find

$$(S^TS)^{-1}S^TC(\boldsymbol{\rho}_l^2 - 2T\boldsymbol{\rho}_l + \mathbf{t}^2 - \mathbf{m}) = 2\mathcal{O}_l\boldsymbol{\tau}_l. \quad (4.14)$$

It follows that

$$\begin{aligned} 4\|\boldsymbol{\tau}_l\|^2 &= (\boldsymbol{\rho}_l^2 - 2T\boldsymbol{\rho}_l + \mathbf{t}^2 - \mathbf{m})^T C^T \underbrace{S(S^TS)^{-2}S^T}_{\Omega} C(\boldsymbol{\rho}_l^2 - 2T\boldsymbol{\rho}_l + \mathbf{t}^2 - \mathbf{m}) \\ &= (\boldsymbol{\rho}_l^2)^T \Omega \boldsymbol{\rho}_l^2 - 4\boldsymbol{\rho}_l^T T \Omega \boldsymbol{\rho}_l^2 + 2(\boldsymbol{\rho}_l^2)^T \Omega (\mathbf{t}^2 - \mathbf{m}) + 4\boldsymbol{\rho}_l^T T \Omega T \boldsymbol{\rho}_l \\ &\quad - 4\boldsymbol{\rho}_l^T \Omega (\mathbf{t}^2 - \mathbf{m}) + (\mathbf{t}^2 - \mathbf{m})^T \Omega (\mathbf{t}^2 - \mathbf{m}), \end{aligned} \quad (4.15)$$

and we could take the pulse-wise difference to find

$$\begin{aligned}
4(\|\boldsymbol{\tau}_l\|^2 - \|\boldsymbol{\tau}_k\|^2) &= (\boldsymbol{\rho}_l^2 - \boldsymbol{\rho}_k^2)^T \Omega (\boldsymbol{\rho}_l^2 - \boldsymbol{\rho}_k^2) - 4(\boldsymbol{\rho}_l - \boldsymbol{\rho}_k)^T T \Omega (\boldsymbol{\rho}_l^2 - \boldsymbol{\rho}_k^2) \\
&\quad + 2(\boldsymbol{\rho}_l^2 - \boldsymbol{\rho}_k^2)^T \Omega (\mathbf{t}^2 - \mathbf{m}) + 4(\boldsymbol{\rho}_l - \boldsymbol{\rho}_k)^T T \Omega T (\boldsymbol{\rho}_l - \boldsymbol{\rho}_k) \\
&\quad - 4(\boldsymbol{\rho}_l - \boldsymbol{\rho}_k)^T \Omega (\mathbf{t}^2 - \mathbf{m}).
\end{aligned} \tag{4.16}$$

Additionally, we could substitute the relation

$$N\|\boldsymbol{\tau}_l\|^2 = \mathbf{1}^T \boldsymbol{\rho}_l^2 - 2\mathbf{1}^T T \boldsymbol{\rho}_l + \mathbf{1}^T (\mathbf{t}^2 - \mathbf{m}) \tag{4.17}$$

to express (4.15) as

$$\begin{aligned}
0 &= (\boldsymbol{\rho}_l^2 - 2T\boldsymbol{\rho}_l + \mathbf{t}^2 + \mathbf{m})^T \Omega (\boldsymbol{\rho}_l^2 - 2T\boldsymbol{\rho}_l + \mathbf{t}^2 + \mathbf{m}) \\
&\quad - \frac{4}{N} \mathbf{1}^T (\boldsymbol{\rho}_l^2 - 2T\boldsymbol{\rho}_l + \mathbf{t}^2 + \mathbf{m}).
\end{aligned} \tag{4.18}$$

Given the one-way distances to the transmitter,  $\mathbf{t}$ , (4.18) is the geometric invariant equation relating the range measurements to the configuration of scatterers. With this information, we can now develop a scheme for estimating the target configuration,  $S$ , by first estimating  $\mathbf{t}$ .

### 4.1.3 Computing Affine Invariants

Given the vector of one-way distances between the transmitter and each of the  $N$  scattering centers, (4.12) will allow us to compute the range of  $S$ . To see this, first rewrite (4.12) as

$$C(R^2 - 2TR)D = 2SYD, \tag{4.19}$$

where  $R \in \mathbb{R}^{N \times L}$  is the matrix of bistatic ranges,

$$R = \begin{bmatrix} \boldsymbol{\rho}_1 & \boldsymbol{\rho}_2 & \dots & \boldsymbol{\rho}_L \end{bmatrix}, \tag{4.20}$$

$R^2 = R \odot R$  is the matrix of squared ranges,

$$Y = \begin{bmatrix} \mathcal{O}_1 \tau_1 & \mathcal{O}_2 \tau_2 & \dots & \mathcal{O}_L \tau_L \end{bmatrix}, \quad (4.21)$$

and right multiplication by  $D$  corresponds to taking pairwise differences of column vectors. From (4.19), we know that each column of  $C(R^2 - 2TR)D$  is an element of  $\text{range}(S)$ . We can then compute the SVD of  $C(R^2 - 2TR)D = V\Sigma U^T$ , and the columns of  $V$  will be an orthonormal basis for  $\text{range}(S)$ . Unfortunately, the diagonal matrix  $T$  is unknown, and so we must first estimate the entries of  $T$  prior to the affine invariant estimation step.

For convenience, we will think of the matrix  $C(R^2 - 2TR)D$  as parameterized by the diagonal matrix  $T$ , and we will call the true matrix of one-way distances  $T^*$ . To estimate  $T^*$ , first notice that  $\text{rank}(C(R^2 - 2T^*R)D) = \text{rank}(S) = d$ , where  $d$  is the dimension of the target (usually  $d = 3$ ). In general, both  $R^2$  and  $R$  are rank- $N$ , so that generic choices of  $T$  will not satisfy this property. The matrix  $C = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$  projects out the one-dimensional subspace spanned by the vector  $\mathbf{1}$ , and so multiplication by  $C$  decreases the rank of a matrix by at most 1. Similarly, the null space of  $D^T$  is spanned by  $\mathbf{1}$ , so that if a matrix  $A$  is rank  $N$ , then  $\text{rank}(CAD) \geq N - 1$ . It follows that  $\text{rank}(R^2 - 2T^*R)$  is either 3 or 4. Since we suspect that  $T^*$  is the only matrix to satisfy such a low-rank condition, we attempt to find  $T^*$  as the solution to the minimization problem,

$$\begin{aligned} & \text{minimize} \quad \text{rank}(R^2 - 2TR) \\ & \text{subject to} \quad t_i > 0 \quad \forall i. \end{aligned} \quad (4.22)$$

As stated, however, this problem is impractical. The problem of minimizing the rank of a matrix is NP-hard [52], since  $\text{rank}(\cdot)$  is a discontinuous, non-convex function of the entries of a matrix. In practice, it is common to minimize a new objective function, which is a convex relaxation of  $\text{rank}(\cdot)$ . For a positive semi-definite matrix  $A$ , it is common to minimize the nuclear norm,

$$\|A\|_* := \text{tr}(\sqrt{A^T A}) = \sum_{i=1} \sigma_i(A)$$

in place of  $\text{rank}(A)$ , where  $\text{tr}(\cdot)$  is the trace operation, and  $\sigma_i(A)$  is the  $i^{\text{th}}$  singular value of  $A$ . In some situations, the minimum of the nuclear norm objective function is identical to the minimum-rank solution [53–55]. Since  $\text{rank}(A) = \text{rank}(A^T A) = \text{rank}(AA^T)$  and  $AA^T$  is positive semi-definite for any choice of  $A$ , we can approximate the solution to our original problem with

$$\begin{aligned} & \text{minimize} && f(T) := \text{tr}((R^2 - 2TR)(R^2 - 2TR)^T) \\ & \text{subject to} && t_i > 0 \quad \forall i. \end{aligned} \tag{4.23}$$

Since this new objective function is convex and differentiable, we can solve the minimization problem exactly. First, we simplify the objective function by rewriting it in terms of vectors. Let  $\boldsymbol{\delta}(A)$  denote the diagonal of  $A$ , reshaped as a vector, and recall that  $\mathbf{t} = \boldsymbol{\delta}(T)$ . Then

$$\begin{aligned} f(T) &= \text{tr}(R^2(R^2)^T - 2TR(R^2)^T - 2R^2R^TT + 4TRR^TT) \\ &= \boldsymbol{\delta}(R^2(R^2)^T)^T \mathbf{1} - 4\boldsymbol{\delta}(R^2R^T)^T \mathbf{t} + 4\boldsymbol{\delta}(RR^T)^T (\mathbf{t} \odot \mathbf{t}). \end{aligned} \tag{4.24}$$

The gradient is then

$$\nabla_{\mathbf{t}} f = -4\boldsymbol{\delta}(R^2R^T) + 8\boldsymbol{\delta}(RR^T) \odot \mathbf{t}, \tag{4.25}$$

and so if we let  $\mathbf{a} = \boldsymbol{\delta}(R^2R^T)$  and  $\mathbf{b} = \boldsymbol{\delta}(RR^T)$ , we have an extremely simple, closed-form solution to (4.23) given by

$$t_i = \frac{a_i}{2b_i}. \tag{4.26}$$

While the trace is a useful heuristic in many applications, we find that the solution to (4.23) is not a sufficiently accurate approximation to the solution of the original problem in (4.22). As inspiration for a new objective function, we note that  $\text{tr}(A^T(AA^T)^{-1}A) = r$  for a matrix  $A \in \mathbb{R}^{r \times M}$  with  $\text{rank}(A) = r$ . The matrix  $AA^T$  is not invertible if  $A$  is not full-rank, and so we cannot use this function directly as a proxy for  $\text{rank}(\cdot)$ . Instead, we follow the analysis by Zhao [56] and define

$$\phi_\epsilon(A) := \text{tr}(A^T(AA^T + \epsilon I)^{-1}A). \tag{4.27}$$

Adding the term  $\epsilon I$  means that  $AA^T + \epsilon I$  is full rank for  $\epsilon \neq 0$ , so that  $(AA^T + \epsilon I)^{-1}$  exists. Further,  $\phi_\epsilon(A)$  has the desirable property that  $\lim_{\epsilon \rightarrow 0} \phi_\epsilon(A) = \text{rank}(A)$ . To see this, suppose  $A$  has the SVD  $A = U\Sigma V^T$ . Then

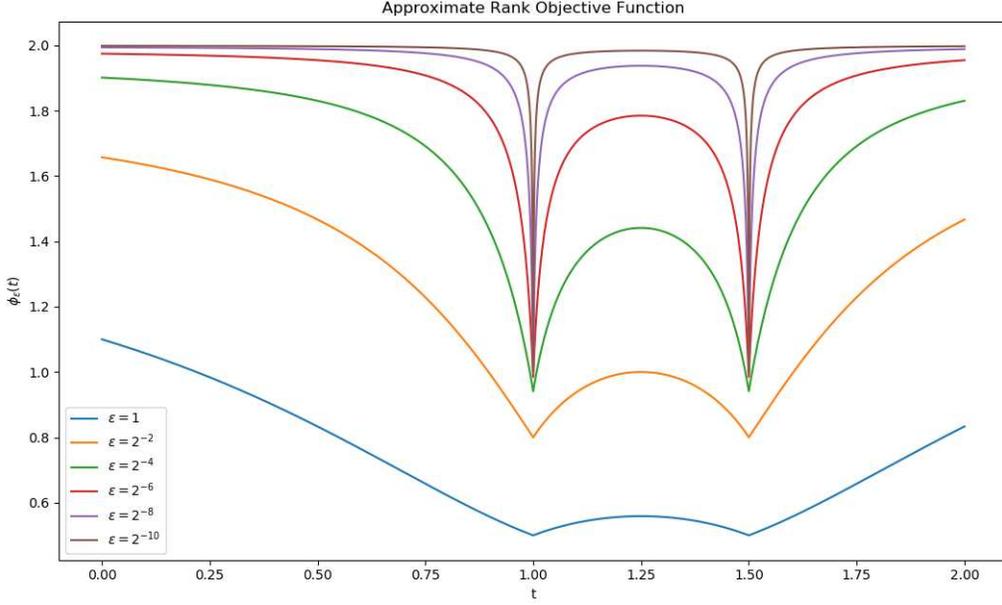
$$\begin{aligned}
\phi_\epsilon(A) &= \text{tr}(A^T(AA^T + \epsilon I)^{-1}A) \\
&= \text{tr}(V\Sigma U^T(U\Sigma^2 U^T + \epsilon I)^{-1}U\Sigma V^T) \\
&= \text{tr}(U^T(U^T(\Sigma^2 + \epsilon I)U)^{-1}U\Sigma) \\
&= \text{tr}((\Sigma^2 + \epsilon I)^{-1}\Sigma^2) \\
&= \sum_{k=1}^r \frac{\sigma_k^2(A)}{\sigma_k^2(A) + \epsilon}.
\end{aligned} \tag{4.28}$$

For a fixed  $\epsilon > 0$ , this objective function is still non-convex, but it is a differentiable function of the entries in  $A$ . This suggests a graduated optimization problem, in which we alternately reduce  $\epsilon$  and minimize  $\phi_\epsilon(A)$  with a local minimization technique. To illustrate the convergence of  $\phi_\epsilon(A)$  to  $\text{rank}(A)$ , we have included a plot of  $\phi_\epsilon(A)$  for various values of  $\epsilon$  in Figure 4.1. Zhao did not take a graduated minimization approach in his paper; he chose instead to reformulate the problem as a bi-level, semi-definite programming problem [56]. Because we are interested in minimizing the rank of

$$Z(T) = R^2 - 2TR, \tag{4.29}$$

where  $T$  is diagonal, our problem is reduced in complexity to a global minimization problem over  $\mathbb{R}^N$ , where  $T$  is  $N \times N$ . To make use of local minimization techniques, we require the gradient, which we can compute one partial at a time. Let  $A(T) = Z(T)Z(T)^T$ , then

$$\begin{aligned}
\partial_{t_i} \phi_\epsilon(A) &= \text{tr}(\partial_{t_i}[(A + \epsilon I)^{-1}A]) \\
&= \text{tr}(\partial_{t_i}[(A + \epsilon I)^{-1}]A + (A + \epsilon I)^{-1}\partial_{t_i}[A]) \\
&= \text{tr}(-(A + \epsilon I)^{-1}\partial_{t_i}[A](A + \epsilon I)^{-1}A + (A + \epsilon I)^{-1}\partial_{t_i}[A]) \\
&= \text{tr}((A + \epsilon I)^{-1}\partial_{t_i}[A](I - (A + \epsilon I)^{-1}A)),
\end{aligned} \tag{4.30}$$



**Figure 4.1:** A plot of  $\phi_\epsilon(A)$ , where  $A(t)$  is  $2 \times 2$  diagonal matrix with diagonal  $[1, (t - 1)(t - 1.5)]$ . For  $t = 1$  and  $t = 1.5$ ,  $\text{rank}(A(t)) = 1$ . Otherwise,  $\text{rank}(A(t)) = 2$ . We can see that as  $\epsilon \rightarrow 0$ ,  $\phi_\epsilon$  more closely approximates rank. In practice, greater values of  $\epsilon$  provide a smoothing effect to the objective, which decreases the chance that local minimization techniques will converge before reaching the global minimum.

where

$$\begin{aligned}
\partial_{t_i} A &= \partial_{t_i} [(R^2 - 2TR)(R^2 - 2TR)^T] \\
&= \partial_{t_i} [R^2(R^2)^T - 2TR(R^2)^T - 2R^2R^T T + 4TRR^T T] \\
&= -2E_i R(R^2)^T - 2R^2 R^T E_i + 4(E_i R R^T T + T R R^T E_i).
\end{aligned} \tag{4.31}$$

Here, we've used  $E_i$  to represent the diagonal matrix with 1 in the  $i^{\text{th}}$  entry of the diagonal and zeros elsewhere. The gradient is sufficient for many local minimization routines, but the Hessian of  $\phi_\epsilon(A)$  is not too difficult to compute, and gives us access to a wider range of minimization algorithms. For simplicity, let  $M = (A + \epsilon I)^{-1}$ . The Hessian is given entry-wise by

$$\begin{aligned}
\partial_{t_j} \partial_{t_i} \phi_\epsilon(A) &= \partial_{t_j} \text{tr} (M \partial_{t_i} [A] (I - MA)) \\
&= \text{tr} (M (\partial_{t_j} \partial_{t_i} [A] - 2 \partial_{t_i} [A] M \partial_{t_j} [A]) (I - MA)),
\end{aligned} \tag{4.32}$$

where

$$\begin{aligned}
\partial_{t_j} \partial_{t_i} [A] &= \partial_{t_j} [-2E_i R (R^2)^T - 2R^2 R^T E_i + 4(E_i R R^T T + T R R^T E_i)] \\
&= 4E_j R R^T E_i + 4E_i R R^T E_j.
\end{aligned} \tag{4.33}$$

Pseudocode describing the graduated optimization problem for estimating  $T^*$  is included in Algorithm 1. Any local minimization procedure can be used in place of FINDLOCALMIN. In Section 4.2, we include accuracy results for estimating  $T^*$  using both an implementation of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm, as well as an implementation of Newton's method.

---

**Algorithm 1** Estimate the one-way distances from target to transmitter,  $T^*$ .

---

```

function ESTIMATE $T(R, numIters)$ 
   $T_0 \leftarrow \text{MINNUCLEARNORM}(R)$ 
   $\epsilon \leftarrow 1.0$ 
  for  $k = 0 : numIters$  do
     $\epsilon \leftarrow 2^{-k} \epsilon$ 
     $T_k \leftarrow \text{FINDLOCALMIN}(\nabla_T \phi_\epsilon, T_{k-1})$ 
  end for
  return  $T_k$ 
end function

```

```

function MINNUCLEARNORM( $R$ )
   $\mathbf{a} \leftarrow \delta(R^2 R^T)$ 
   $\mathbf{b} \leftarrow \delta(R R^T)$ 
  for  $i = 1 : N$  do
     $T[i, i] \leftarrow 2a[i]b[i]^{-1}$ 
  end for
end function

```

---

Once we have computed  $T$ , our estimate of  $T^*$ , we compute the affine invariants of  $S$  according to (4.19) by finding the singular value decomposition of the left hand side,

$$V\Sigma U = C(R^2 - 2TR)D. \tag{4.34}$$

As in Section 1.3.1, we have found an orthonormal basis for the columns of  $S$ .

#### 4.1.4 Computing Euclidean Invariants

Given that we have reliable estimates of the affine invariants,  $V$ , and the one-way distances from the scattering centers to the transmitter,  $T$ , it is possible to estimate the affine invariants of  $S$  according to the usual process in Section 1.3.2. Reorganize the terms of (4.18),

$$\begin{aligned} \frac{4}{N} \mathbf{1}^T \underbrace{(\boldsymbol{\rho}_l^2 - 2T\boldsymbol{\rho} + \mathbf{t}^2)}_{\mathbf{r}_l^2} &= (\boldsymbol{\rho}_l^2 - 2T\boldsymbol{\rho} + \mathbf{t}^2 - \mathbf{m})^T \Omega (\boldsymbol{\rho}_l^2 - 2T\boldsymbol{\rho} + \mathbf{t}^2 - \mathbf{m}) + \frac{4}{N} \mathbf{1}^T \mathbf{m} \\ \frac{4}{N} \mathbf{1}^T \mathbf{r}_l^2 &= (\mathbf{r}_l^2 - \mathbf{m})^T \Omega (\mathbf{r}_l^2 - \mathbf{m}) + \frac{4}{N} \mathbf{1}^T \mathbf{m}, \end{aligned} \quad (4.35)$$

where  $\mathbf{r}_l^2 = \boldsymbol{\rho}_l^2 - 2T\boldsymbol{\rho} + \mathbf{t}^2$  is known for each pulse, and define  $\mathbf{b}_l = V^T \mathbf{r}_l^2$ . We have now reduced the passive, bistatic, near-field case to the monostatic, near-field case, for which we have described the solution in Section 1.3.4.

The approach described in Section 1.3.4 constructs a non-linear least-squares (NLLS) problem whose solution describes the Euclidean invariants of the target, but then linearizes this problem by introducing extra variables in 1.66. It is possible to solve the NLLS problem without introducing extra variables. Starting from the reduced invariant equation,

$$\frac{4}{N} \mathbf{1}^T \mathbf{r}_l^2 = (\mathbf{b}_l - V^T \mathbf{m})^T W^{-1} (\mathbf{b}_l - V^T \mathbf{m}) + \frac{4}{N} \mathbf{1}^T \mathbf{m}, \quad (4.36)$$

where  $W^{-1} = V^T \Omega V = \mathcal{O} \Sigma^{-2} \mathcal{O}^T$  is a  $3 \times 3$  symmetric matrix, as in Section 1.3.4. Note that

$$\mathbf{m} = \begin{bmatrix} \|s_1^T\|^2 \\ \vdots \\ \|s_N^T\|^2 \end{bmatrix} = \text{diag}(S^T S) = \text{diag}(VWV^T), \quad (4.37)$$

so that we can express  $\mathbf{m}$  in terms of the parameters in  $W$ .  $W$  has exactly 6 free parameters, so that our NLLS problem is a global optimization problem over  $\mathbb{R}^6$ . Let

$$W := \begin{bmatrix} w_1 & w_4 & w_5 \\ w_4 & w_2 & w_6 \\ w_5 & w_6 & w_3 \end{bmatrix}, \quad (4.38)$$

and we can expand  $\bar{\mathbf{m}}$  as

$$\bar{\mathbf{m}} = \underbrace{\begin{bmatrix} \mathbf{v}_1^2 & \mathbf{v}_2^2 & \mathbf{v}_3^2 & 2\mathbf{v}_1 \odot \mathbf{v}_2 & 2\mathbf{v}_1 \odot \mathbf{v}_3 & 2\mathbf{v}_2 \odot \mathbf{v}_3 \end{bmatrix}}_A \mathbf{w}, \quad (4.39)$$

where  $\mathbf{w} := \begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 & w_6 \end{bmatrix}^T$ , and  $\mathbf{v}_i$  is column  $i$  of  $V$ , the matrix of affine invariants. We can now express (4.36) in terms of the parameters in  $W$ ,

$$\frac{4}{N} \mathbf{1}^T \mathbf{r}_l^2 = (\mathbf{b}_l - V^T A \mathbf{w})^T W^{-1} (\mathbf{b}_l - V^T A \mathbf{w}) + \frac{4}{N} \mathbf{1}^T A \mathbf{w}, \quad (4.40)$$

To solve the NLLS problem directly, we minimize the objective

$$f(\mathbf{w}) := \frac{1}{2} \|\mathbf{r}(\mathbf{w})\|^2, \quad (4.41)$$

where  $\mathbf{r}(\mathbf{w})$  is the residual, defined element-wise by

$$r_l(\mathbf{w}) := (\mathbf{b}_l - V^T A \mathbf{w})^T W^{-1} (\mathbf{b}_l - V^T A \mathbf{w}) + \frac{4}{N} \mathbf{1}^T A \mathbf{w} - \frac{4}{N} \mathbf{1}^T \mathbf{r}_l^2. \quad (4.42)$$

We will also require the Jacobian of  $\mathbf{r}$ ,

$$\begin{aligned} \partial_{w_i} r_l(\mathbf{x}) &= -2(V^T A \mathbf{e}_i)^T W^{-1} (\mathbf{b}_l - V^T A \mathbf{w}) \\ &\quad + (\mathbf{b}_l - V^T A \mathbf{w})^T \partial_{w_i} (W^{-1}) (\mathbf{b}_l - V^T A \mathbf{w}) \\ &\quad + \frac{4}{N} \mathbf{1}^T A \mathbf{e}_i, \end{aligned} \quad (4.43)$$

where  $\mathbf{e}_i$  is the  $i^{\text{th}}$  standard basis vector and

$$\partial_{w_i}(W^{-1}) = -W^{-1}\partial_{w_i}(W)W^{-1}. \quad (4.44)$$

To use a second-order method, we will also require the Hessian of  $f$ ,

$$Hf(\mathbf{w}) = \sum_{l=1}^L r_l(\mathbf{w}) \nabla^2 \mathbf{r}(\mathbf{w}), \quad (4.45)$$

where  $\nabla^2 \mathbf{r}(\mathbf{w})$  is given entry-wise by

$$\begin{aligned} \partial_{w_i} \partial_{w_j} r_l(\mathbf{w}) = & -2(V^T \mathbf{A} \mathbf{e}_i)^T W^{-1} (\mathbf{b}_l - V^T \mathbf{A} \mathbf{w}) \\ & -2(V^T \mathbf{A} \mathbf{e}_j)^T W^{-1} (\mathbf{b}_l - V^T \mathbf{A} \mathbf{w}) \\ & + (\mathbf{b}_l - V^T \mathbf{A} \mathbf{w})^T \partial_{w_i} \partial_{w_j} (W^{-1}) (\mathbf{b}_l - V^T \mathbf{A} \mathbf{w}) \\ & + 2(V^T \mathbf{A} \mathbf{e}_j)^T W^{-1} (V^T \mathbf{A} \mathbf{e}_i), \end{aligned} \quad (4.46)$$

where

$$\partial_{w_i} \partial_{w_j} (W^{-1}) = W^{-1} (\partial_{w_i} (W) W^{-1} \partial_{w_j} (W) + \partial_{w_j} (W) W^{-1} \partial_{w_i} (W)) W^{-1}. \quad (4.47)$$

Given an initial value for  $\mathbf{w}$ , we then minimize  $f$  with any iterative minimization technique. For this work, we use a fast NLLS solver supplied by Rutherford Appleton Laboratories [57, 58]. In Section 4.2.2, we discuss the accuracy of the shape representative constructed using this approach, as compared to the usual linear least-squares approach.

## 4.2 Numerical Results

In Section 4.1.3, we presented a novel approach for estimating the affine invariants of a target given data collected in in the passive, bistatic, near-field sensor modality. Following that, in Section 4.1.4, we presented the accompanying procedure for estimating the Euclidean invariants of the target. Now, we will investigate the effectiveness of those algorithms with numerical testing.

## 4.2.1 Computing Affine Invariants

As a first step, we generate synthetic range data on which to test Algorithm 1. For each trial, we construct a shape matrix,  $S$ , with  $N = 10$  scatterers by randomly selecting the 3D coordinates of each scatterer, and then translating the shape so that its centroid coincides with the origin of our coordinate system. We then randomly select the transmitter position coordinates,  $\mathbf{x}_t$ , along with a set of  $L = 300$  receiver position coordinates,  $\mathbf{x}_r(l)$ .  $L$  corresponds to the collective number of pulses measured by all receivers. At one extreme, we could have a single moving receiver take all 300 measurements, and at the other extreme would be an array of 300 receivers each taking a single measurement. For each receiver coordinate, the passive, BNF distance to a scatterer is simply the distance from the transmitter to scatter, plus the distance from the scatterer to the receiver for each  $l = 1, \dots, L$ ,

$$\rho_{nl} = \|\mathbf{s}_n - \mathbf{x}_t\| + \|\mathbf{s}_n - \mathbf{x}_r(l)\|. \quad (4.48)$$

Using this to construct the  $N \times L$  matrix of range measurements,  $R$ , we have synthetic range measurements to a target with  $N$  scatterers. In the `julia` script, `estimateTrial.jl`, we have implemented Algorithm 1 using both Newton's method and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm and listed the convergence results in Table 4.1. Also included in this table are the results of the algorithm when various levels of Gaussian noise are added to the range measurements prior to estimation of the invariant. For these trials, the amplitude of the noise is described in terms of the variance in the range measurements, and is tuned with the multiplicative factor  $\alpha$  according to

$$amp_{noise} = \alpha^{-1} \sigma^2(R).$$

Let  $T^*$  be the diagonal matrix of exact one-way measurements from transmitter to scatterers, and let  $V^*$  be an orthonormal basis for the columns of  $S$ . To quantify the error in the invariant estimates, we measure the error in the one-way distances with the usual Euclidean two norm on  $\mathbb{R}^N$ ,  $\|T - T^*\|_2^2 = \sum_i^N ((t_{ii} - t_{ii}^*)^2)$ . The matrices  $V$  and  $V^*$  both represent subspaces of dimension 3 in  $\mathbb{R}^N$ , and so are members of the Grassmannian  $G(3, N)$ , which is the set of 3-dimensional subspaces

of an  $N$ -dimensional vector space. To measure the error in  $V$ , we use the geodesic distance on  $G(3, N)$ . If  $\theta_1, \theta_2, \theta_3$  are the principal angles between the subspaces defined by  $V$  and  $V^*$ , then the geodesic distance is

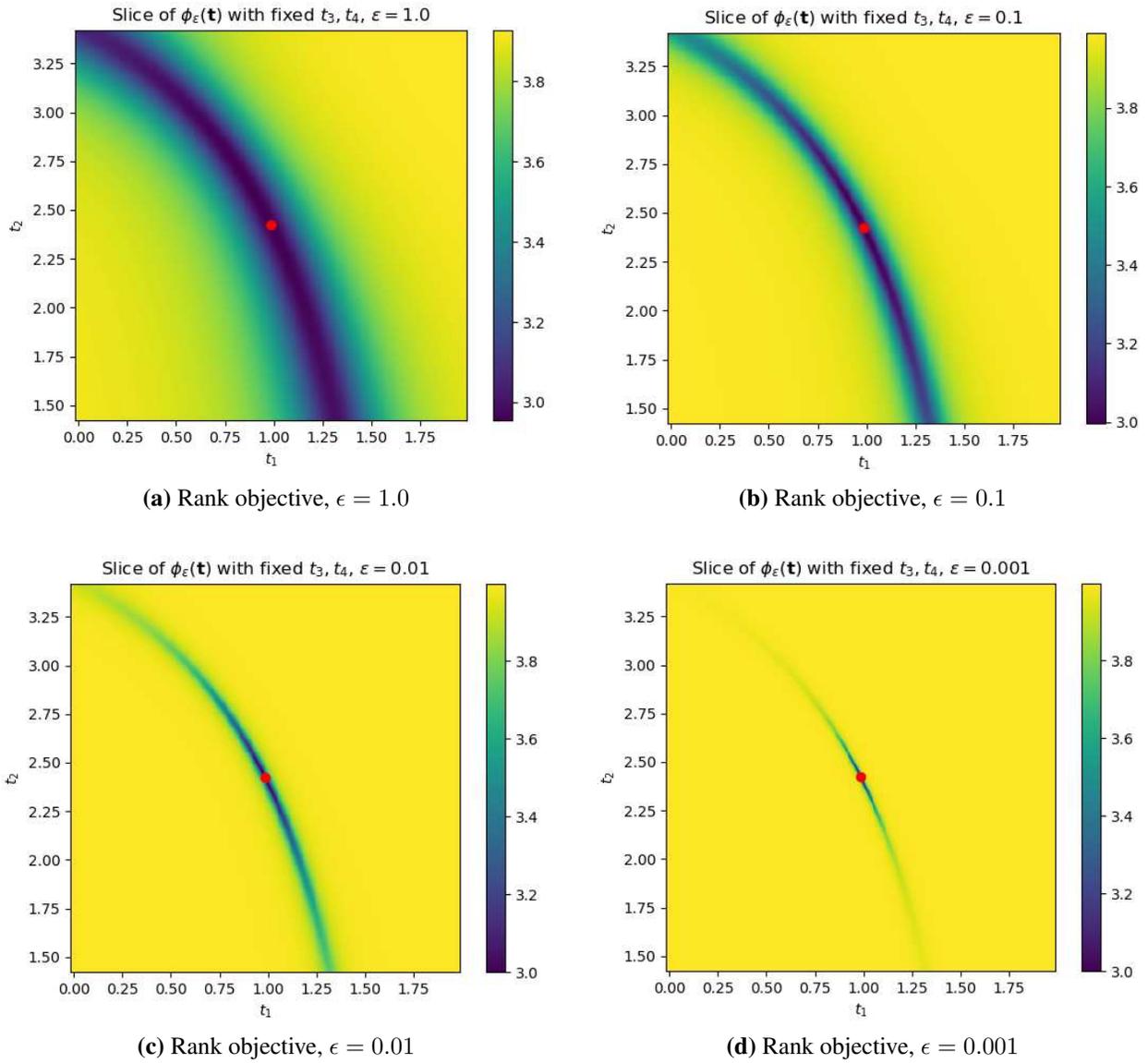
$$d_{geo}(V, V^*) := \sqrt{\theta_1^2 + \theta_2^2 + \theta_3^2}. \quad (4.49)$$

We can see in Table 4.1 that the quality of invariants estimated by the current technique degrades quickly as the noise amplitude increases. This is because the additive matrix of Gaussian white noise,  $E$ , is almost always full-rank, so that the number of significant eigenvalues of  $A(T) + E$  increases with the noise amplitude. It follows that the global minimum of  $\phi_\epsilon(A(T) + E)$  will not necessarily coincide with  $T^*$ . Modifications to this technique would account for noisy range measurements, and could perform a joint minimization over the space of possible  $T$  and  $E$ . To derive an effective approach for use with a real system, one would need to identify an accurate model for the noise present in the range measurements.

## 4.2.2 Computing Euclidean Invariants

Given accurate estimates of  $T$ , we now investigate the numerical performance of the Euclidean invariant estimation processes. For the following trials, we generated synthetic data as in Section 4.2.1, estimated the affine invariants,  $V$ , and then applied either the linear least-squares (LLS) approach described in Section 1.3.4, or the full non-linear least-squares (NLLS) approach described in Section 4.1.4. The NLLS implementation leverages a fast, `Fortran`-based NLLS solver, `RALfit` [57, 58], for which we have written a `julia` interface. In each trial, we solve for  $w$  using the LLS approach, and then refine the estimate by passing it as the starting point to the NLLS method. Table 4.2 summarizes the results. To quantify the error in our Euclidean invariants, we compute a shape representative,  $S$ , as in Section 1.3.2 and compare it to the true target shape,  $S^*$ . To account for arbitrary rotations and reflections on the shape estimate, we first solve the orthogonal Procruste’s problem for  $S$  and  $S^*$ , and then take the Frobenius norm of the result. The final error measurement is

$$d_{pro}(S, S^*) = \min_{\mathcal{O}} \|S\mathcal{O} - S^*\|_F, \quad (4.50)$$



**Figure 4.2:** These plots illustrate the values of the approximate rank objective function,  $\phi_\epsilon(A(T))$ , where  $N = 4$ . We vary  $t_1, t_2$  and fix the remaining  $t_i$ . The optimal value for  $(t_1, t_2)$  is indicated in red. As  $\epsilon$  decreases from top-left to bottom-right, we can see that the blue region shrinks around the global minimum.

**Table 4.1:** This table summarizes the mean performance of Algorithm 1 over 50 trials, using two different local minimization techniques. Error in  $T$  is measured with the usual two norm on  $\mathbf{R}^{10}$ , and error in the subspace estimate,  $V$ , is measured with the geodesic distance on  $G(3, 10)$ , defined in (4.49). For each set of trials, we used an initial value of  $\epsilon = 1.0$  and set  $numIters = 4$ . We should note that the initial choice of  $\epsilon$ , as well as the number of iterations performed, can influence the accuracy of this technique. Further, it is not necessary to reduce  $\epsilon$  by a factor of two in every iteration; smaller reductions in  $\epsilon$  are beneficial, but increase the number of iterations required.

Local Min. Technique	Number Converged	Mean Error in $T$	Mean Error in $V$
BFGS	49	1.67258e-7	2.68492e-8
Newton	43	9.48078e-8	2.32872e-8
$\alpha = 100$			
BFGS	45	0.023001	0.0022071
Newton	35	0.024127	0.0021673
$\alpha = 50$			
BFGS	48	0.084055	0.0052336
Newton	30	0.127339	0.0103484
$\alpha = 25$			
BFGS	44	0.297952	0.0132395
Newton	43	0.290028	0.0132577
$\alpha = 10$			
BFGS	45	2.892260	0.1251810
Newton	50	3.393770	0.1991010

where  $\|\cdot\|_F$  denotes the Frobenius norm and  $\mathcal{O} \in O(3)$  is an arbitrary rotation or reflection. We can see that both the LLS and NLLS solution are nearly identical, so that the reduced computational complexity of the LLS method makes it the preferred method for estimating Euclidean invariants.

### 4.3 Further Work

The bistatic results in this chapter are limited in two primary respects. First, the invariant equation (4.18) only applies for bistatic scenarios in which either the receiver or the transmitter is stationary. Second, the affine invariant estimation step for the passive, near-field case is not well-adapted to noisy range measurements. The second concern could be addressed by studying

**Table 4.2:** This table summarizes the mean performance of our Euclidean invariant estimation algorithms over 50 trials, with  $N = 10$ ,  $L = 300$ . Error in the Euclidean invariants is measured according to (4.50). From this side-by-side comparison, we can see that both methods are robust to white noise on the range tracks, but that the Euclidean invariant estimates generated by each technique are nearly identical.

Solution Type	No Noise	$\alpha = 100$	$\alpha = 50$	$\alpha = 25$	$\alpha = 1$
LLS	1.34443e-15	3.35177e-4	7.1002e-4	1.37775e-3	5.30153e-2
NLLS	1.45124e-15	3.35177e-4	7.1002e-4	1.37775e-3	5.41893e-2

the nature of the errors in range tracks provided to the algorithm, and then compensating for those errors by adding extra parameters to the minimization problem. This would increase the cost of invariant estimation, but could improve the robustness of the method to measurement error. The first limitation is the more difficult to address. To solve this problem, one would need to finish the derivation started in Chapter 3 for the bistatic, near-field invariant equation. This could involve advancing the state-of-the-art in elimination of variables from polynomial systems, or perhaps our particular system has some structure that would simplify the elimination process. It is also possible that the problem could be solved with more computing power, but it is difficult assess the necessary scope of those computing resources. While difficult, addressing the first concern would eliminate the second concern, as the bistatic, near-field invariant equation would also apply to passive, bistatic range data. This would be very exciting, as this discovery would complete our knowledge of geometric invariant equations for all sensor configurations and geometries, and theoretically enable the existence of a new class of geolocation algorithms.

# Bibliography

- [1] M. A. Stuff, R. C. Sullivan, Jr., B. J. Thelen, and S. A. Werness, “Automated two- and three-dimensional, fine-resolution radar imaging of rigid targets with arbitrary unknown motion,” 1994.
- [2] M. Cheney and B. Borden, *Fundamentals of Radar Imaging*. Society for Industrial and Applied Mathematics, 2009.
- [3] M. Ferrara and G. Arnold, “Shape and motion estimation from near-field echo-based sensor data,” *SIAM Journal of Imaging Sciences*, vol. 2, no. 2, 2009.
- [4] J. R. Fienup, “Detecting moving targets in SAR imagery by focusing,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, pp. 794–809, July 2001.
- [5] C. Baker, Jr, “Moving target indicator radar,” Oct. 29 1957. US Patent 2,811,715.
- [6] W. M. Brown, “Synthetic aperture radar,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-3, pp. 217–229, March 1967.
- [7] J. L. Walker, “Range-doppler imaging of rotating objects,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-16, pp. 23–52, Jan 1980.
- [8] J. R. Fienup and A. Kowalczyk, “Detecting moving targets in SAR imagery by using a phase-error correction algorithm,” in *Proceedings of SPIE*, vol. 2487, pp. 17–21, 1995.
- [9] S. Barbarossa and A. Scaglione, “Autofocusing of SAR images based on the product high-order ambiguity function,” *IEE Proceedings - Radar, Sonar and Navigation*, vol. 145, pp. 269–273, Oct 1998.
- [10] J. R. Moreira and W. Keydel, “A new MTI-SAR approach using the reflectivity displacement method,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, pp. 1238–1244, Sep 1995.

- [11] S. A. Werness, M. A. Stuff, and J. R. Fienup, "Two-dimensional imaging of moving targets in SAR data," in *1990 Conference Record Twenty-Fourth Asilomar Conference on Signals, Systems and Computers, 1990.*, vol. 1, pp. 16–, Oct 1990.
- [12] S. A. S. Werness, W. G. Carrara, L. S. Joyce, and D. B. Franczak, "Moving target imaging algorithm for SAR data," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, pp. 57–67, Jan 1990.
- [13] M. A. Stuff, "Three-dimensional analysis of moving target radar signals: methods and implications for ATR and feature-aided tracking," 1999.
- [14] M. A. Stuff, "Three-dimensional invariants of moving targets," 2000.
- [15] M. Stuff, *Derivation and Estimation of Euclidean Invariants of Far Field Range Data*. PhD thesis, University of Michigan, 2002.
- [16] M. A. Stuff, P. Sanchez, and M. Biancalana, "Extraction of three-dimensional motion and geometric invariants from range dependent signals," *Multidimensional Syst. Signal Process.*, vol. 14, pp. 161–181, Jan. 2003.
- [17] M. Stuff, M. Biancalana, G. Arnold, and J. Garbarino, "Imaging moving objects in 3D from single aperture synthetic aperture radar," in *Proceedings of the 2004 IEEE Radar Conference (IEEE Cat. No.04CH37509)*, pp. 94–98, April 2004.
- [18] G. J. Meyer, *Classification of Radar Targets Using Invariant Features*. PhD thesis, Air Force Institute of Technology, April 2003.
- [19] M. Ferrara, J. Jackson, and M. Stuff, "Three-dimensional sparse-aperture moving-target imaging," *SPIE Proceedings*, vol. 6970, April 2008.
- [20] M. Ferrara, G. Arnold, and M. Stuff, "Shape and motion reconstruction from 3D-to-1D orthographically projected data via object-image relations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 1906–1912, Oct 2009.

- [21] G. Arnold, M. Ferrara, and J. T. Parker, “Multiple-object shape and motion reconstruction with missing radar data,” *Proc. SPIE*, vol. 8746, pp. 87460G–87460G–15, 2013.
- [22] J. Mayhan, “Phase-enhanced 3D snapshot ISAR imaging and interferometric SAR,” *MIT Lincoln Labs*, 2009.
- [23] M. Martorella, D. Stagliano, F. Salvetti, and N. Battisti, “3D interferometric ISAR imaging of noncooperative targets,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, pp. 3102–3114, October 2014.
- [24] C. Tomasi and T. Kanade, “Shape and motion from image streams under orthography: a factorization method,” *International Journal of Computer Vision*, vol. 9, pp. 137–154, Nov 1992.
- [25] O. Özyesil, V. Voroninski, R. Basri, and A. Singer, “A survey on structure from motion,” *CoRR*, vol. abs/1701.08493, 2017.
- [26] D. Bates, J. Hauenstein, A. Sommese, and C. Wampler, “Bertini: Software for numerical algebraic geometry.” Available at [betini.nd.edu](http://betini.nd.edu).
- [27] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler, *Numerically solving polynomial systems with Bertini*, vol. 25. SIAM, 2013.
- [28] M. Compagnoni, R. Notari, A. A. Ruggiu, F. Antonacci, and A. Sarti, “The algebro-geometric study of range maps,” *Journal of Nonlinear Science*, vol. 27, pp. 99–157, Feb 2017.
- [29] T. Berry, “Points at rational distance from the vertices of a triangle,” *Acta Arithmetica*, vol. 62, no. 4, pp. 391–398, 1992.
- [30] O. Bottema, R. Erne, and R. Hartshorne, *Topics in Elementary Geometry*. Mathematics and Statistics, Springer New York, 2008.
- [31] D. Cox, J. Little, and D. O’Shea, *Ideals, varieties, and algorithms*, vol. 3. Springer, 1992.

- [32] W. Decker, G.-M. Greuel, G. Pfister, and H. Schönemann, “SINGULAR 4-1-0 — A computer algebra system for polynomial computations.” <http://www.singular.uni-kl.de>, 2016.
- [33] S. Collart, M. Kalkbrener, and D. Mall, “Converting bases with the Gröbner walk,” *Journal of Symbolic Computation*, vol. 24, no. 3-4, pp. 465–469, 1997.
- [34] Q.-N. Tran, “A fast algorithm for Gröbner basis conversion and its applications,” *Journal of Symbolic Computation*, vol. 30, no. 4, pp. 451–467, 2000.
- [35] Q.-N. Tran, “Efficient Groebner walk conversion for implicitization of geometric objects,” *Computer Aided Geometric Design*, vol. 21, no. 9, pp. 837–857, 2004.
- [36] A. Steel, “Gröbner basis timings page.” <http://magma.maths.usyd.edu.au/~allan/gb/>, 2004.
- [37] W. Bosma, J. Cannon, and C. Playoust, “The Magma algebra system. I. The user language,” *J. Symbolic Comput.*, vol. 24, no. 3-4, pp. 235–265, 1997. Computational algebra and number theory (London, 1993).
- [38] A. L. Dixon, “The eliminant of three quantics in two independent variables,” *Proceedings of the London Mathematical Society*, vol. 2, no. 1, pp. 49–69, 1909.
- [39] A. Cayley, “On the theory of elimination,” *Cambridge and Dublin Math. J.*, vol. 3, pp. 116–120, 1848.
- [40] D. Kapur, T. Saxena, and L. Yang, “Algebraic and geometric reasoning using Dixon resultants,” in *Proceedings of the international symposium on Symbolic and algebraic computation*, pp. 99–107, ACM, 1994.
- [41] D. Kapur and T. Saxena, “Extraneous factors in the Dixon resultant formulation,” in *Proceedings of the 1997 international symposium on Symbolic and algebraic computation*, pp. 141–148, ACM, 1997.
- [42] D. Kapur, “Algorithmic elimination methods,” in *Tutorial Notes, Intl. Symp. on Symbolic and Algebraic Computation (ISSAC), Montreal*, 1995.

- [43] D. Kapur, Y. N. Lakshman, and T. Saxena, “Computing invariants using elimination methods,” in *Proceedings of International Symposium on Computer Vision - ISCV*, pp. 97–102, Nov 1995.
- [44] D. Kapur, “Automated geometric reasoning: Dixon resultants, Gröbner bases, and characteristic sets,” in *Automated Deduction in Geometry* (D. Wang, ed.), (Berlin, Heidelberg), pp. 1–36, Springer Berlin Heidelberg, 1997.
- [45] A. D. Chtcherba and D. Kapur, “Constructing Sylvester-type resultant matrices using the Dixon formulation,” *Journal of Symbolic Computation*, vol. 38, no. 1, pp. 777–814, 2004.
- [46] M. Minimair, “Computing the Dixon Resultant with the Maple package DR,” in *Applications of Computer Algebra* (I. S. Kotsireas and E. Martínez-Moro, eds.), pp. 273–287, Springer International Publishing, 2017.
- [47] Toronto: Maplesoft, a division of Waterloo Maple Inc., *Maple 2015*, 2005-2015.
- [48] R. H. Lewis, “Fermat computer algebra system,” 2008.
- [49] R. H. Lewis, “Dixon-EDF: The premier method for solution of parametric polynomial systems,” in *Special Sessions in Applications of Computer Algebra*, pp. 237–256, Springer, 2015.
- [50] A. D. Chtcherba and D. Kapur, “Conditions for determinantal formula for resultant of a polynomial system,” in *Proceedings of the 2006 international symposium on Symbolic and algebraic computation*, pp. 55–62, ACM, 2006.
- [51] X. G. Fang and G. Havas, “On the worst-case complexity of integer Gaussian elimination,” in *Proceedings of the 1997 international symposium on Symbolic and algebraic computation*, pp. 28–31, ACM, 1997.
- [52] B. K. Natarajan, “Sparse approximate solutions to linear systems,” *SIAM journal on computing*, vol. 24, no. 2, pp. 227–234, 1995.

- [53] M. Fazel, *Matrix rank minimization with applications*. PhD thesis, PhD thesis, Stanford University, 2002.
- [54] B. Recht, M. Fazel, and P. A. Parrilo, “Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization,” *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.
- [55] B. Recht, W. Xu, and B. Hassibi, “Null space conditions and thresholds for rank minimization,” *Mathematical programming*, vol. 127, no. 1, pp. 175–202, 2011.
- [56] Y.-B. Zhao, “An approximation theory of matrix rank minimization and its application to quadratic equations,” *Linear Algebra and its Applications*, vol. 437, no. 1, pp. 77–93, 2012.
- [57] N. Gould, T. Rees, and J. Scott, “A higher order method for solving nonlinear least-squares problems,” tech. rep., Technical Report RAL-P-2017-010, STFC Rutherford Appleton Laboratory, 2017.
- [58] N. Gould, T. Rees, and J. Scott, “Convergence and evaluation-complexity analysis of a regularized tensor-Newton method for solving nonlinear least-squares problems,” tech. rep., Technical Report RAL-P-2017-009, STFC Rutherford Appleton Laboratory, 2017.
- [59] M. Born, “Zur quantenmechanik der stoßvorgänge,” *Zeitschrift für Physik*, vol. 37, pp. 863–867, Dec 1926.

# Appendix A

## Conventions

Since the engineering and mathematics communities often differ in common notations, here is a guide to the conventions followed in this work.

### A.1 Symbol Reference

Symbol	Meaning	Section Defined
$:=$	equal by definition	
$i = \sqrt{-1}$	imaginary unit	
$\nu$	frequency	1.1
$\omega = 2\pi\nu$	spatial frequency	1.1
$k = c_0^{-1}\omega$	angular wavenumber	1.1
$c_0$	speed of light in free space	1.1
$\mathcal{E}$	time-dependent electric field	1.1
$E$	frequency-dependent electric field	1.1
$\partial_t$	partial derivative with respect to $t$	1.1
$\nabla^2$	Laplacian operator	1.1
$\mathbf{s}_n$	target centered coordinate of $n^{\text{th}}$ scatterer	1.3
$S$	configuration/shape matrix	1.3
$\mathbf{x}_{nl}$	position of $n^{\text{th}}$ scatterer at time $l$	1.3
$X_l$	target position at time $l$	1.3
$\rho_{nl}$	range to $n^{\text{th}}$ scatterer at time $l$ .	1.3,1.3.3,1.3.5
$\boldsymbol{\rho}_l$	vector of ranges at time $l$	1.3,1.3.3,1.3.5
$R$	range data matrix, or track matrix	1.3

## A.2 Fourier Transform

There are different conventions for the Fourier transform, each of which is convenient in some situation. In this work, we will follow the conventions

$$F(\nu) = \int e^{2\pi i\nu t} f(t) dt \quad (\text{A.1})$$

$$f(t) = \int e^{-2\pi i\nu t} F(\nu) d\nu, \quad (\text{A.2})$$

where  $F(\nu)$  is the Fourier transform of the function  $f(t)$ . We sometimes use the spatial frequency for convenience, in which case we denote the transforms with

$$F(\omega) = \int e^{i\omega t} f(t) dt \quad (\text{A.3})$$

$$f(t) = \frac{1}{2\pi} \int e^{-i\omega t} F(\omega) d\omega. \quad (\text{A.4})$$

# Appendix B

## Approximations

### B.1 Born Approximation

Consider the Lippmann-Schwinger (LS) equation in the Fourier domain:

$$E^{\text{sc}}(\omega, \mathbf{x}) = - \int G(\omega, \mathbf{x} - \mathbf{z}) V(\mathbf{z}) \omega^2 E^{\text{tot}}(\omega, \mathbf{z}) d\mathbf{z}. \quad (1.10 \text{ revisited})$$

Recall that  $E^{\text{tot}} = E^{\text{in}} + E^{\text{sc}}$ , so that we can add the incident field to both sides and rewrite the LS equation in terms of operators,

$$E^{\text{tot}} + \mathcal{G}\mathcal{V}E^{\text{tot}} = E^{\text{in}}, \quad (\text{B.1})$$

with  $\mathcal{G}$  representing convolution with  $G$  and  $\mathcal{V}$  denoting multiplication by the reflectivity function,  $V$ . If possible, we would solve (B.1) with  $E^{\text{tot}} = (I + \mathcal{G}\mathcal{V})^{-1}E^{\text{in}}$ . This form suggests the Neumann series, a generalization of the geometric series. As in the geometric series  $(1 + z)^{-1} = \sum_{n=0}^{\infty} (-1)^{n-1} z^n$ , we can write

$$E^{\text{tot}} = E^{\text{in}} - \mathcal{G}\mathcal{V}E^{\text{in}} + (\mathcal{G}\mathcal{V})^2 E^{\text{in}} - (\mathcal{G}\mathcal{V})^3 E^{\text{in}} + \dots . \quad (\text{B.2})$$

The Neumann series converges when the operator norm  $\|\mathcal{G}\mathcal{V}\| < 1$ , which we can interpret to mean that  $\mathcal{G}\mathcal{V}$  is somehow small; it is common to refer to the convergence of this series as a weak-scattering assumption [2].

The Neumann series also has an interesting physical interpretation. The function  $G(\mathbf{x} - \mathbf{z})$  describes the propagation of the EM wave from  $\mathbf{z}$  to  $\mathbf{x}$ , and  $V(\mathbf{z})$  describes the reflectivity of the scatterer at  $\mathbf{z}$ . Together, we see that each application of  $\mathcal{G}\mathcal{V}$  describes the wave propagation from a single scattering event. With this in mind, we can see that the Neumann series in (B.2) is the incident field, plus singly-reflected waves ( $\mathcal{G}\mathcal{V}$  term), plus twice-reflected waves ( $(\mathcal{G}\mathcal{V})^2$  term), etc.

The Born approximation [59] involves truncating the Neumann series so that we consider only the effects of singly-reflected waves. The resulting model is

$$E^{\text{tot}} = E^{\text{in}} - \mathcal{G}\mathcal{V}E^{\text{in}}, \quad (\text{B.3})$$

from which we can now subtract the incident field and replace  $\mathcal{G}\mathcal{V}$  to find

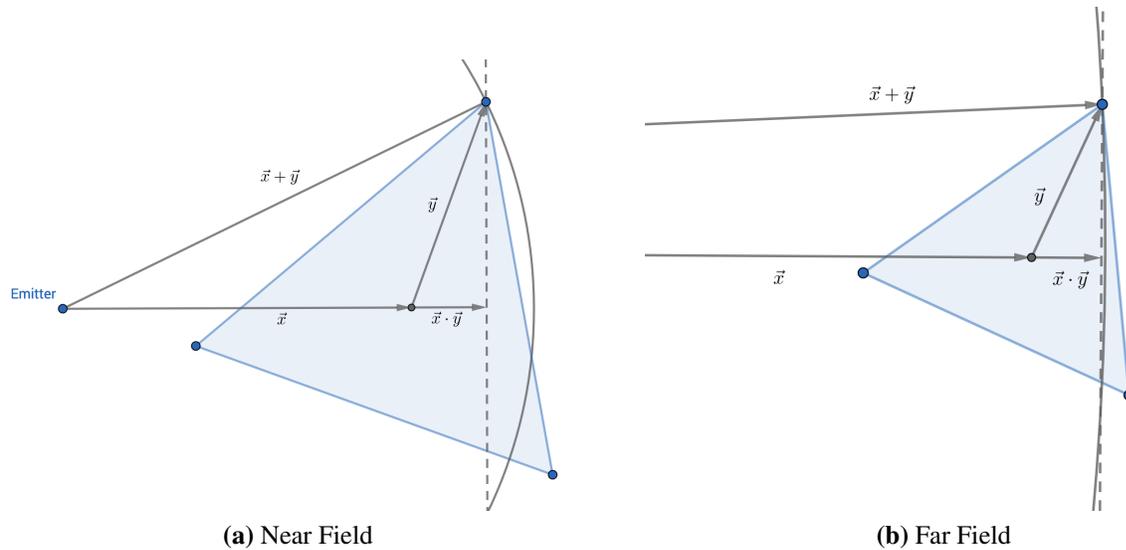
$$E^{\text{sc}} = - \int \frac{e^{ik|\mathbf{x}-\mathbf{z}|}}{4\pi|\mathbf{x}-\mathbf{z}|} V(\mathbf{z}) \omega^2 E^{\text{in}} d\mathbf{z} \quad (\text{B.4})$$

## B.2 Far-Field Approximation

In reality, wavefronts emitted from our antenna will be curved. If we imagine an antenna composed of a single point, the wavefronts will be spheres propagating outward from the point. In 2D, the wavefronts will be circles. This means that when we measure the range to a point in space with a radar system, our measurement is an estimate of the Euclidean distance between the point-antenna and the point-scatterer on the target. Suppose our antenna is centered at the origin, our target is centered on the point  $\mathbf{x}$ , and our measurement comes from scatterer at point  $\mathbf{y}$  relative to the target center, as in Figure B.1. Relative to the sensor, the scatterer is then at position  $\mathbf{x} + \mathbf{y}$ , so that the range is the distance from the sensor to the scatterer,

$$\begin{aligned} \|\mathbf{x} + \mathbf{y}\| &= \sqrt{(\mathbf{x} + \mathbf{y})^T (\mathbf{x} + \mathbf{y})} \\ &= \sqrt{\|\mathbf{x}\|^2 + 2\mathbf{x}^T \mathbf{y} + \|\mathbf{y}\|^2} \\ &= \|\mathbf{x}\| \sqrt{1 + \left( 2 \frac{\hat{\mathbf{x}}^T \mathbf{y}}{\|\mathbf{x}\|} + \frac{\|\mathbf{y}\|^2}{\|\mathbf{x}\|^2} \right)}, \end{aligned} \quad (\text{B.5})$$

where we define  $\hat{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|}$  to be the unit vector in the direction of  $\mathbf{x}$ , the target centroid. In the far-field,  $\|\mathbf{x}\| \gg \|\mathbf{y}\|$  since the distance from the sensor to the target centroid is much greater than the distance from the target centroid to the scatterer. This allows us to use the Taylor expansion  $\sqrt{1+z} = 1 + \frac{z}{2} + \frac{z^2}{8} + \dots$  to approximate the range with



**Figure B.1:** When the distance between scatterers is small compared to the distance from the target to the emitter, we approximate the circular wavefronts with parallel planes. We can see that this is a poor approximation in the near field, but that the accuracy improves as the wavefront curvature in the region near the object decreases.

$$\begin{aligned} \|\mathbf{x} + \mathbf{y}\| &= \|\mathbf{x}\| \left[ 1 + \left( \frac{\hat{\mathbf{x}}^T \mathbf{y}}{\|\mathbf{x}\|} + \frac{\|\mathbf{y}\|^2}{2\|\mathbf{x}\|^2} \right) + \frac{1}{2} \left( \frac{\hat{\mathbf{x}}^T \mathbf{y}}{\|\mathbf{x}\|} + \frac{\|\mathbf{y}\|^2}{2\|\mathbf{x}\|^2} \right)^2 + \dots \right] \\ &\approx \|\mathbf{x}\| + \hat{\mathbf{x}}^T \mathbf{y}, \end{aligned} \quad (\text{B.6})$$

where we have dropped terms containing powers of  $\frac{\hat{\mathbf{x}}^T \mathbf{y}}{\|\mathbf{x}\|}$  and  $\frac{\|\mathbf{y}\|^2}{\|\mathbf{x}\|^2}$ . This approximation is extremely useful when it is valid, as it linearizes the range measurements. In radar applications, the combination of the Born approximation and the far-field approximation simplifies the imaging operator in (1.24) to a Fourier transform. For more information, see the book by Cheney et al. [2].