Summer 7-2022

One-Bit Algorithm Considerations for Sparse PMCW Radar

Ethan Triplett
University of Nebraska-Lincoln, etriplett2@huskers.unl.edu

ONE-BIT ALGORITHM CONSIDERATIONS FOR SPARSE PMCW RADAR

by

Ethan Triplett

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Telecommunications Engineering

Under the Supervision of Professor Andrew Harms

Lincoln, Nebraska

July, 2022

ONE-BIT ALGORITHM CONSIDERATIONS FOR SPARSE PMCW RADAR

Ethan Triplett, M.S.

University of Nebraska, 2022

Advisor: Andrew Harms

Phase Modulated Continuous Wave (PMCW) radar an emerging technology for autonomous cars. It is more flexible than the current frequency modulated systems, offering better detection resolution, interference mitigation, and future development opportunities. The issue preventing PMCW adoption is the need for high sample-rate analog to digital converters (ADCs). Due to device limits, a large increase in cost and power consumption occurs for every added resolution bit for a given sampling rate. This thesis explores radar detection techniques for few-bit and 1-bit ADC measurements. 1-bit quantization typically results in poor amplitude estimation, which can limit detections if the target signals are weak. Time Varying quantization Thresholds (TVTs) are a way to preserve that amplitude information.

An existing few-bit Fast Iterative Shrinkage Thresholding Algorithm (FISTA) was adapted to use 1-bit TVT quantization. Three test scenarios compared the original FISTA using 1 and 2-bit quantization to the TVT approach. Tests included widely spaced targets, adjacent targets, and high dynamic range targets. Performance metrics included normalized mean squared error (NMSE) of target amplitude estimation and Receiver operating characteristic (ROC) curves for detection accuracy. Results showed the TVT implementation operated over the widest range of SNR values, had the lowest amplitude estimate NMSE at high SNR, and comparable NMSE with 2-bit FISTA at low SNR. There was an $84-93\%$ reduction in NMSE compared to 1-bit FISTA without TVTs. Few-bit FISTA had the best detection rates at specific SNR values, but was more sensitive to noise. AUC values averaged across the full SNR range for TVT FISTA were the most robust, measuring $13-46\%$ higher than 1-bit FISTA and $48-74\%$ higher than 2-bit FISTA.

To my mother, Judy, my father, Dan, and my cat, Sam.

Acknowledgments

First, I have to thank my friends and my parents. You came together for me through the greatest challenges I have known. I also want to thank Dr. Harms, my advisor. If his classes weren't so fun I would not have discovered my enthusiasm for this topic, and his depth of knowledge meant I was always introduced to a concept right when I was ready to use it. I am truly blessed to have such wonderful and supportive people in my life.

Contents

List of Tables

List of Figures

List of Algorithms

List of Acronyms

ADAS Advanced Driver Assistance Systems

ADC Analog to Digital Converter

ADMM Alternating Direction Method of Multipliers

AMP Approximate Message Passing

AUC Area Under the Curve

AWGN Additive White Gaussian Noise

CDF Cumulative Distribution Function

CGLS Conjugate Gradient Least Squares

CPI Coherent Processing Interval

CRB Cramer-Rao Bound

DR Dynamic Range

EM Expectation Maximization

FFT Fast Fourier Transform

FISTA Fast Iterative Shrinkage Thresholding Algorithm

FMCW Frequency Modulated Continuous Wave

FoM Figure of Merit

FN False Negative

FP False Positive

FPR False Positive Rate

FPS Frames Per Second

GAMP Generalized Approximate Message Passing

IAA Iterative Adaptive Approach

IF Intermediate Frequency

LASSO Lease Absolute Shrinkage and Selection Operator

LIDAR Light Detection And Ranging

LIKES Likelihood-based Estimation of Sparse parameters

MIMO Multiple Input Multiple Output

ML Maximum Likelihood

MM Maximization Minimization

MSE Mean Squared Error

NMSE Normalized Mean Squared Error

NRMSE Normalized Root Mean Squared Error

PDF Probability Density Function

PMCW Phase Modulated Continuous Wave

PSD Power Spectral Density

RADAR Radio Detection And Ranging

ROC Receiver Operating Characteristic

RGB Red, Green, and Blue

SAE Society of Automotive Engineers

SISO Single Input Single Output

SLAM Simultaneous Localization and Mapping

SLIM Sparse Learning via Iterative Minimization

SNR Signal-to-Noise Ratio

SpaRSA Sparse Reconstruction by Separable Approximation

SPICE Sparse Iterative Covariance-based Estimation

TN True Negative

TP True Positive

TPR True Positive Rate

TVT Time Varying Thresholds

VAMP Vector Approximate Message Passing

VCO Voltage Controlled Oscillator

List of Symbols

R	Target range
ΔR	Range resolution
R_{max}	Maximum unambiguous range
v	Target velocity
Δv	Velocity resolution
v_{max}	Maximum unambiguous velocity
f_c	Carrier frequency
λ	Signal wavelength
$\mathbf{B}$	Signal bandwidth
Δf	Doppler frequency shift
$\mathbf{c}$	Speed of light
K	Number of targets
$s(t)$	Transmit signal
α	Signal power at the transmitter
$e(t)$	Additive White Gaussian Noise

$\mathbf{0}_N$	Zero-vector of length N
γ_k	Amplitude of the target return per target k
$\Delta\phi$	Phase difference between two signals
$T_{\Delta c}$	Time difference between sending two chirps
T_f	Number of chirps per CPI
L_p	Number of pulses per CPI
T_p	Pulse duration
L_c	Total number of chips per pulse
T_c	Chip duration
M	Number of samples collected at the receiver
Δt	Sampling period
$\Re\{\}$	Function that keeps only real component of a complex value
$\Im\{\}$	Function that keeps only imaginary component of a complex value
x	Sparse complex signal vector
$\tilde{\text{x}}$	All-real signal from vectorization of real and imaginary magnitudes of x
$\hat{\tilde{\text{x}}}$	Estimated all-real signal vector
$\tilde{\text{x}}_g$	Set of real and imaginary magnitudes of x for group g
M	Length of the sparse signal vector
N	Number of collected signal measurements
A	Complex valued measurement matrix of dimension $\mathbb{C}^{MxN}$

$\tilde{A}$	All-real measurement matrix formed from A with dimension $\mathbb{C}^{2Mx2N}$
$\tilde{a}_m$	One row of the measurement matrix
y	Measured/Received complex signal
$\tilde{y}$	All-real measurements from vectorization of real and imaginary magnitudes of y
$\bar{y}$	Quantized measurements
Q()	Real-valued quantization function
q	Bit-depth used to determine the number of quantization values
D	Number of quantization thresholds
R_{max}	95% of the largest measured signal amplitude
Δs	Quantization interval step size
u_d	The quantization threshold immediately below the measured value
u_{d+1}	The quantization threshold immediately above the measured value
U_d	Vector of all thresholds immediately below the measured values of y
U_{d+1}	Vector of all thresholds immediately above the measured values of y
h	Time-varying threshold vector of complex thresholds
$\tilde{h}$	All-real TVT from vectorization of real and imaginary magnitudes of h
i	Iteration index
G	Number of groups formed in FISTA, equal to the length of x
z	Scaled signal estimate for FISTA
$h(\mathrm{z})_g$	Function to shift z by one step in the direction of the gradient

$L(\mathrm{z}\|\bar{y})$	The negative log-likelihood function
$\nabla L(\mathrm{z}\|\bar{y})$	Gradient of the negative log-likelihood function with respect to z
σ	Noise standard deviation
ρ	Sparse regularization coefficient
ζ	Step size for FISTA iterations
$\phi()$	Standard Normal PDF
$\Phi()$	Standard Normal CDF
η	Acceleration term for FISTA

CHAPTER 1

Introduction

Advanced driver assistance systems (ADAS) are a collection of sensors and computing power designed to enable autonomous driving. Automotive radar is one of the essential parts of these systems, valued for performance even in adverse conditions such as rain, fog, snow, and at night [1]. It can inherently gather information about the relative speed, direction, and distance of nearby objects. It can be designed for both long and short range applications. In an advanced setup, more than one radar module will be mounted to a car, each pointing in different directions to create a full coverage of the environment. These are the reasons why radar is commonly included in an ADAS design.

Demands on ADAS are great due to the number of environments cars operate in. Complex areas with dense traffic, pedestrians, and cyclists must be interpreted accurately and quickly . Many unique cases also occur and have to be accounted for. Situations such as road debris, driving in tunnels, and on roads with poorly defined boundaries each have unique challenges. One noteworthy example from 2020 is a crash that occurred in Taiwan [2]. A Tesla Model 3 failed to detect an overturned truck on the highway, leading to the car driving directly into the truck at highway speeds. This happened despite clear, sunny weather and otherwise good road conditions. Accidents like this raise concerns about the safety of autonomous cars. Those concerns grow as more cars become equipped with ADAS

sensors, which increases signal interference on the road [3]. These challenges have pushed ADAS radar designs to use higher capability, more efficient, and smaller package designs [4]. This has pushed development towards phase modulated radars because they offer tighter integration and other opportunities for growth in ADAS.

The chapter below provides the background necessary to understand the demands of ADAS and the fundamentals of radar.

1.1 Vehicle Automation

Modern cars are evolving to use more technology aimed at driver comfort, convenience, and safety. The ultimate realization of this goal is the fully autonomous car, that can take a passenger from start to destination with no human input. This would allow people to travel regardless of age, disability, or impairment such as fatigue. Autonomous cars also promote the safety of others, with collision avoidance and pedestrian detection as fundamental requirements. Within the industry this has resulted in an increasing amount of ADAS sensors embedded in each car. This section will discuss the sensors used in an autonomous car and the different levels of autonomy that can be achieved.

1.1.1 Levels of Automation

The abilities of ADAS are categorized into five levels of autonomy, defined by the society of automotive engineers (SAE). [5]. Each level ascending from L0 to L5 increases the amount of control that the car is capable of while operating autonomously. At L0 the driver is solely responsible for controlling the vehicle with no assistance. L1 is considered to be assistance only, introducing basic abilities such as radar cruise control. These are simple systems that only try to maintain a safe distance to a car directly ahead, in the same lane as the controlled vehicle. They can also have the ability to do some braking. L2 automation is the first stage to be considered automation, with lane keeping and other steering controls forming a basic

ADAS system. There is no capacity for planning at this stage, so the driver has to make decisions about things like passing a slow vehicle.

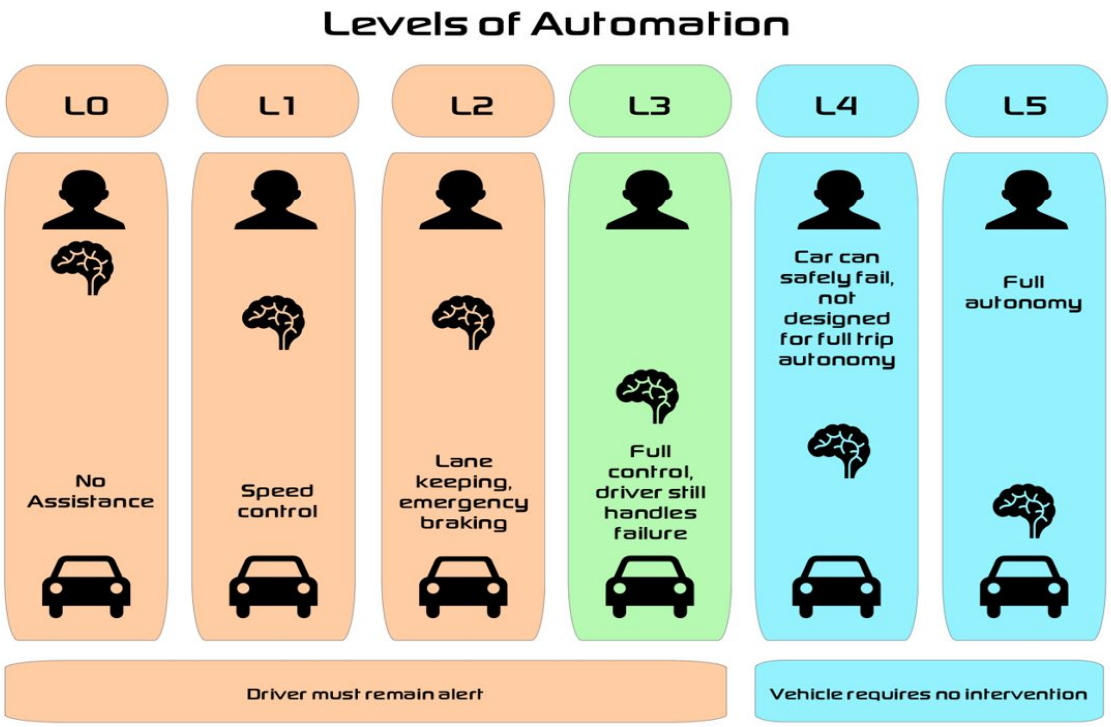

Figure 1.1: *Levels of automation as defined by SAE International.*

L3 is the first level with true automation. The system is expected to know its limitations and monitor for failures, but otherwise it is capable of fully independent operation. The driver in this case has to be able to take control when the system identifies that it is outside of its design constraints. L4 introduces the ability to reach a minimal risk condition if the system fails, so the driver is no longer expected to provide any input or supervision. Alerts can still be issued that the driver should take over, but the car can still find safety if the driver does not respond. This level does not require the ability to handle all environments, so an example system might only handle highway operation with the driver taking over once it is time to exit the highway. Finally, L5 represents complete autonomy, with the ability to handle a full trip with the destination as the only input. The system will find a safe condition for extreme events such as a blizzard or flood where it cannot continue.

Hardware and software performance demand is high regardless of which level a vehicle is operating at. Hazardous events must go from detection to action as quickly as possible, with the safest outcome always chosen. Each level has to be clear about how well the car can make these decisions so the driver knows when to take over. From the perspective of ADAS, this becomes a balance between detection speed and detection accuracy. Failure to strike that balance can cause a false detection or a missed detection. Both can be equally dangerous.

1.1.2 Sensor Types

Autonomous cars need to be able to interpret their surroundings in order to work. There are three main sensors used for this purpose: light detection and ranging (Lidar), cameras, and radio detection and ranging (Radar) [3][6]. Each has unique strengths and weaknesses that determine how they are used in ADAS. This section will explore the properties of these three sensors and how they contribute to autonomous driving.

First is Lidar, which provides very detailed distance measurements. It works by sending out laser pulses, then tracking the time-of-flight for each pulse to reflect off an object before returning [3][7]. Because it is a round trip journey, each measurement is twice the distance between the Lidar and the nearby object. The collected measurements are called a point cloud, which can be thought of as a list of every direction the light was aimed, and the corresponding delay at that angle. The scanning area depends on the design of the Lidar module, with one, two, and three dimensional options. The requirements for Lidar in ADAS includes at least 100 m of range, the ability to distinguish objects closer than 25 cm apart, and at least 25 frames per second (fps) update rate [8]. There is a large number of manufacturers producing Lidar modules that meet these specifications [7]. The range and resolution capabilities of Lidar mean that it is effective for tasks such as lane changes, adaptive cruise control, parking, and collision warnings.

There are a few drawbacks to Lidar that must be considered [7][9]. First, the preci-

sion of using a laser-based system makes proper calibration a challenge. Imperfections during manufacturing and installation, or damage accumulated over time can disrupt the accuracy of measurements. Next there are environmental challenges. Interference from other autonomous cars occurs anytime signals from one Lidar module interfere with the measurements of another. This is addressed by using unique transmit patterns from vehicle to vehicle. Lidar is also weak in poor weather because the signals reflect off of heavy rain, snow, or fog. These reflections stand out when compared to neighboring point cloud values, allowing for identification and filtering through software.

There are challenges in data processing as well. A 3D Lidar with high resolution creates a large amount of data that has to be processed, in the Gbps range. Interpreting and acting on that volume of data is difficult. Solutions focus on compressing the data before processing. Once the data size is reduced enough for processing, the final task is identifying what has been measured. Ground segmentation is the process of identifying the area in the point-cloud associated with road surfaces. Isolating this area makes it easier to focus on lane-keeping, and to identify non-ground objects nearby. The relative position of the car with its surroundings has to be tracked as well, using Simultaneous Localization and Mapping (SLAM). The difficulty here lies in identifying which groups of distance measurements can be combined into one object. Overall the weaknesses of Lidar are either being addressed directly, or by combining data with other sensors which will be discussed in the sensor fusion section [7][10].

Cameras are the next major sensor common to ADAS. They collect information by storing light color and intensity measurements in a grid of values, called a frame [11]. A video is made up of a series of frames, while a picture is just one frame. Each grid location within the frame is called a pixel. For black and white data each pixel is a single value representing brightness. For color, three values are stored that represent the relative amount of red, green, and blue (RGB) detected. The resolution of the camera is defined by the number of these pixels that can be measured at a time. Camera requirements for ADAS

include resolutions up to 1920x1080, or 2 Megapixels, while capturing up to 60 fps [12]. Of the three main ADAS sensors, cameras collect the most detailed information. They are used for object detection and classification, but also have the unique ability to read signs and identify colors [3]. They are useful in all typical driving operations including pedestrian detection and emergency steering.

Cameras have a few big challenges. They do not provide their own illumination, so they only work in areas that are clearly lit and visible [9]. Next, a camera chosen with enough resolution for ADAS will end up collecting too much data. Computer vision algorithms have to sort through potentially millions of pixels, each of which could be stored as the three RGB values. This is a resource intensive process. The images themselves can be distorted by vibrations while driving. Finally, the data cameras collect only represents a two-dimensional version of the environment [6]. Detecting the distance between objects cannot be done using only one camera. There are stereo cameras that combine two camera sensors a fixed distance apart. This allows the cameras to process the angles between objects in their frames to calculate distance. Altogether cameras capture the most detail, but also have the greatest computational load in an ADAS setup.

The last common sensor in ADAS is radar. It works by transmitting a radio frequency signal in a fixed direction, where it will bounce off of any object close enough to be detected [1]. Reflected signals return back to the radar receiver where they are processed. This is the same process as Lidar, with the key difference being the use of radio waves instead of laser signals. The signals that return to the receiver experience a delay and a Doppler shift. The delay comes from the time required for the signal to reach the target and return, providing a distance measurement. The Doppler shift is a change in the signal frequency that occurs when there is a speed difference between the radar source and the objects being detected. Vehicles traveling the same speed and direction will observe no frequency shift. This comes from the Doppler effect, where a signal source moving closer has a shift to a higher frequency, and moving away shifts the frequency lower. Angle measurements are

also possible. More detail will be provided in the radar fundamentals section.

Radar used in ADAS can be divided into two categories [3]. There is short-range radar which uses 24 GHz or 79 GHz signals to detect targets across a wide field of view. The range is limited to 30 meters, so the priority for detection is nearby vehicles or other obstacles. Long range radar uses 77 GHz signals. A narrower field of view is required to increase in range, but targets can be detected as much as 250 meters away [13]. ADAS radar is used for lane keeping, cruise control, and blind spot detection [9].

The main weakness in radar is a lack of measurement resolution [3]. As of 2019, the state of the art systems were only capable of 10 cm accuracy [13]. This makes it hard to detect small targets, but it also means that objects that are close together will be seen as one entity. Radar is also vulnerable to noise and interference[14]. This can come from radar modules on other cars, but can also appear in highly reflective environments where the original signal may bounce between multiple objects before returning. Improving both detection capability and interference rejection are active research areas for radar.

1.1.3 Sensor Fusion

As the previous section established, each of the three sensors used in ADAS applications has strengths and weaknesses. While research is always improving their individual performances, the solution for ADAS systems on the road today is to combine them. Sensor fusion is the process of taking data from a group of individual sensors to create one unified understanding of the environment [15]. This approach uses the complementary sensor features to provide higher ADAS performance. Examples include the combination of Lidar and cameras. Here the large data size of a camera image can be filtered by referencing Lidar data, which indicates where objects are likely to be [10]. A similar approach can be taken when combining cameras with radar [16]. Using both approaches provides even more redundancy, since Lidar can provide high resolution data in good weather, and radar can provide accurate data in bad weather.

Companies that are testing autonomous cars rely on sensor fusion to get the best ADAS performance. A real world example is Google's early modified Toyota Prius, that used two Lidar systems, four radars, and a camera [17]. More recent is Waymo's autonomous Fiat-Chrysler Pacifica, which also features four radar modules but increases to three Lidars and a 360 degree camera [18]. Both of these cars completed hundreds of thousands of miles driving autonomously. Other autonomous car approaches also use at least two of the three sensors [6]. This shows that no one sensor is favored for ADAS applications.

1.2 Radar Fundamentals

This section will focus on the core concepts of radar. First is a high level look at the stages a radar signal goes through from start to finish. Next is a breakdown of how range, speed, and angle measurements are extracted from the returned signals. Finally, the process of taking a signal from analog to digital is explored.

1.2.1 Radar System Model

There are many approaches to building a radar module, but all designs incorporate the following elements in some way. First a signal is created in the transmitter. The choice of signal has a significant impact on what hardware is required both in the transmitter and receiver. Next the signal is transmitted, where it will reflect off of nearby objects, or targets. This is when the useful information about the target distance and speed is formed. The reflected signal returns to the receiver where it is mixed with a copy of the original signal. How this is done depends on the signal type, but the goal is to highlight the differences between the original and reflected signals.

The next stage removes any unnecessary frequencies through filtering to reduce the data size. The final hardware stage is an analog to digital converter (ADC). As the name suggests, it turns the analog signal into a binary equivalent output. The accuracy of the signal after

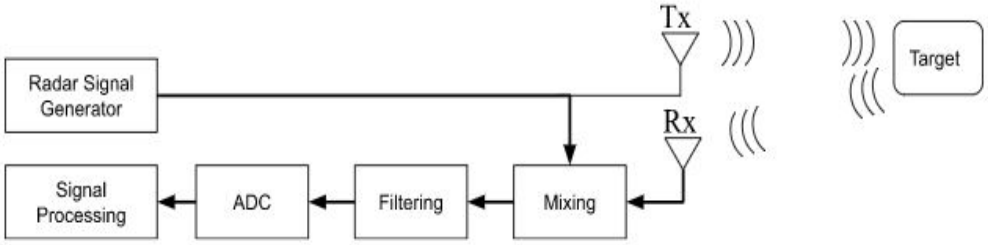

Figure 1.2: *Block diagram of a typical radar system.*

conversion is a consideration when choosing an ADC, and will be discussed in the sampling and quantization section. Finally, the signal is ready to be processed. This stage includes identifying the distance, speed, and angle information within the signal, tracking targets as more reflections arrive, and managing interference. The signal-specific details of these stages will be covered later in the transmit signal approach section.

1.2.2 Radar Performance Metrics

This section will cover how range, velocity, and angle information is measured with radar. The quality of these measurements is described in terms of coverage, resolution, and scan time [1]. Coverage describes the span of values that can be measured, and is affected by signal attenuation, signal structure, and levels of noise and interference. The individual step size that can be distinguished within the coverage area is called resolution. Resolution is determined by the signal frequency and the waveform type. Scan time is the rate of data collection, which depends on the hardware and software design chosen.

Radar measures range by the time elapsed between transmitting and receiving. The radio waves used in radar travel at the speed of light c, so for a given travel time τ, the range is

$$R = \frac{c\tau}{2} \tag{1.1}$$

This delay can only be measured if there is a distinct feature in the signal to use as a

timing reference [19]. The range coverage is determined by the maximum unambiguous delay, which is determined by the type of signal transmitted. The most basic example of this is a pulsed radar that sends a short burst and then waits for it to return. If two identical pulses were sent in close succession, there would be no way to know if a reflection came from the first pulse or the second. Range resolution is based on the bandwidth of the transmitted signal, and is defined as

$$\Delta R = \frac{c}{2B} \tag{1.2}$$

Increasing the signal bandwidth therefore provides better range resolution. Signal specific range details will be covered while discussing waveform types.

Velocity measurements are taken using the Doppler shift in the received signal [1] [20]. They are based on the carrier frequency and the difference in velocity between the radar source and target. It is calculated by

$$\Delta f = \Delta v \frac{2 f_c}{c} \tag{1.3}$$

A target traveling at the same speed and direction as the radar source will appear with zero Doppler shift. Velocity coverage comes from the max unambiguous velocity, which depends on the transmit signal type, as does the velocity resolution. Improving velocity measurements generally requires a longer signal duration though, which increases the scan time [20].

The last measurement category is the angle of the target relative to the radar module. Angle measurements require more hardware in the radar system. One approach is mechanical scanning, where the radar is rotated so that signals are sent in multiple directions. With this approach a target will appear to be closest when the signal is sent directly at it. Aiming the signal in different directions like this takes time, so it increases the scan time. A better option is to use more antennas at the transmitter and receiver. This is referred to as a multiple

input, multiple output (MIMO). With this approach reflected signals reach the antennas at different times, which is used to determine the angle it originated [20]. Mathematically, this is calculated with

$$\theta = sin^{-1}(\frac{\lambda \Delta \phi}{2\pi l}) \tag{1.4}$$

Where the signal wavelength is λ, $\Delta\phi$ is the change in phase due to a difference in travel time, and l is the distance between the antennas. The angle coverage and resolution both depend on the radar field of view and are inversely related. Transmitting over a wider area decreases the resolution.

1.2.3 Sampling and Quantization

Before a signal can be processed digitally it must be converted from a continuous analog signal into a discrete digital representation. This is done by the ADCs in the receiver through sampling and quantization [21]. Sampling occurs first. This involves measuring the amplitude of the signal at the input of the ADC. The frequency that the measurements can be taken is called the sampling rate or conversion rate. It is represented in Hertz and is the inverse of the sampling interval, or the time between two samples. Figure 1.3 shows what the sampling process looks like with a sinusoidal signal. In this example there is an average of three samples per period of the wave. The sampling interval is small enough that a wide range of amplitude measurements are captured. A signal cannot be accurately represented if the sampling rate is too low, because there will be too many changes in signal level between samples. The standard sample rate for signals of finite bandwidth is twice the signal frequency, called the Nyquist rate [22].

Once the signal is sampled, it can then be quantized. This involves a set of thresholds called a partition, and a set of binary output values called a codebook. Each threshold is set at a voltage level that allows for even spacing across the range of inputs. One codebook value

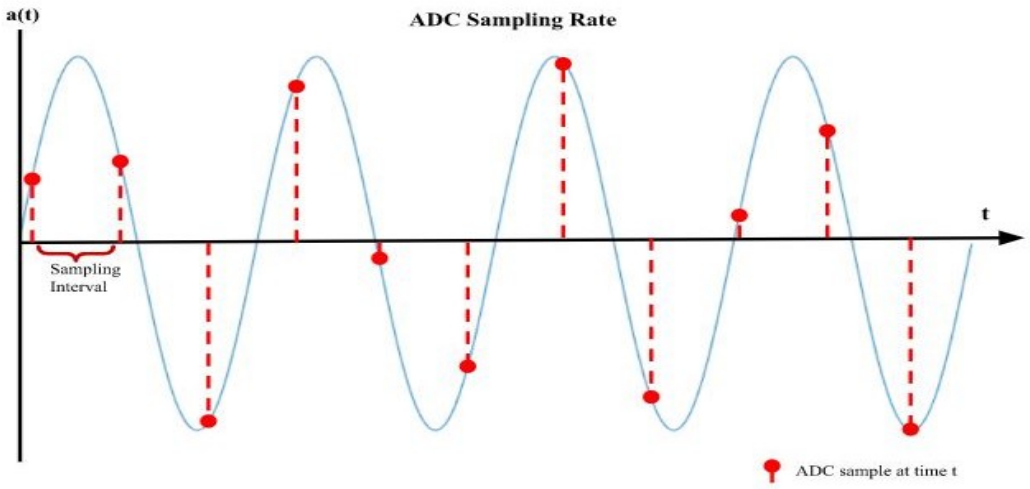

Figure 1.3: *Sampling a sinusoidal signal with an ADC.*

is assigned for each region created by the thresholds. The amplitude of each sample is then compared to determine which region it belongs to. The result is a stair-step approximation of the original signal. Figure 1.4 shows an example comparing an original signal to its quantized output. Note the first two samples, which have nearly the same amplitude but are assigned different values due to how close they are to a threshold. Similarly, the symmetric shape of the original signal is distorted because of the sampling rate and the number of output levels.

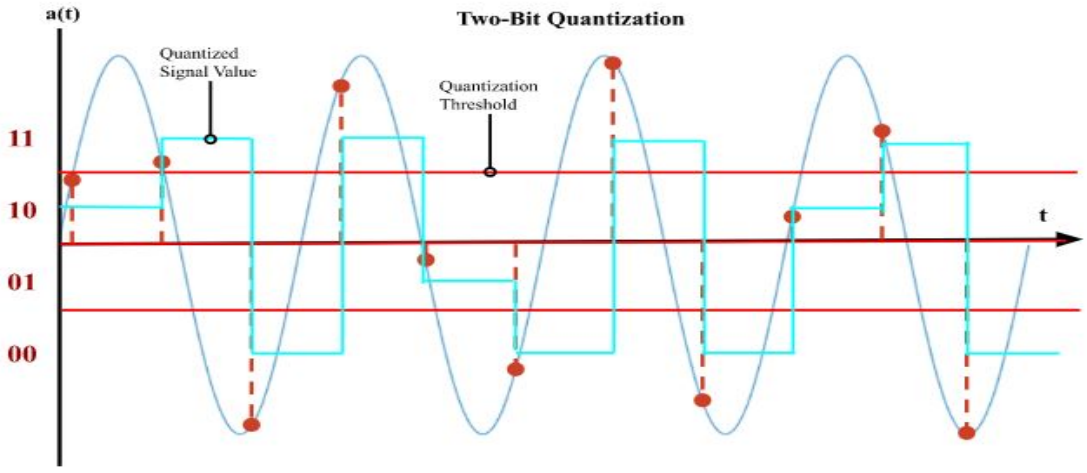

Figure 1.4: *Quantization by assigning binary values to ADC samples.*

When designing an ADC, manufacturers choose how many bits will be available to

represent the output values. This is called the bit depth or resolution of the ADC [23]. For n bits there are 2^n available values and 2^n-1 thresholds. The resolution determines how well changes in signal amplitude can be measured. Because signals are effectively rounded to the nearest discrete value, using more bits will reduce the error from quantization [24]. Depending on the application this can have an impact on the power consumption or the sampling rate.

1.2.4 Radar Digital Representation

After quantization the radar data is now in a discrete digital form. After processing this data can be thought of as a grid formed by every possible combination of range and Doppler measurements for a given transmit pattern. The original analog signals are not likely to land exactly on a grid point, so the grid must be made as fine as possible to reduce the potential estimation error.

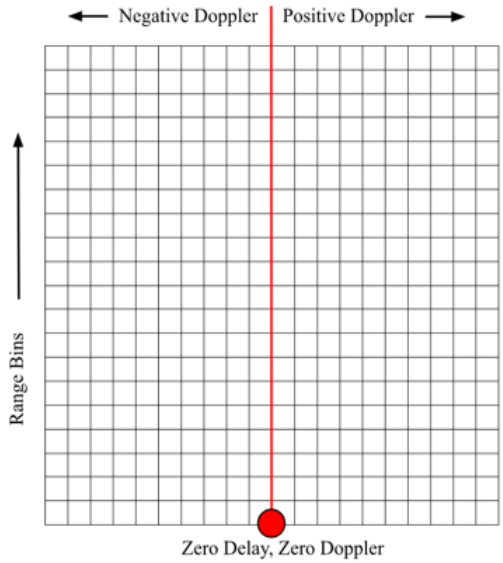

Figure 1.5: *Discrete grid based representation of radar data.*

Another important note is that most of the grid values will not have targets associated with them. This is because many of the radar signals will not reflect off of objects close enough to be detected. The result is a signal comprised mostly of noise, interference, and a

small number of target reflections. A signal like this with few non-zero entries is considered to be sparse.

1.2.5 Sparse Signal Processing

The mathematical technique for estimating the non-zero entries is called sparse signal processing, or compressed sensing. It was first developed in 2005 by Candes, Romberg, and Tao [25], and was further defined by Donoho in 2006 [26]. The fundamental concept adapts classic linear regression to favor sparse solutions by adding a penalty term. This penalty term is typically an L1, L2, or log norm scaled by a sparse regularization coefficient. A larger regularization term causes the regression to favor the all-zero vector, while a smaller term gives results closer to the original linear regression output.

1.3 Thesis Organization

The remainder of the thesis is divided into the following sections: First, the problem statement section will compare two popular ADAS radar signal types to isolate the current research challenges. Next, the literature review section summarizes existing research solutions to those challenges, noting their strengths and weaknesses. After that is the proposed solution section that covers the specific solution explored in this work. All results generated from that approach are discussed in the results and analysis section. Finally a conclusion summarizes this work and any open research questions that remain.

CHAPTER 2

Problem Statement

The goal of this thesis is to examine how the continued growth of ADAS radar can be met without sacrificing power efficient design. This chapter will begin by examining two popular transmit signal approaches for ADAS radar. They will then be compared against the current problems in the industry. The best modulation scheme will then be selected for further exploration.

2.1 Radar Transmit Signal Approaches

The waveform parameters at the transmitter have a large impact on overall detection capability. The signal structure will determine the maximum range, vulnerability to interference, and the ability to distinguish targets. There are two methods that will be discussed. The first is frequency modulated continuous wave (FMCW), which is the most common in the industry today. The other is phase modulated continuous wave (PMCW). It is an approach that has recently become feasible due to advancements in circuit technology. This section will discuss their characteristics and necessary considerations for their implementation.

2.1.1 FMCW Radar

FMCW radar is the current standard in the automotive industry. It works by sending a continuous series of frequency modulated signals called chirps. Each chirp is a waveform with linearly increasing or decreasing frequency over a fixed period in time. The signal produced at the transmitter is defined as

$$s(t) = e^{j2\pi(f_c)+0.5\alpha t)t} \quad 0 \leq t \leq T \tag{2.1}$$

Where f_c is the carrier frequency and α is the chirp slope [19].

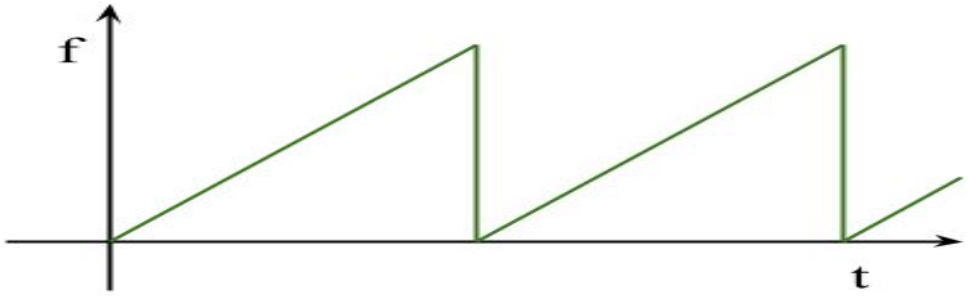

Figure 2.1: *A frequency modulated chirp waveform.*

After transmission, any reflections from the environment return to the receiver. The mixing stage combines a delayed copy of the original signal with the received signal. This process is called dechirping, which preserves only the differences between the two signals [27]. What remains is called a beat signal, or intermediate frequency, which contains the data about all target returns with additive noise. The range of beat frequencies is lower than the chirp frequencies, so the signal then goes through a low pass filter before sampling. This means that the signal bandwidth entering the ADC is much smaller than the transmitted waveform, which is one of the main advantages of FMCW radar. Finally, detections are made using Fast-Fourier Transforms to locate peaks in the beat signal.

The use of the chirp waveform is due to the benefits it offers when processing target data. The maximum unambiguous range for FMCW is defined as

$$R_{max} = \frac{cT_c}{2} \tag{2.2}$$

Where T_c is the duration of one chirp. The range resolution is defined as

$$\Delta R = \frac{c}{2B} \tag{2.3}$$

Where B is the bandwidth of one chirp. These two equations provide insight into why FMCW is useful in radar. Without the modulation there is a trade-off between signal duration and bandwidth. Increasing the range decreases the resolution and vice versa. The frequency sweeping pattern in FMCW allows for both a longer duration signal and a larger bandwidth, providing both range and resolution benefits. This concept is called pulse compression. Velocity processing in FMCW compares the phase of two returned chirps. It is calculated by

$$v = \frac{\lambda \Delta\phi}{4\pi T_{\Delta c}} \tag{2.4}$$

Where $\Delta\phi$ is the phase difference between the signals, $T_{\Delta c}$ is the delay between the first and second chirp, and λ is their wavelength [20]. Velocity resolution is calculated with

$$\Delta v = \frac{\lambda}{2T_f} \tag{2.5}$$

Where T_f is a frame, or collection of chirps. Velocity resolution improves for longer signal lengths, which gives it an inverse relationship with range resolution.

2.1.2 PMCW Radar

PMCW radar relies on changes in phase instead of frequency when generating the transmit signal. Sequences of binary values called chips are used to alternate the phase of the transmit waveform. Mathematically, this appears as

$$s(t) = c\alpha cos^{2\pi f_c t} \tag{2.6}$$

Where α is the signal power and c is the chip value, either +1 or -1 to create alternate phases [1]. When the signal returns to the receiver it is down converted in the mixing stage before being sampled by the ADC. The key difference here compared to FMCW is that the signal still has the full transmitted bandwidth, so there is more data to convert at this stage. This also means the full received signal is available for signal processing. Detections are made using the auto-correlation of the transmitted and received waveforms.

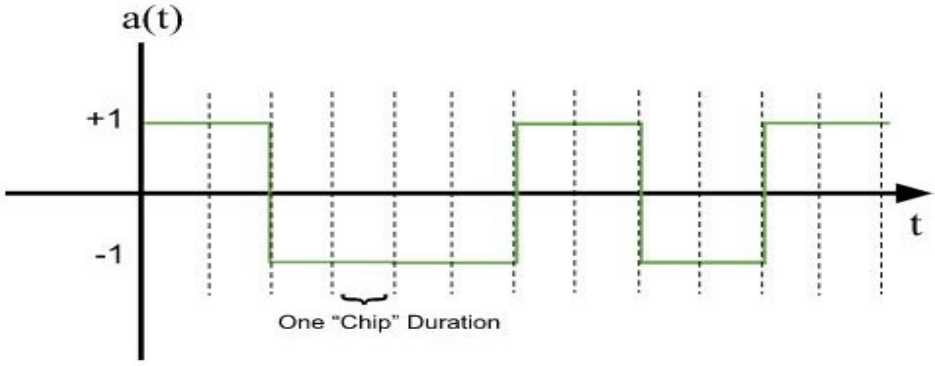

Figure 2.2: *A Binary Chip Sequence.*

The use of a chip sequence is the reason PMCW is useful in radar. In addition to providing the pulse compression aspects of FMCW chirps, they also have the ability to design for better auto-correlation properties [28]. The ideal sequence would have a large peak at zero-offset, and values as small as possible otherwise in the side-lobes. Range coverage in PMCW is based on the overall sequence length, with the maximum unambiguous range given by

$$R_{max} = c\frac{T_c L_c}{2} \tag{2.7}$$

Where T_c is the duration of a chip, c is the speed of light, and L_c is the number of chips in the sequence [1]. The maximum range can also be thought of as the furthest distance

measured before the chip sequence has to be repeated. The range resolution is based on an individual chip length

$$\Delta R = \frac{cT_c}{2} \tag{2.8}$$

Which means that a smaller chip duration will allow for closer range measurements. Velocity measurements in PMCW are based on the length of the chip sequence. The velocity coverage has a maximum unambiguous range of

$$V_{max} = \frac{c}{4f_c T_c L_c} \tag{2.9}$$

Where f_c is the carrier frequency. The number of collected sequences N being processed reduces it to

$$V_{max} = \frac{c}{4N f_c T_c L_c} \tag{2.10}$$

Velocity resolution in PMCW is purely based on the carrier frequency

$$\Delta v = \Delta f \frac{c}{2f_c} \tag{2.11}$$

Where Δf is the Doppler frequency shift.

2.2 The Future of ADAS Radar

This section will use the information from chapter 1 to decide which modulation scheme is the best for the future of ADAS radar. The ideal design has to take into account three areas. First, the solution must have ways to handle the growing problem of radar interference. Next is the ability to make both capable and efficient designs. Finally, radar is only one element in the larger ADAS environment, so there needs to be support for future growth towards full autonomy. Once a design has been selected, the remaining challenges will be identified as

the core problem to be addressed.

2.2.1 Interference Considerations

Radar interference occurs anytime the signal from one radar disrupts the detection ability of another. The range of effects from interference can include ghost targets, lines or ridges in the range-Doppler map, or a general increase in the noise floor at the receiver [29]. It depends on the type of modulation used, and the parameters within that modulation type. A worst case example would be two FMCW radars transmitting the same chirp slope, duration, and carrier frequency or two PMCW radars with the same chip sequence, rate, and frequency. In these cases there is no way to determine which signal originated from which radar. The obvious solution is then to use a variety of chirp parameters or chip sequences so that there is distinction in the signals. This is a common part of the designs for both systems, but without coordination between vehicles it is still possible for the parameters to synchronize accidentally.

For PMCW-PMCW interference, it has been shown that slight differences in the waveform parameters result in complete disruption in detection ability [30]. The range-Doppler grid in this case shows interference appearing either as a ridge that dominates an entire velocity range, or as noise that obscures all range-Doppler measurements. A similar level of impairment is shown for an FMCW-FMCW case as well [29]. These extreme cases are unlikely in practice. When comparing unrelated signal parameters for both FMCW-FMCW and PMCW-PMCW, it was shown that either combination can result in a raised noise floor level [31]. The only notable difference was that FMCW-FMCW interference resulted in a ridge-like pattern across the range-Doppler grid while PMCW-PMCW interference had uniform increase in the noise floor. This gives PMCW a slight detection advantage in the presence of unmitigated noise.

2.2.2 Hardware Efficiency

The push to improve detection capabilities in ADAS radar has made energy efficiency a concern. The current range resolution of 10 cm is still too large to separate close objects or small obstacles. That resolution corresponds to a signal bandwidth of 1.5 GHz, which is enough to limit hardware choices. For FMCW the limit is in the transmitter, where the chirp slope has to remain highly linear to preserve the range resolution [32][33]. PMCW does not have this linearity requirement.

In PMCW the limit is in the receiver, where the large signal bandwidth requires ADCs with a high sample rate. The problem here is the relationship between sampling rate and ADC resolution. Two Figures of Merit (FoM) have been established that link ADC resolution, sampling rate, and power consumption into one comparable value [34]. First is the Walden FoM that states that power consumption rises at a rate of 2x per added bit of depth. This is an older standard that described ADCs that were limited by the efficiency of their design. It is still relevant for low speed ADC designs. Newer high speed ADCs are limited by thermal noise instead, and use the Schreier FoM which shows a 4x increase in power consumption per added bit.

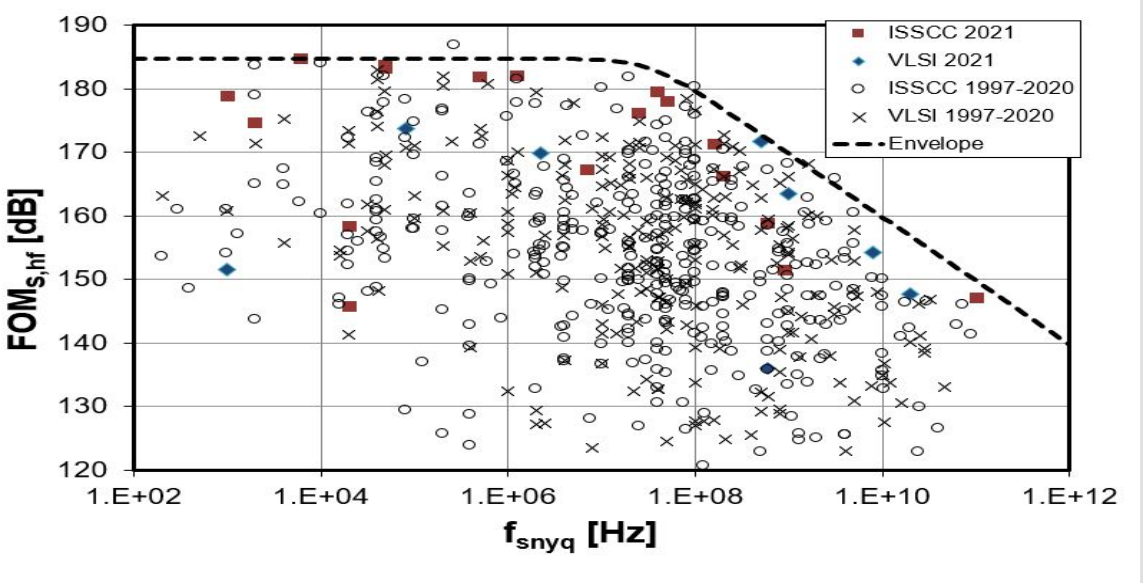

Figure 2.3: *Efficiency of ADCs by sampling rate using the Schreier FoM [34].*

ADC data compiled by Boris Murmann in Figure 2.3 shows how sampling rate affects the Schreier FoM. There is a clear drop in performance starting at a sampling rate of 100 MHz. This is far below the mmWave frequencies used in ADAS radar, which can have signal bandwidths as large as 4 GHz [4]. A compromise must be made then between the high sampling rate that is required, and the number of resolution bits that can be added while maintaining acceptable power consumption. This gives PMCW a clear advantage as long as it can maintain detection capabilities with fewer ADC resolution bits.

2.2.3 Future Growth

Autonomous cars have a long way to go before they become the standard. A big step on that journey would be cooperation between individual cars. To do this they would need a medium to communicate quickly and effectively. Radar modules have the potential to perform that role because they use the same fundamental equipment other communications devices. Joint radar-communication systems is an area of research that is trying to integrate both functionalities in one device.

This is no convenient way to do this with FMCW. The chirp waveforms cannot be modified to add communication data due to the linearity requirement. Other approaches have to be used, like isolating segments of bandwidth to dedicate to communication [35]. This means less bandwidth for both applications. The PMCW approach is already very similar to a communications system. The chip sequence can be used to encode data as long as the auto-correlation properties are preserved [36]. This is a clear advantage for PMCW.

2.3 Summary

Advantages in both hardware requirements and the potential for future development makes PMCW the clear choice for ADAS radar. The challenge then is the choice of ADC, where every added bit of sampling resolution increases power 4x. The problem can be approached

from both a hardware and software perspective. Existing detection algorithms can be explored to determine how many bits are required for adequate performance. Similarly, modifications to algorithms can be explored to determine if they can adapt to lower resolution ADCs.

CHAPTER 3

Literature Review

This chapter will review current research to find approaches to the reduced resolution ADC problem for PMCW radar. Solutions that use lower resolution ADCs will be favored based on the conclusions from chapter 2.

W. Van Thillo et al. looked at this problem by asking how many ADC bits were required to perform pedestrian detection [37]. They performed a full system simulation in MATLAB for a 79 GHz PMCW radar. The model transmitted a signal towards two targets then passed the return signal through an ADC. From there correlators estimated target range and Fast Fourier Transform (FFT) blocks estimated the Doppler shift. Their results compared outputs with no quantization noise, equivalent to an infinite-bit ADC, then with 4, 3, 2, and 1 bit quantization. The 4-bit version offered identical performance to the no quantization approach. The 3-bit and 2-bit models began to see distortion in the Doppler domain, and 1-bit model did not work at all for their model. The explanation for this was signal spillover from the transmitter to the receiver due to their close proximity. Overall this paper shows that 2-4 resolution bits could be enough PMCW radar. This was a proof-of-concept paper however, so the scope of the results is narrow. Only a single 2-target scenario is presented, and their results are analyzed through visual inspection of one output. It is not clear then how successful radar detection software can be using the outputs of this model.

The field of communications is very similar to radar, so low resolution ADC approaches

need to be considered there as well. A task very similar to radar detection is channel estimation. The goal of channel estimation is to determine how a signal changes as it travels from the transmitter to the receiver. To do this, predetermined values called pilot signals are added to the signal being sent, with a known spacing between each pilot. The receiver can then analyze an incoming signal against knowledge of the pilot signals to determine how the channel has affected the data. It is then possible to recover information that may have changed during transmission. This is analogous to the radar receiver having a known transmit signal used to classify radar returns.

The work of Jianhua Mo, Philip Schniter, and Robert Heath looks at using one-bit ADCs to perform channel estimation for mmWave systems [38]. This paper investigates how few-bit ADCs impact channel estimation capabilities. It takes advantage of group sparsity to get better estimations. ADC resolutions of 1-4 bits are considered. Their unique insight was to relate sparsity observed in delay measurements to sparsity in angle measurements to form groups. This allowed for modifications to the Approximate Message Passing (AMP) algorithm that did not require prior information about channel sparsity, somthing that had limited other sparse channel estimation approaches. Two versions of AMP are considered: Generalized AMP (GAMP) and Vector AMP (VAMP). The novel modified versions use elements from Expectation Maximization (EM) to estimate a prior on channel distribution, which normally has to be provided in normal GAMP and VAMP.

The simulations considered the Normalized Mean Squared Error (NMSE) between the real and estimated channel parameters. In a test comparing ADC bit depth and SNR, the 2, 3, 4, and infinite resolution approaches had identical performace below 10 dB of SNR. For higher SNR values each added bit showed a small improvement to NMSE before leveling off, with the infinite resolution example showing a linear improvement as SNR increased. The 1-bit approach had slightly higher error to begin with, and stopped improving above 5 dB of SNR. This shows that most applications would have no benefit from using more than 2-4 bit ADCs, and could use a 1-bit setup if some error is tolerated. Finally, comparison

plots were provided showing how the proposed algorithms compared against four existing algorithms. The modified GAMP performed similarly to the other methods, but the modified VAMP had much lower error even in the first iterations than any approach. Clearly then the choice of algorithm has a large impact on performance.

For a closer look at algorithm considerations for few-bit ADCs, the mathematics domain was explored. Ji Liu and Stephen Wright provided one of the first optimization approaches specifically for data with low resolution quantization [39]. They use an understanding of both quantization and saturation errors to develop a constrained optimization problem. First they show a standard basis pursuit model, minimizing the L1 norm of the sparse vector such that the L2 least squares error is less than noise both from the channel and quantization. A modification is proposed where the objective function is now minimizing the least squares error, with the scaled L1 norm of the signal estimate as a penalty term. A new constraint was introduced, bounding the infinity norm of the estimation error, the largest estimation error value, to be less than the quantization noise. This method was named LASSO(infinity).

The Alternating Direction Method of Multipliers (ADMM) algorithm was used to find an efficient solution. It uses an augmented Lagrangian to maintain separability while creating a convex approximation. They define constraints in separate u and v terms, representing quantization and saturation terms respectively. Each iteration alternately solves for these values and their Lagrange multipliers before finding an updated signal estimate. The signal update step requires the use of an additional algorithm, and the authors make a few suggestions including Sparse Reconstruction by Separable Approximation (SpaRSA) and the Fast Iterative Shrinkage Thresholding Algorithm (FISTA) [40]. Simulations were performed using a Gaussian random measurement matrix, and a sparse vector with random entries of random amplitude between 0 and 1. The results compare this novel approach to four similar algorithms. Each algorithm is evaluated based on the average SNR value across 30 trials for each set of test parameters. Most of the tests use a quantization bit-depth of 3-5, and in each of these tests the proposed algorithm is competitive, though all five algorithms

tend to show similar results. Most significantly, the test comparing bit depths from 2-12 shows a linear increase in SNR for every algorithm as more bits are added.

Overall this showed how to form constrained optimization approach when lower ADC resolutions were used. The need to pass a stage of this algorithm into another algorithm at every iteration, as well as the calculation and storage of the augmented Lagrangian terms makes it resource intensive. The linear relationship of performance to bit-depth also means the promising 2 and 3 bit data from the previous paper would suffer from both the hardware and software side.

Interestingly, Chao-Yi Wu et al. used FISTA as the main algorithm for few-bit signal recovery instead of nesting two algorithms together [41]. This paper was specifically trying to solve the few-bit ADC, PMCW radar problem. The original problem formulation used unconstrained optimization to define the objective function. A negative log-likelihood term was developed that determined the probability of the estimated signal vector given prior knowledge of the measured signal. Specifically, this model uses the standard normal distribution to find how well the current estimate is explained by the two thresholds that the known measurement fell between. Uniquely, this paper added a group sparse penalty term by separating the real and imaginary signal amplitudes to form real valued groups.

The use of FISTA was due to the large data sizes created by the combined range, Doppler, and angle dimensions. The algorithm was modified to use the group sparse model of the objective function. Each iteration found separate gradient values for the real and imaginary parts. The combined magnitude of change for each group was calculated for use in the signal update step. Here, each real and imaginary index was updated based on the ratio of their individual change over the combined magnitude of change. Simulations were performed to detect three targets in close and widely spaced configurations. A measurement matrix was derived from the selected range, Doppler, and angle parameters. Comparisons were made between 1 and 2 bit quantized estimates. In the widely spaced case both bit depths were able to detect the targets accurately, though significant amplitude loss was noted in the 1-bit

case. When two of the targets were close together the 1-bit version created a false-positive near the two real targets.

This paper showed that leveraging information about the measurements could result in successful detections, even with 1 and 2 bit quantization. The loss of amplitude information is a concern though, with lower SNR making accurate detections more difficult.

Another few-bit PMCW radar specific paper from Chao-Yi Wu et al. looks at a more sophisticated estimation algorithm [42]. The unique contribution is a detection approach for cases where the noise variance is unknown. To do this a modified version of Sparse Learning via Iterative Minimization (SLIM) is used. Outputs from this algorithm are then fed into the relaxation algorithm RELAX to improve results. They begin by finding the Cramer-Rao bound for 2 and 3 bit models to evaluate their results against. It is noted that this bound does not exist for the 1-bit case with unknown noise variance. The SLIM problem formulation begins with the same negative log-likelihood function used in the previous paper, with separate terms for the real and imaginary amplitudes. A sparse penalty term is added using a sub-L1 pseudo-norm. The problem is then solved using a Maximization Minimization (MM) approach, alternating between updates of the sparse vector estimate and the unknown noise variance. A closed form solution is provided for the sparse vector, but the variance is updated by passing information to another algorithm, the interior point method. Once a solution has been found, the RELAX method uses the estimated noise variance to iteratively thresholding the sparse vector estimate.

The simulation results compared the algorithm outputs to the Cramer-Rao bound of the Normalized Root Mean Squared Error (NRMSE). In each test error was reduced when additional ADC bits were added, but performance was generally similar between the 2 and 3 bit approaches. They were also able to show that the combined SLIM and RELAX approach performed better than SLIM by itself for all bit-depths. The 3-bit case performed nearly as the infinite-resolution result. In a final test, detection performance was evaluated when the dynamic range of the target amplitudes was high. 2-bit and infinite-bit estimations were

compared, and the 2-bit case showed lower estimated target amplitudes and some false positive detections. Altogether this paper showed a promising direction of signal estimation with unknown noise variance for few-bit applications. It did continue the trend of poor amplitude estimates in fewer-bit solutions, which lead to bad performance when target dynamic range was high. The need to call on a separate algorithm as a step in each iteration also makes it more resource intensive.

The amplitude estimation issues from the previous methods may come from algorithms designed for higher resolution. Christopher Gianelli, Luzhou Xu, Jian Li, and Petre Stoica looked at designing a 1-bit algorithm that would preserve signal amplitude [43]. Their paper focused on the communications problem of multi-tone parameter estimation. This is another area where high sample rates force lower-resolution ADCs to be used. The authors note that most of the research prior to theirs used a fixed threshold of zero for 1-bit quantization, and that amplitude information is unnecessarily lost because of it. As an alternative, they suggest using Time Varying Thresholds (TVT) that retain more amplitude information by leveraging the threshold level. To study this effect, a novel RELAX algorithm is introduced that is modified to use TVT.

Their signal model at the receiver keeps the sign of the signal, plus noise, minus the threshold value. A negative measurement therefore indicates that the signal amplitude was lower than the threshold amplitude. The threshold values are then used as prior information for the estimation algorithm. The likelihood function used for optimization finds the difference between the current signal estimate and the known threshold, then multiplies by the sign from the corresponding measurement value. The novel 1-bit RELAX algorithm takes a greedy optimization approach. It finds the single strongest signal index to update, and improves the estimate of that signal until it converges. It then finds the next strongest signal index and it updates it and the previous index until convergence. Each new index increases the dimension of the problem by 1. The negative log-likelihood version of the defined likelihood function is used, and two versions are provided to show how to add

dimensions for each major iteration.

The simulation used 8 random TVT values per run, with values between -1 and 1. Six signals were used, each with amplitudes set at 1. Results were averaged over 100 trials for 1-bit RELAX and three similar likelihood based approaches. It was shown that 1-bit RELAX outperformed other methods at both frequency and amplitude estimation, based on the average MSE. Overall this paper showed that an algorithm designed to match the limitations of 1-bit quantization performs better than more general approaches.

The TVT approach has been applied to radar by Xiaolei Shang, Heng Zhu, and Jian Li [44]. This paper looked at range-Doppler target estimation with unknown noise variance and a 1-bit specific approach. Their model used the same 1-bit negative log-likelihood function as the previous paper. Similar to the FISTA paper, they also separate the real and imaginary signal components, but they do not use them to create a group sparse problem. To promote sparsity, they use a log norm penalty term, which tends to favor sparser solutions. Their model is not convex, so they use the MM algorithm to solve it. A second order Taylor series is used to upper bound the objective function, which is minimized at each iteration to find the new signal estimate. In order to be solved efficiently, two values in the algorithm have to be found by separate instances of Conjugate Gradient Least Squares (CGLS).

For simulation parameters, a range-Doppler measurement matrix formulation is provided. Their sparse signal vector has 20 randomly placed targets with random amplitudes between 0 and 1. Measurements are quantized by keeping the sign after subtracting one of eight TVT values from the full measurements. The thresholds are selected randomly from a set of eight values between -1 and 1. Noise variance is set for 10 dB SNR. With this setup the results show range-Doppler plots of their algorithm compared to two matched filter estimations, one with infinite precision the other using a 1-bit ADC. The matched filter approaches show large sidelobes across the range bins around most of the targets. A traditional matched filter does nothing to prevent sidelobes on its own, and no additional processing was indicated, so this is normal. The proposed algorithm does not show these sidelobes and was more

accurate with the signal amplitudes. The final test shows that for SNR values between -10 and 10 the algorithm converges in less than 15 iterations.

In summary, this paper provided useful insight into how TVTs can be added into a radar detection algorithm. It also showed a slightly different MM technique to signal estimation with unknown noise variance to the SLIM-RELAX algorithm above. Just as in that method, each iteration requires an additional algorithm to compute values, which could be resource heavy. The results again suggest that a TVT approach provides good amplitude estimates. It is hard to say how this algorithm performs in the larger scheme of sparse, few-bit radar detection algorithms though. It was not compared against any other algorithms, only a matched filter output. This might be due to a lack of comparable algorithms in PMCW radar currently. Finally, it would have been good to include plots showing the average estimation error rate under different SNR values.

Finally, Xiaolei Shang, Jian Li, and Petre Stoica provided a way to adapt a whole class of algorithms to a 1-bit form [45]. Sparse Iterative Covariance-based Estimation (SPICE) algorithms are valued because they require no parameter tuning. An example of such a parameter from the previous papers is in the sparse penalty terms. These terms are scaled by a sparse regularization parameter, a coefficient that determines the level of sparsity the algorithm will favor. This paper presented four 1-bit specific algorithms: 1bSPICE, 1bLIKES, 1bIAA, and 1bSLIM. The original versions of each of these algorithms work through different minimizations using the covariance matrix for the sparse measurements. The main difference between them is unique weighted sparse penalty terms in their objective function. To tie them together, the authors adapt the MM approach from the previous paper, which included a 1-bit TVT specific negative log-likelihood definition. The result is an approach very similar to the SLIM-RELAX algorithm [42], but with an added step to update the weights for 1bSPICE, 1bLIKES, and 1bIAA.

To simulate the performance of these new adaptations, a range-Doppler sparse measurement matrix is defined, eight random TVT thresholds are selected based on the maximum

signal amplitude, and the noise variance is set for 20 dB of SNR. For comparison, the ADMM algorithm and a 1-bit periodogram (1bPER) were included. Both FMCW and PMCW radar detection examples are provided. In an average of 500 FMCW test runs, the 1bSLIM, 1bLIKES, and 1bIAA had the best NMSE as SNR was swept from 0-25 dB. The sidelobe power across these runs was also measured, and 1bSLIM performed significantly better than the other approaches for all SNR values. In CPU computation time measurements, the four proposed algorithms performed the best, with 1bSLIM running the fastest. In the PMCW test range-Doppler plots were provided for each algorithm. The new methods performed well, but 1bSPICE had false positives and 1bSLIM missed a weaker target. 1bIAA performed the best in this test.

The final experiment returned to FMCW, showing range-Doppler plots of each algorithm. Two additional plots showed the result of FFT measurements and conventional IAA with high-resolution measurements. Again the proposed methods perform well, and it is noted that 1bIAA performs almost the same as high-resolution IAA. A lot of work was covered in this paper. Ultimately it showed that 1-bit specific algorithms can perform well for both FMCW and PMCW applications. It was also the only known paper to compare 1-bit adapted algorithms against each other. The inclusion of 1bPER, ADMM, and conventional IAA approaches was also good, though it would be nice to see a comparison with few-bit IAA along with high-resolution.

The trends from the literature can be summarized like this:

1) In all reviewed papers, there was never a benefit to using more than 4-bit ADC resolution outside of the highest SNR environments. In many cases 2 and 3-bit resolution was enough.

2) 1-bit estimation is possible, but generally performs worse than higher bit-depths in a more general few-bit algorithm. A clear loss of amplitude information is shown in the 1-bit case.

3) The use of time-varying quantization thresholds and 1-bit specific algorithms can

capture more amplitude information. This is the main solution found to improve 1-bit detection capabilities.

4) While several papers introduced novel few and 1-bit algorithms, very little work has directly compared their performance. When comparisons were made, it was not between a 1-bit specific TVT approach and its few-bit equivalent. No research was found that would indicate when a 1-bit TVT approach would be a better choice than 2-bit without TVT, for example.

5) Radar signals can be recovered without prior noise variance information, but the current techniques require multiple algorithms combined to be computationally efficient. For larger data sets, it would be good to know how this affects resource usage such as memory and processing time.

CHAPTER 4

Proposed Solution

The analysis from the previous chapter showed that signals could be recovered in the 1-bit quantization case, but algorithms not adapted for 1-bit quantization lose amplitude information normally provided by having multiple quantization thresholds. It was shown that a TVT adaptation allows this amplitude information to still be estimated using a 1-bit ADC. A lack of work comparing few-bit and 1-bit specific approaches was apparent. The novel contribution of this thesis is to adapt a few-bit FISTA implementation [41] into a 1-bit specific TVT FISTA using a TVT-based negative log-likelihood function [43][45]. This approach will be called FISTA-1b. Test results from both the TVT and original few-bit methods will be directly compared in the next chapter.

The rest of this chapter will be as follows: First the original group-sparse FISTA model will be provided. Next, the changes necessary for a 1-bit specific modification will be provided. Finally the evaluation criteria will be specified to allow for quantitative analysis of both methods.

4.1 Problem Formulation

In the reviewed FISTA approach for PMCW radar, a useful technique was shown that created a group sparse problem out of separated real and imaginary signal components. This allowed

the estimation algorithm to be relatively simple compared to other approaches that combine two or three algorithms together to be solved efficiently. Unfortunately their approach was too general for the 1-bit case. Their results showed that target amplitudes were not preserved, which lowers the SNR for target detection and may have caused the poor performance when targets were closely spaced. Even the more complex few-bit SLIM-RELAX algorithm suffered amplitude-related dynamic range issues. Few and 1-bit algorithms need to be designed to preserve signal amplitude if they are to address these issues. This section will show how to modify FISTA using the techniques shown in the development of 1-bit SPICE algorithms.

4.1.1 Taking Sparse Measurements

To begin, a sparse vector $\mathrm{x} \in \mathbb{C}^N$ is defined as the collection of K target signals out of N measurement locations, where $K \ll N$. The values of x are then given by

$$\mathrm{x}_i = \begin{cases} \gamma_k & : \mathrm{x}_i = k \in K \\ 0 & : \mathrm{x}_i \neq K_i \end{cases} \tag{4.1}$$

Where γ_k is the complex amplitude of the k^{th} target. Measurements are taken using a measurement matrix $A \in \mathbb{C}^{M \times N}$, where M corresponds to the number of samples collected from each location N. The final measurement vector is then found by

$$y = A\mathrm{x} + e \tag{4.2}$$

Where $e \in \mathbb{C}^M$ is additive white Gaussian noise. These measurements need to be converted to an all-real equivalent before they can be quantized. The new form of the measurements becomes

$$\tilde{y} = \tilde{A}\tilde{\mathrm{x}} + \tilde{e} \tag{4.3}$$

Where real and imaginary magnitudes are separated by

$$\begin{aligned} \tilde{y} &= \begin{bmatrix} \Re\{y\} \\ \Im\{y\} \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} \Re\{A\} & -\Im\{A\} \\ \Im\{A\} & \Re\{A\} \end{bmatrix}, \\ \tilde{\mathrm{x}} &= \begin{bmatrix} \Re\{\mathrm{x}\} \\ \Im\{\mathrm{x}\} \end{bmatrix}, \quad \tilde{e} = \begin{bmatrix} \Re\{e\} \\ \Im\{e\} \end{bmatrix} \end{aligned} \tag{4.4}$$

Now that the data is all-real, it can be quantized by passing through a q-bit resolution ADC. This process is different between the few-bit and 1-bit TVT cases. The few-bit case will be shown first. The measurements now take the form

$$\bar{v} = Q(v) \qquad when\ v \in [u_d, u_{d+1}) \tag{4.5}$$

With u_d as the threshold immediately below v, and u_{d+1} the threshold immediately above. Given q resolution bits, there are $D = 2^q + 1$ thresholds forming the partition and 2^q output values for the codebook. The outermost thresholds take the values $u_1 = -\infty$ and $u_D = \infty$. The step size Δs is set using the largest measurable amplitude R_{max} to get $\Delta s = R_{max}/2^q$. In practice the value R_{max} would be found using automatic gain control. The remaining thresholds can then be calculated with

$$u_d = (-2^{q-1} + d)\Delta s, \qquad d = 1, ..., 2^q - 1 \tag{4.6}$$

With the partition thresholds set, the quantization codebook values are offset by half the step size

$$\begin{aligned} \bar{v}_0 &= -(2^{q-1} - 2^{-1})\Delta s \\ \bar{v}_d &= u_d + \frac{\Delta s}{2}, \qquad d = 1, ..., 2^q - 1 \end{aligned} \tag{4.7}$$

The 1-bit TVT approach to quantization begins by establishing the TVT vector $h \in \mathbb{C}^M$. Its values are defined by $h = h_R + jh_I$, where real and imaginary thresholds $h_R, h_I \in [u_1, u_D)$ are drawn from D uniformly distributed discrete values. The all-real arrangement of h is then

$$\tilde{h} = \begin{bmatrix} \Re\{h\} \\ \Im\{h\} \end{bmatrix} \tag{4.8}$$

Finally, the quantized values are found by

$$\bar{y} = sign(\tilde{y} - \tilde{h}) \tag{4.9}$$

Where $sign()$ takes values [-1, 1]. Both quantization methods are now established.

4.1.2 Few-bit FISTA Model

The original few-bit specific FISTA had two main model components. First was the negative log-likelihood function, which was defined as

$$L(\tilde{\mathrm{x}}|\bar{y}) = -\sum_{m=1}^{2M} ln\left[\Phi\left(\frac{u_{d+1} - \tilde{a}_m^T\tilde{\mathrm{x}}}{\sigma/\sqrt{2}}\right) - \Phi\left(\frac{u_d - \tilde{a}_m^T\tilde{\mathrm{x}}}{\sigma/\sqrt{2}}\right)\right] \tag{4.10}$$

Where $\tilde{a}_l^T\,\tilde{\mathrm{x}}$ a single value in the vector of measurements, $\tilde{y}_l$. The noise standard deviation is given by σ, and the cumulative distribution function (CDF) of the standard normal distribution is given by $\Phi()$. This function essentially finds the probability of the signal values given prior knowledge of the two thresholds the resulting measurements fell between.

The other component to this model is a group-sparsity promoting penalty term. Groups are formed from the real and imaginary magnitudes of x, therefore the penalty term is defined

$$\|\tilde{\mathrm{x}}\|_{2,1} = \sum_{g=1}^{G} (\tilde{\mathrm{x}}_g^2 + \tilde{\mathrm{x}}_{g+G}^2)^{1/2} = \sum_{g=1}^{G} \|\tilde{\mathrm{x}}_g\|_2 \tag{4.11}$$

Where $G = N$ groups of $\tilde{\mathrm{x}}_g = [\tilde{\mathrm{x}}_g,\ \tilde{\mathrm{x}}_{g+G}]$ are created. The objective function is the minimization of those two terms, given by

$$\min_{\tilde{\mathrm{x}}} \ F(\tilde{\mathrm{x}}) = L(\tilde{\mathrm{x}}|\bar{y}) + \rho\|\tilde{\mathrm{x}}\|_{2,1} \tag{4.12}$$

Where ρ is the sparse regularization coefficient that determines the level of sparsity that will be promoted. To solve this problem efficiently a quadratic approximation is formed by

$$\hat{\tilde{\mathrm{x}}}^{\,i} = \arg\min_{\tilde{\mathrm{x}}} \left\{ \sum_{g=1}^{G} \frac{\|\tilde{\mathrm{x}}_g - h(\mathrm{z})_g\|_2^2}{2\zeta} + \rho\|\tilde{\mathrm{x}}_g\|_2 \right\} \tag{4.13}$$

where $\hat{\tilde{x}}^{\,i}$ is the signal estimate for iteration i, $\mathrm{z} \in \mathbb{R}^{2G}$ is a FISTA specific scaled version of $\hat{\tilde{\mathrm{x}}}^{\,i}$ that will be defined later, ζ is the step size, and $h(\mathrm{z})_g = \mathrm{z} - \zeta\nabla L(\mathrm{z}|\bar{y})$. The gradient $\nabla L(\mathrm{z}|\bar{y})$ is defined

$$\nabla L(\mathrm{z}|\bar{y}) = \frac{\partial L(\mathrm{z}|\bar{y})}{\partial \mathrm{z}} = \frac{\sqrt{2}}{\sigma}\tilde{A}^T f'(\mathrm{z}) \tag{4.14}$$

Where $f'(\mathrm{z})$ is

$$f'(\mathrm{z}) = \frac{\phi\left(\frac{U_{d+1} - \tilde{A}\mathrm{z}}{\sigma/\sqrt{2}}\right) - \phi\left(\frac{U_d - \tilde{A}\mathrm{z}}{\sigma/\sqrt{2}}\right)}{\Phi\left(\frac{U_{d+1} - \tilde{A}\mathrm{z}}{\sigma/\sqrt{2}}\right) - \Phi\left(\frac{U_d - \tilde{A}\mathrm{z}}{\sigma/\sqrt{2}}\right)} \tag{4.15}$$

which is the ratio of the standard normal probability density function (PDF) to the CDF for the current value of z. The variables U_{d+1} and U_d are vectorized versions of the threshold priors based on $\bar{y}$. The groups are now leveraged by finding the combined magnitude of change for this iteration, given by

$$\|h(\mathrm{z})_g\|_2 = \sqrt{h(\mathrm{z})_g^2 + h(\mathrm{z})_{g+G}^2} \tag{4.16}$$

The signal estimate $\hat{\hat{\mathrm{x}}}^{\,i}$ calculation is separable, so a parallel shrinkage operation is performed to update it. The amount of shrinkage is scaled using the group magnitudes as follows

$$\hat{\hat{\mathrm{x}}}_g^{\,i} = \frac{h(\mathrm{z})_g}{\|h(\mathrm{z})_g\|_2} \max(0, \|h(\mathrm{z})_g\|_2 - \rho\zeta), \quad g = 1, ..., G \tag{4.17}$$

The last two steps are the reason for the "Fast" part of FISTA. A variable η is introduced that is used accelerate updates to z as the number of elapsed iterations grows. The two are updated by

$$\eta_{i+1} = \frac{1 + \sqrt{1 + 4\eta_i^2}}{2}, \tag{4.18}$$

$$\mathrm{z}^{i+1} = \hat{\hat{\mathrm{x}}}^{\,i} + \left(\frac{\eta_i - 1}{\eta_{i+1}}\right)(\hat{\hat{\mathrm{x}}}^{\,i} - \hat{\hat{\mathrm{x}}}^{\,i-1}) \tag{4.19}$$

The update of z takes the current signal estimate and adds to it a fraction of the change that occurred from the previous estimate. The ratio of the η terms starts at zero then grows with each iteration, allowing the difference in signals between iterations to have more effect over time. This completes the FISTA algorithm, which is summarized in Algorithm 1.

Algorithm 1 FISTA

Input: $\rho, \zeta, i = 1, \eta_1 = 1, \mathrm{z}^1 = \hat{\hat{\mathrm{x}}}^{\,1} = \mathbf{0}_{2G}$
Output: $\hat{\hat{\mathrm{x}}}$

1: **repeat**
2: Solve for $\hat{\hat{\mathrm{x}}}^{\,i+1}$ using (4.17)
3: Compute η_{i+1}
4: Compute z^{i+1}
5: $i \leftarrow i + 1$
6: **until** max iterations or convergence

Output: $\hat{\hat{\mathrm{x}}}$

4.1.3 Proposed FISTA-1b

The model for FISTA-1b requires few changes compared to the few-bit model. The objective function needs to be rewritten first to account for the use of TVT. For this the negative log-likelihood function from [44][45] is modified in two ways. First, the noise standard deviation is assumed to be known. Second, all complex values have been separated to create all-real data to match the required inputs to FISTA. The result is

$$L(\tilde{\mathrm{x}}|\bar{y}) = -\sum_{m=1}^{2M} ln\left(\Phi\left(\bar{y}_m \frac{\tilde{a}_m^T \tilde{\mathrm{x}} - \tilde{h}_m}{\sigma/\sqrt{2}}\right)\right) \tag{4.20}$$

This function works by estimating the probability that the sign of the measurements has been flipped. The same sparse penalty term can be used, which means the objective function (4.11) takes the same form. The same separable quadratic approximation is used, but now the new likelihood function is used when computing the gradient. The new gradient is given by

$$\nabla L(\mathrm{z}|\bar{y}) = \frac{\partial L(\mathrm{z}|\bar{y})}{\partial \mathrm{z}} = \frac{\sqrt{2}}{\sigma}\tilde{A}^T f'(\mathrm{z}) \tag{4.21}$$

Where $f'(\mathrm{z})$ is

$$f'(\mathrm{z}) = -\frac{\phi\left(\bar{y}\frac{\tilde{A}\mathrm{z}-\tilde{h}}{\sigma/\sqrt{2}}\right)}{\Phi\left(\bar{y}\frac{\tilde{A}\mathrm{z}-\tilde{h}}{\sigma/\sqrt{2}}\right)} \tag{4.22}$$

Because the gradient calculation is the only functional change, the new approach uses the same steps listed in Algorithm 1.

4.2 Evaluation Criteria

The experiments performed will focus on three approaches. They are the 2-bit FISTA, 1-bit FISTA, and finally FISTA-1b. There are a four metrics to define in order to provide a

fair evaluation. The first is the SNR measurement, which is based on the strongest signal magnitude and the noise standard deviation. It is given by

$$SNR = 20Log\left(\frac{\max(|\gamma_k|)}{\sigma}\right) \tag{4.23}$$

The use of the maximum signal amplitude here provides a higher SNR value than an average signal amplitude calculation would. This should therefore be considered an upper bound on SNR. In all tests there will be high and low SNR examples included to provide a complete depiction of functionality. The second metric is the NMSE measurement, which will compare the measured amplitude at each target location to the actual amplitude. It is defined

$$NMSE = \frac{1}{250}\sum_{k=1}^{K}\sum_{i=1}^{50}\frac{|\hat{\gamma}_k^i - \gamma_k|^2}{\gamma_k^2} \tag{4.24}$$

Together these two will be used to determine how accurately the target amplitudes are estimated as the noise level is varied. For theses tests the priority is the amplitude estimation accuracy. The calculation will include the amplitudes of missed targets since a missed detection is another form of underestimation. This creates an upper bound to the amount of error contributed by each target.

The estimated amplitude is always less important than overall detection accuracy in radar applications. The last two metrics represent target accuracy by determining the rates of true detections and false positives. The combination of the two rates are represented by a Receiver Operating Characteristic (ROC) curve. They are calculated by

$$\begin{aligned} TPR &= \frac{TP}{TP+FN} \\ FPR &= \frac{FP}{FP+TN} \end{aligned} \tag{4.25}$$

Where TP is the number of true positives, or correctly identified targets, FN is the

number of false negatives, or missed detections, FP is the number of false positives, or detections where no target is present, and TN is the number of true negatives, or locations correctly identified as having no targets. This results in the true positive rate (TPR) and false positive rate (FPR), which represent the relative likelihood of both conditions. These rates are calculated while sweeping a threshold for noise rejection across the signal estimate. Signals above the noise threshold are counted as detections, while all signals below are classified as noise. Once the rates are found for all threshold values they can be plotted as ROC curves. The area under the curve (AUC) will be found for each ROC plot, which is used as a way to compare the relative rates of each method. An AUC of 1 indicates that all targets can be found with no false positives. Any AUC of 0.5 or lower indicates that the algorithm has worse detection ability than random guessing.

Within the scope of radar signal estimation, false positives come from noise peaks large enough to be counted as detections. In a traditional radar setting there is always some level of noise measurable at every possible target location. This is not true for the sparse signal processing approach used in this thesis. Almost all signal indices with low measurement contributions have their values reduced to zero during signal recovery. This creates a large number of true negatives with some possible false negatives. The large number of true negatives in a sparse signal distorts the false positive rate, so for these tests any signal set to zero will not be included in the true negatives calculation. This introduces the possibility of a FPR of zero for all thresholds if all noise is rejected, which is not typical in radar estimation. In these cases, the AUC will be approximated by the largest TPR value measured. This is a lower bound on the AUC, which is always increasing with FPR.

CHAPTER 5

Simulation and Analysis

This chapter contains analysis of the algorithms presented in section 4. Each of the following sections will present the test that was performed, the testing parameters, and the results obtained. All results were obtained through MATLAB simulations. Section 5.1 will test widely spaced targets, 5.2 will test closely spaced targets, and 5.3 will test targets with a large dynamic range. Finally, section 5.4 will draw conclusions based on the the results and note areas in need of future work.

5.1 Widely Spaced Targets

The goal of this test was to determine how accurately targets were detected under ideal conditions. The targets had similar amplitudes and wide spacing, with the goal that no single target would bias the ability to detect the others.

5.1.1 Test Configuration

A measurement matrix $A \in \mathbb{C}^{M \times N}$ was formed from Gaussian random values. The dimensions of A were chosen as $M = 1016$ and $N = 4064$. The sparse signal vector had $K = 5$ targets with their locations and amplitudes given in table 4.1. The FISTA iteration step size was set to 10^{-6}. The two versions approached solutions at different rates, so they were

tuned differently. The original FISTA used $\rho = 60$ and stopped after 1000 iterations or when a minimum change of 10^{-6} was reached. FISTA-1b used $\rho = 150$ and stopped after 100 iterations or when a minimum change of 10^{-5} was reached. The NMSE was gathered to determine the accuracy of the amplitude estimates as the SNR was increased from -30 to 50 dB. ROC curves were produced to determine the detection accuracy of each method. All plots show data averaged over 50 trials.

	Index	**Amplitude**
Target 1	2	$2.5e^{j0.7821}$
Target 2	1400	$3e^{-j0.349}$
Target 3	2500	$2.5e^{j0.7821}$
Target 4	3250	$2.5e^{j0.7821}$
Target 5	3750	$3e^{-j0.349}$

Table 5.1: *Target parameters for widely spaced tests*

5.1.2 Results

A typical output for widely spaced target detection can be seen in Figure 5.1. It shows the complex magnitudes of the signal values for each method, as well as the original target signal. A few things stand out; First, it is clear that 1-bit FISTA retained far less signal amplitude than either of the other approaches. Second, the 2-bit FISTA and FISTA-1b estimates both have targets that were confidently identified, and some that were not. The second and fifth targets were poorly estimated by FISTA-1b, while the fifth target was low for 2-bit FISTA. This shows that all of the methods tested have the potential for estimation errors.

There is also noise present for all three estimates. In each case it is proportional to the overall detection amplitude. The 1-bit FISTA estimate shows very small noise amplitude, for example. The noise detected by 2-bit FISTA is large enough that it would mask the detection of the fifth target. FISTA-1b had relatively few indices with noise signals.

The NMSE data showing the accuracy of the amplitude measurements can be seen in

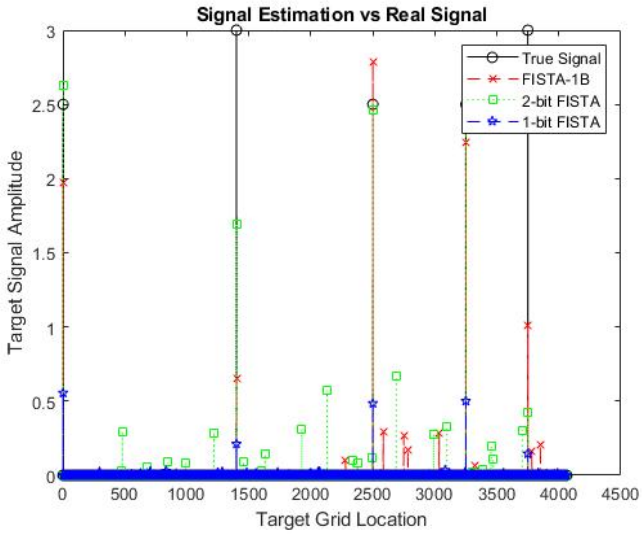

Figure 5.1: *Sample detection output for widely spaced targets*

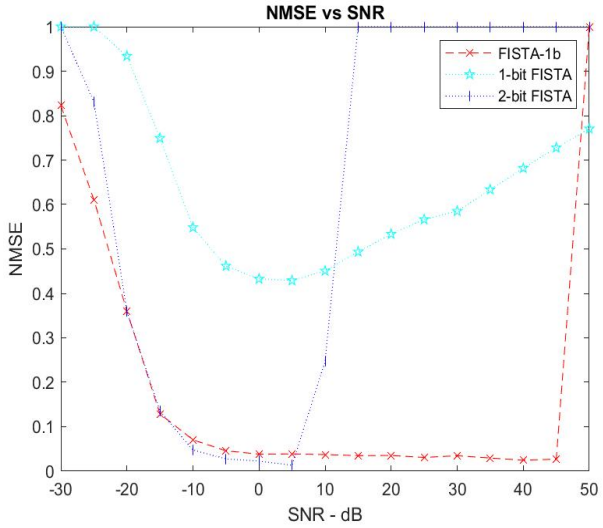

Figure 5.2: *Average NMSE for widely spaced targets over 50 trials*

Figure 5.2. As expected, the 1-bit FISTA approach had much higher error levels regardless of noise. The lowest error rate was 42% between the 0 and 5 dB SNR levels, with error rising steadily in both directions from there. The other two approaches had error rates that decreased quickly until a plateau from -10 to 5 dB. FISTA-1b continued at the same error rate of 3.8% until 45 dB, where it stopped functioning abruptly. 2-bit FISTA had the narrowest working range of the three, with performance breaking down sharply above 5 dB. Before that it reached a minimum error of 1.3%. A summary of the NMSE values is

provided in Table 5.2.

There are a few trends on this plot that stand out. First, the error rates of both FISTA-1b and 2-bit FISTA converge at a similar minimum level near 0.04. This suggests that there is a lower limit to the noise level that needs to be identified. Tests with higher quantization levels would also need to be performed to see if this limit is consistent for all bit-depths. Next, 1-bit FISTA has an error rate that rises slowly after 5 dB SNR. This shows that the additive channel noise dominates performance below that point. Additive noise might actually increase the amplitude estimates when the noise power is large enough, reducing the error. Something else is causing the error increase above 5 dB, possibly quantization noise. Finally, the 1-bit approaches are much less sensitive to changes in noise level, with much wider functional SNR ranges than 2-bit FISTA. These trends are all good topics for future research.

	FISTA-1b	**1-bit FISTA**	**2-bit FISTA**
Min NMSE	2.5%	42.8%	1.3%
SNR at Min	40 dB	5 dB	5 dB
Optimal Range	55 dB	10 dB	15 dB
SNR Limits	[NA,45]	[-20,NA]	[-25,10]
Total SNR Range	75<	70<	35

Table 5.2: *Summary of NMSE data for widely spaced targets. Optimal range is the width in dB where error rates are within* 10% *of min NMSE. Total SNR range is the width in dB that an algorithm is able to make detections.*

Finally, three ROC curves are shown in Figure 5.3. These were captured at different SNR levels to see how well the detection accuracy related to trends in the NMSE plot. The plot in the top left was captured at an SNR of -20 dB, which is the point that 2-bit FISTA and FISTA-1b converge. In the top right the curves were found at 0 dB SNR, one of two points where the NMSE was lowest for all three approaches. The bottom center shows results at 40 dB SNR, after 2-bit FISTA had broke down and shortly before FISTA-1b did.

The two original FISTA methods detected targets without false positives at -20 dB, indicating that all noise was being eliminated during estimation. Some real targets were

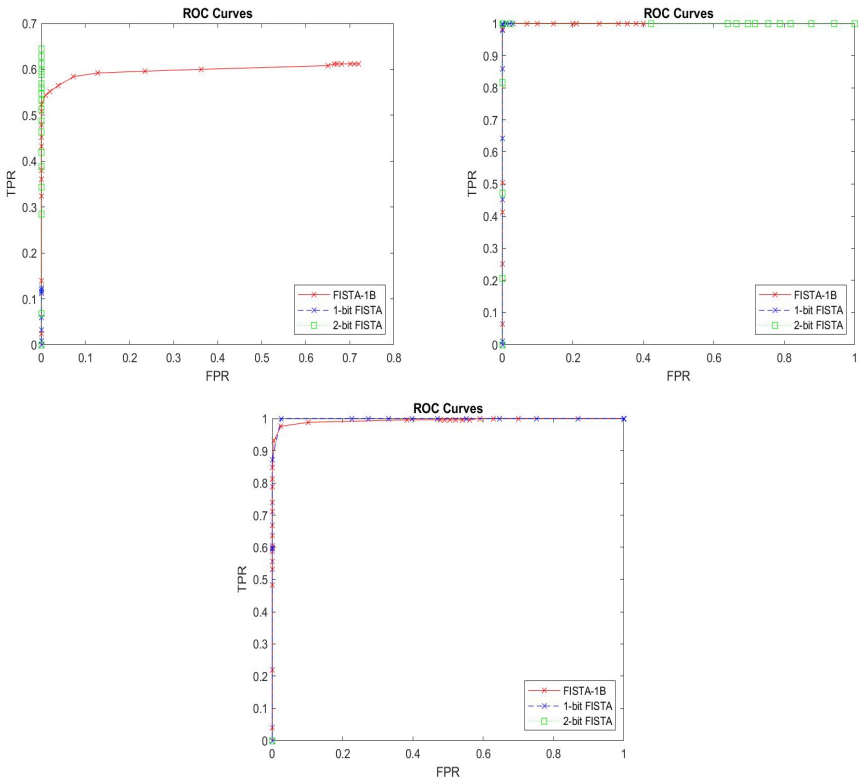

Figure 5.3: *ROC curves for widely spaced targets. SNR levels are -20 dB top left, 0 dB top right, 40 dB bottom.*

being eliminated as well, with 2-bit FISTA finding 65% of targets and 1-bit FISTA only finding 15%. Adjusting the sparse regularization term might allow for more detections for SNR this low, at the risk of introducing false positives. The curve for FISTA-1b shows that some false positives are admitted before reaching a TPR of 0.6, with only 50% detection without false positives.

The 0 dB SNR case shows perfect detections for all three methods, making it the best case scenario. The only distinguishing factor is the amplitude of smallest false positives. Both 1-bit approaches only make it partway across the top of the graph, which means that most of the ROC thresholds were larger than the noise signals. The precision of MATLAB data means that small values functionally equal to zero are still represented by positive decimal values. Finally, in the third plot for 40 dB SNR there is near identical performance from the two 1-bit methods, with 1-bit FISTA detecting all targets with fewer false positives

than FISTA-1b.

	FISTA-1b	**1-bit FISTA**	**2-bit FISTA**
-20 dB	0.58	0.15	0.65
0 dB	1	1	1
40 dB	0.99	0.99	0
Average	0.86	0.71	0.55

Table 5.3: *AUC of the ROC curves for widely spaced targets*

Finally, the AUC calculations for each test is shown in Table 5.3. As expected based on the NMSE and ROC curves, 1-bit FISTA does not work effectively at the low -20 dB SNR. The other two methods have poor rates but are still over 50%, showing that they are still functional. There is only a 1% loss observed for the 1-bit approaches from 0 to 40 dB.

5.2 Closely Spaced Targets

The goal of this test was to see if detection errors were more likely when two targets were close together. It would be noteworthy if the number of false positives increased, or if fewer targets were detected.

5.2.1 Test Configuration

The only change for this test was the locations of the targets within the sparse signal vector. To get a better understanding of the impact of proximity, two different target pairs were formed. The first pair had a separation of 15 and the second pair had a separation of 1. The target parameters are provided in table 5.2.

5.2.2 Results

The initial tests of the estimation with closely spaced targets showed that performance did not change much compared to the widely spaced case. In Figure 5.4 the detections at 10 dB

	Index	Amplitude
Target 1	2	$2.5e^{j0.7821}$
Target 2	2485	$3e^{-j0.349}$
Target 3	2500	$2.5e^{j0.7821}$
Target 4	3250	$2.5e^{j0.7821}$
Target 5	3251	$3e^{-j0.349}$

Table 5.4: *Target parameters for closely spaced tests*

show the same trends as the previous section. When taking a closer look at the two pairs, shown in Figure 5.5, it is clear that there are no false detections made in this case.

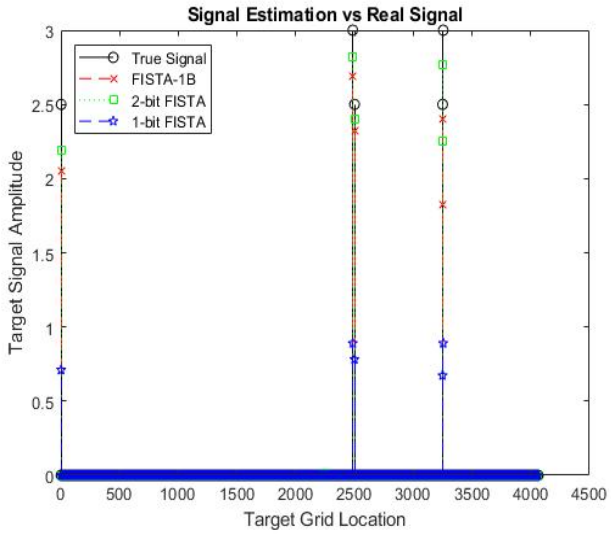

Figure 5.4: *Sample estimates for closely spaced targets at 10 dB*

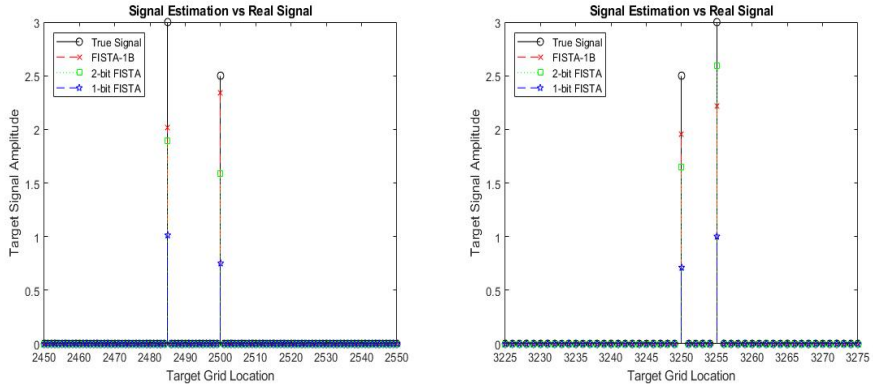

Figure 5.5: *Close up view of the close targets*

It's possible that the lack of change is due to the method of taking measurements.

Because the measurement matrix is composed of Gaussian random entries instead of entries constructed by real signal measurements, there is a lack of inter-symbol interference between adjacent indices. Revisiting this test with full PMCW radar signal simulation would be a good option for future research.

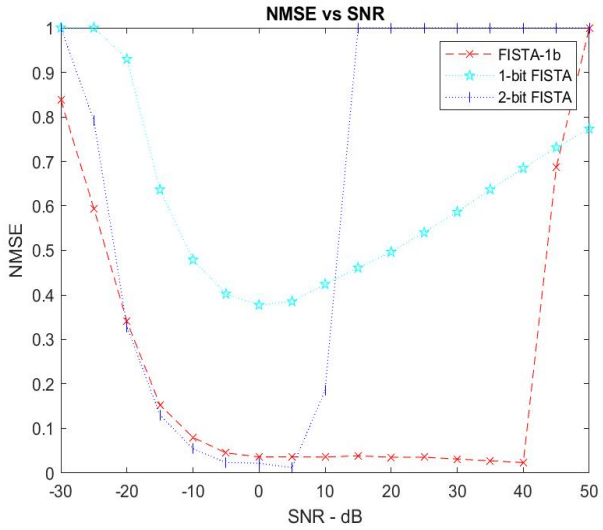

Figure 5.6: *Average NMSE for closely spaced targets over 50 trials*

When looking at the NMSE for amplitude estimation, the results are fairly similar to the widely spaced tests. The most noticeable change is that FISTA-1b stops detecting at 45 dB instead of 50 dB SNR. It has a lower minimum error level of 2.3% The 2-bit FISTA error is exactly the same as before at 1.3%, indicating that close proximity between targets is not affecting it. There is still a sharp breakdown above 10 dB. 1-bit FISTA has a lower error minimum of 37% at 5 dB. It has a linear increase in error as SNR increases from there, and does not break down at high SNR. This shows that 1-bit FISTA has predictable results, even if the estimation error is high. A summary of the NMSE data is provided in Table 5.5.

The ROC curves are shown in Figure 5.7. The three SNR values tested remained the same due to the similarity in the NMSE graphs. In the top left the curves are measured at -20 dB SNR. All three curves match the results from the widely spaced test, with only a slight increase in TPR from 1-bit FISTA. At 0 dB SNR, shown in the top right, FISTA-1b is no longer detecting all targets without false positives. The plot on the bottom is taken at

	FISTA-1b	1-bit FISTA	2-bit FISTA
Min NMSE	2.3%	37.7%	1.2%
SNR at Min	40 dB	0 dB	5 dB
Optimal Range	50 dB	25 dB	15 dB
SNR Limits	[NA,40]	[-20,NA]	[-25,10]
Total SNR Range	70<	70<	35

Table 5.5: *Summary of NMSE data for closely spaced targets. Optimal range is the width in dB where error rates are within* 10% *of min NMSE. Total SNR range is the width in dB that an algorithm is able to make detections.*

40 dB SNR, which again matches the widely spaced equivalent. The AUC values show the same results, and are shown below in Table 5.6. Overall the closely spaced target scenario only affected FISTA-1b at mid-range SNR.

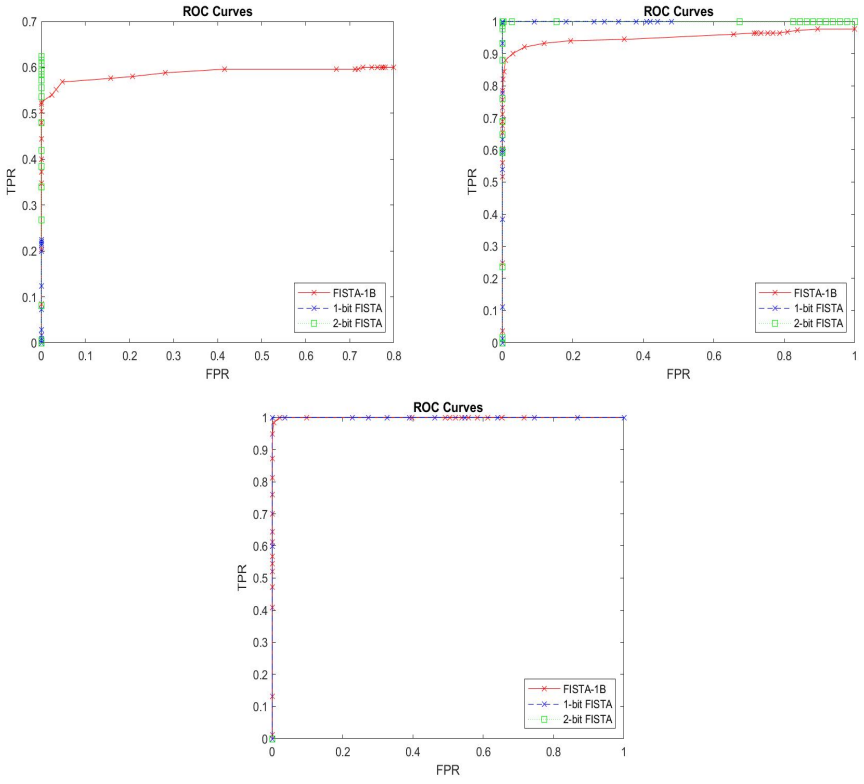

Figure 5.7: *ROC curves for closely spaced targets. SNR levels are -20 dB top left, 0 dB top right, 40 dB bottom.*

	FISTA-1b	1-bit FISTA	2-bit FISTA
-20 dB	0.57	0.24	0.63
0 dB	0.95	1	1
40 dB	0.99	0.99	0
Average	0.84	0.74	0.54

Table 5.6: *AUC of the ROC curves for closely spaced targets*

5.3 Targets With Large Dynamic Range

The goal of this test was to see if targets with the amplitudes from previous tests were still detected when a single large amplitude signal was present. In order to test this, one of the five targets was given a much larger amplitude than the others. This difference in amplitudes is referred to as dynamic range (DR), and it can be measured in decibels using

$$DR = 20Log\left(\frac{max(|\gamma_k|)}{min(|\gamma_k|)}\right) \tag{5.1}$$

5.3.1 Test Configuration

The fourth target was assigned the higher amplitudes for this set of tests. Three different DR values were tested: 6, 12, and 15 dB. These correspond to a ratio of 2:1, 4:1, and 6:1 between the largest and smallest signal amplitudes. The algorithms were more sensitive to noise in these tests, so the low SNR ranges were limited to -10 and -5 dB for the 12 dB and 15 dB dynamic ranges, respectively.

	Index	Amplitude
Target 1	2	$2.5e^{j0.7821}$
Target 2	1400	$3e^{-j0.349}$
Target 3	2500	$2.5e^{j0.7821}$
Target 4	3250	$[5,10,15] * e^{j0.7854}$
Target 5	3750	$3e^{-j0.349}$

Table 5.7: *Target parameters for dynamic range tests*

5.3.2 Results For 6 dB Dynamic Range

Figure 5.8 shows an example output from the 6 dB dynamic range tests. The noise level was 15 dB in this case, and all three approaches have visible noise below an amplitude of 0.5. FISTA-1B has a few noise peaks that are higher than that, which could be mistaken for targets during detection. None of the three approaches get close to estimating the full magnitude of the largest target, which would cause the NMSE to be proportionally higher even though it was detected.

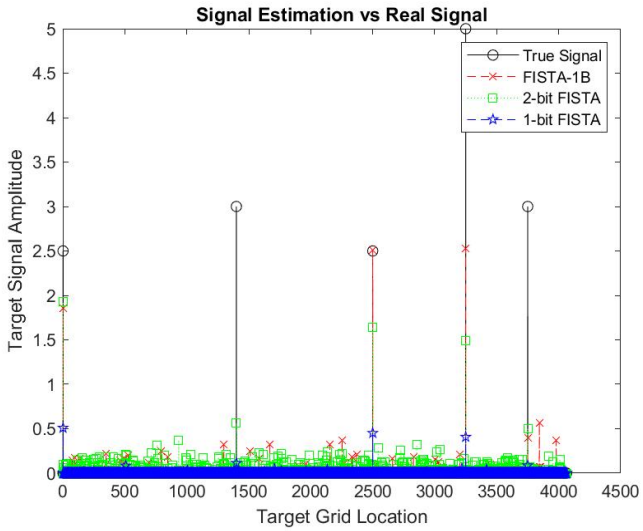

Figure 5.8: *Sample detection output for dynamic range of 6 dB*

The 6 dB DR case was the closest to the target parameters from previous tests, and the NMSE in Figure 5.9 shows similar behavior. All three approaches have their minimum error rates shifting to the right compared to before. 1-bit FISTA reaches its minimum at 10 dB instead of 0, for FISTA-1b the minimum moves from 40 dB to 50, and 2-bit FISTA has a narrower minimum at 5 dB instead of being stable from -5 to 5. This shows that dynamic range impacts the noise sensitivity for each approach. Error rates are also higher. 1-bit FISTA's minimum error raised to 50%, up from 40%. The other two approaches still reach similar minimums, showing that noise is not the only source of error still. Outside of the minimums their error rates have steeper slopes as SNR is reduced. It is worth noting that

estimations on the large signal can account for a higher percentage of the total error since it has a larger percentage of the signal power. The NMSE data is summarized in Table 5.8.

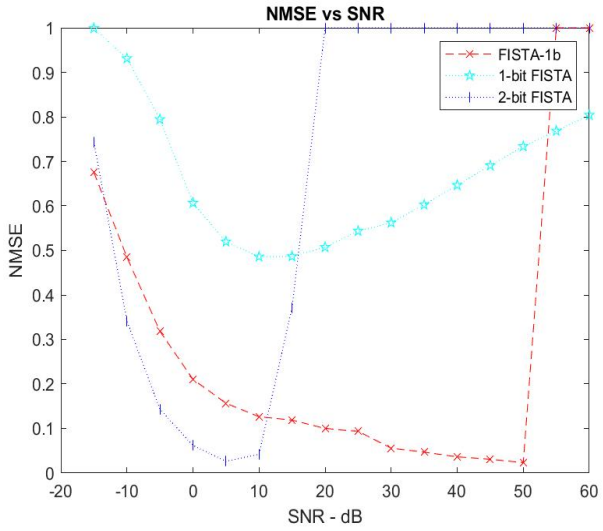

Figure 5.9: *NMSE for each approach at a dynamic range of 6 dB*

	FISTA-1b	**1-bit FISTA**	**2-bit FISTA**
Min NMSE	2.3%	48.5%	2.6%
SNR at Min	50 dB	10 dB	5 dB
Optimal Range	40 dB	25 dB	10 dB
SNR Limits	[NA,50]	[-10,NA]	[NA,15]
Total SNR Range	65<	70<	30<

Table 5.8: *Summary of NMSE data at 6 dB dynamic range. Optimal range is the width in dB where error rates are within* 10% *of min NMSE. Total SNR range is the width in dB that an algorithm is able to make detections.*

The ROC curves in Figure 5.10 are similar to before, with only a few changes. First, the SNR values they were measured at had to be adjusted to account for the shift in NMSE. The top left is measured at -10 dB SNR, where error rates have risen for each approach due to noise. In the top right the curves are measured at 5 dB SNR, the point where 2-bit FISTA has the lowest error. The bottom plot is at 50 dB SNR, where FISTA-1b has its lowest error rate.

The same patterns are present compared to the last two tests. At low SNR none of the

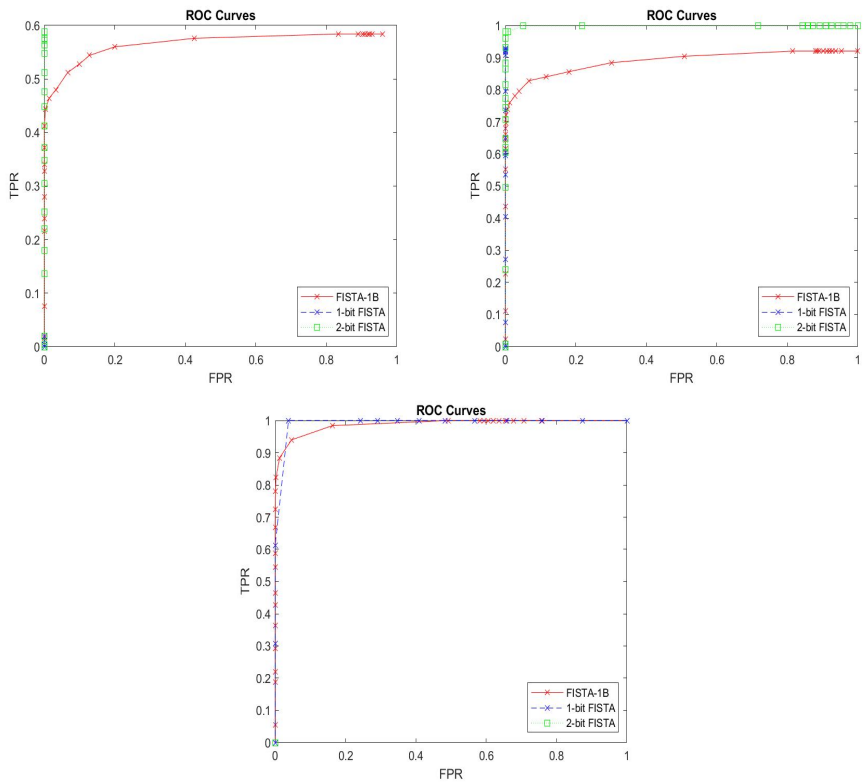

Figure 5.10: *ROC curves for dynamic range of 6 dB. SNR levels are -10 dB top left, 5 dB top right, 50 dB bottom.*

approaches detect all of the targets. 1-bit FISTA stops working entirely here, while the other two methods detect around 60% of the targets. FISTA-1b is still the only one with false positives, which could potentially be reduced by increasing the sparse regularization term. At 5 dB SNR 2-bit FISTA has perfect detections, which corresponds with its minimum NMSE at this point. The two 1-bit approaches are only detecting around 90% of targets, so there is a 10% reduction in their detection abilities. At high SNR the 1-bit approaches are still the only two functioning, and both show a small increase in false positives before reaching 100% detection. The AUC values in Table 5.9 reflect these trends.

	FISTA-1b	**1-bit FISTA**	**2-bit FISTA**
-10 dB	0.54	0.02	0.60
5 dB	0.89	0.93	0.99
50 dB	0.98	0.99	0
Average	0.80	0.65	0.53

Table 5.9: *AUC of the ROC curves for dynamic range of 6 dB*

5.3.3 Results For 12 dB Dynamic Range

A sample output for the 12 dB dynamic range tests is shown in Figure 5.11. The SNR for this sample is at 20 dB. It shows that both 2-bit FISTA and FISTA-1b have noise rising to an amplitude of 0.5, with some detections at similar values. Targets 1, 3, and 4 have confident detections from both. The estimates from 1-bit FISTA are almost to small to see for targets 1 and 3, with the only large estimate limited to target 4. The general reduction in amplitude estimates shows that the higher dynamic range is beginning to have an impact for each method.

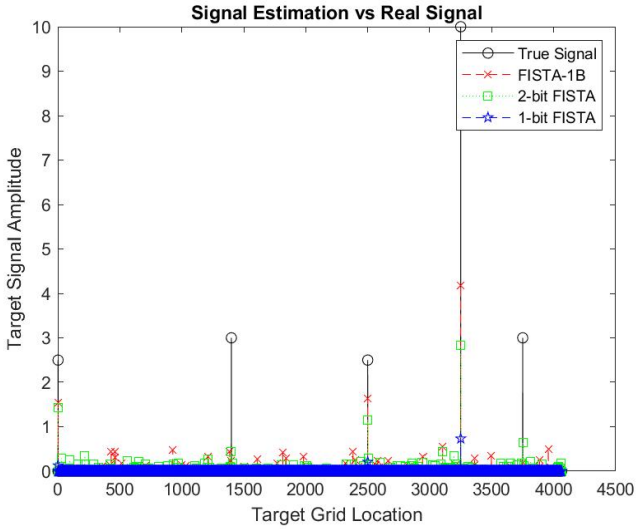

Figure 5.11: *Sample detection output for dynamic range of 12 dB*

The NMSE measurements for this test continue the trends that started at 6 dB of dynamic range. Figure 5.12 shows that each algorithm has higher minimum error, and that the minimums occur at higher SNR values than before. 2-bit FISTA rose from 2% minimum error at 5 dB SNR to 13.6% at 15 dB SNR. There is a noticeable narrowing of the functional range as well, with only 20 dB of SNR values below 80% error instead of 30 dB in the last test. The large jumps in error occur when the smaller four targets are no longer detected. This shows a lack of robustness to increasing dynamic range in 2-bit FISTA.

1-bit FISTA had its minimum estimation error rise from 48% to 63%, and has a flatter

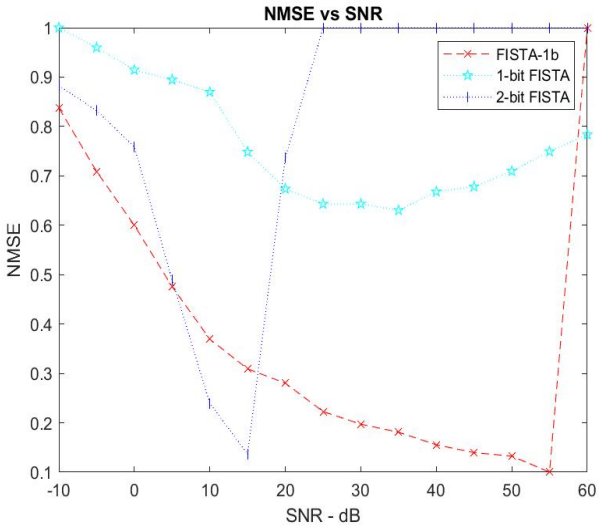

Figure 5.12: *NMSE for each approach at a dynamic range of 12 dB*

error curve than before. It hovers near its lowest error rate from 20 dB to 40 dB SNR, which is much broader than the 5 dB width minimums in all previous tests. At 40 dB SNR the error rate is 66%, compared to 65% and 68% from previous tests. The NMSE slope as SNR increases from 40 dB is also similar to previous tests. This is continued evidence that channel noise is not the main source of error at higher SNR values. Moving down from 10 dB SNR there is a distinct change in the error slope. Sharp changes like this indicate that fewer targets are being detected, which upper limits the amount of error when estimating those targets. The NMSE data is summarized in Table 5.10.

	FISTA-1b	**1-bit FISTA**	**2-bit FISTA**
Min NMSE	10.0%	63.0%	13.6%
SNR at Min	55 dB	35 dB	15 dB
Optimal Range	25 dB	30 dB	5 dB
SNR Limits	[NA,55]	[-5,NA]	[NA,20]
Total SNR Range	65<	65<	30<

Table 5.10: *Summary of NMSE data at 12 dB dynamic range. Optimal range is the width in dB where error rates are within* 10% *of min NMSE. Total SNR range is the width in dB that an algorithm is able to make detections.*

FISTA-1b was the least impacted by the increase in dynamic range. Its minimum error

had the smallest increase, from 2.3% to 10%. Like the other two approaches the minimum shifted to a higher SNR value of 55 dB, up from 50 dB in the last test. The estimation error level increases at a steady rate as SNR decreases, until a maximum NMSE of 83% is reached at -10 dB SNR. This shows that FISTA-1b is noise limited across its entire working range.

Once again the changes in the NMSE plot meant that new SNR values were selected to produce ROC curves. The low SNR example was taken at 5 dB, the point where 2-bit FISTA's NMSE rose above that of FISTA-1b. Next was 15 dB SNR, the location of 2-bit FISTA's minimum NMSE value. The last set of curves was produced at 55 dB SNR, where FISTA-1b had its lowest NMSE value. Figure 5.13 shows the results.

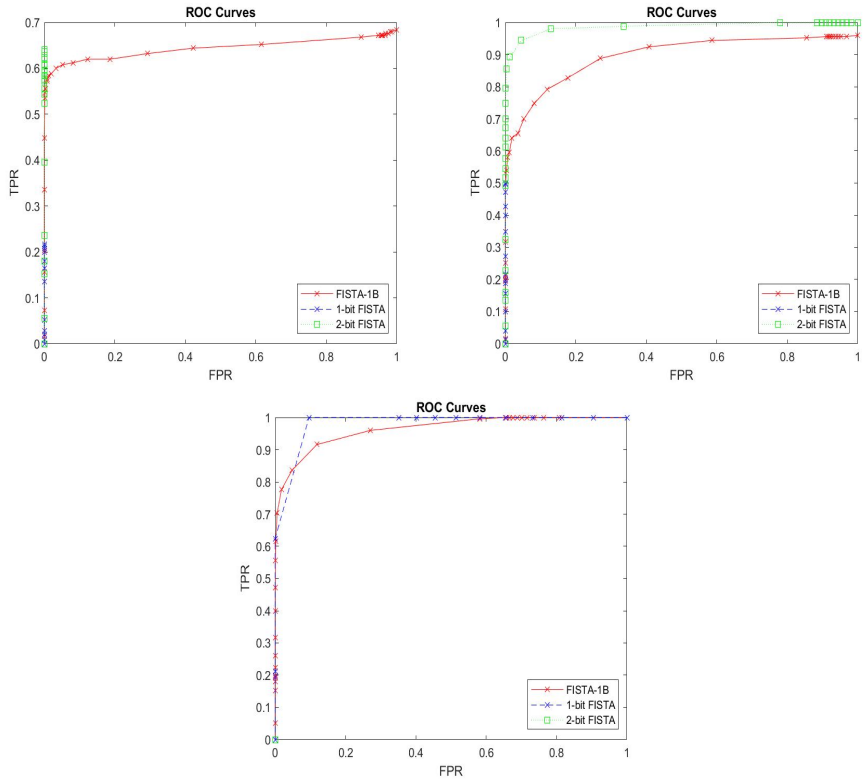

Figure 5.13: *ROC curves for dynamic range of 12 dB. SNR levels are 5 dB top left, 15 dB top right, 55 dB bottom.*

The low SNR plot is surprisingly unchanged compared to all previous tests. 2-bit FISTA and FISTA-1b still have maximum TPR values near 60%, with FISTA-1b admitting false positives to get there. 1-bit FISTA still performs poorly, with only 20% of targets detected.

It is only when moving to the medium SNR case that the impact of higher dynamic range can be seen. For the first time 2-bit FISTA cannot detect all targets without admitting some false positives. 1-bit FISTA is only detecting 50% of targets. When considering that with the shift of minimum NMSE to a higher SNR, it appears that 1-bit FISTA needs more than 15 dB of SNR to operate at this dynamic range. Like the previous tests, FISTA-1b settles into a detection rate of 90% at medium SNR. The difference this time is false positives start appearing much sooner, starting at 60% instead of 75% and 85% before. In the 55 dB SNR curves there is still similar performance between the two 1-bit methods. Both are able to detect all targets with some false positives. The AUC values for each curve are listed in Table 5.11.

	FISTA-1b	**1-bit FISTA**	**2-bit FISTA**
5 dB	0.64	0.24	0.65
15 dB	0.89	0.50	0.98
55 dB	0.96	0.98	0
Average	0.83	0.57	0.54

Table 5.11: *AUC of the ROC curves for dynamic range of 12 dB*

5.3.4 Results For 15 dB Dynamic Range

An Example of detection outputs at 15 dB of dynamic range are shown in Figure 5.14. In the higher SNR case of 20 dB all three methods are shown to work, though 1-bit FISTA only has one apparent detection. The other two methods have a large amount of noise close to the same amplitudes as their smaller target estimates. The amplitude estimates for the large target are low for all three methods, which will cause a large increase in the NMSE test if it happens consistently.

The NMSE for 15 dB of dynamic range can be seen in Figure 5.15. Once again error rates were higher across all SNR values. 2-bit FISTA appears to be at its limit here. The narrowing of the functional SNR range has continued with only one NMSE value below 50%. This minimum has a value of 27% and occurs at 20 dB SNR, 5 dB higher than the

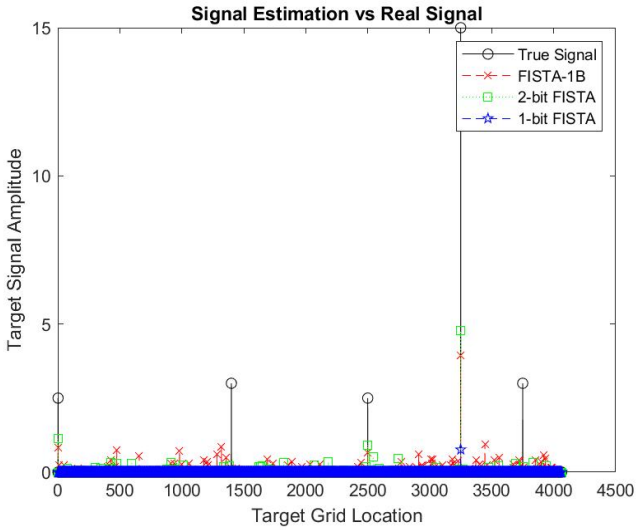

Figure 5.14: *Sample detection output for dynamic range of 15 dB*

last test. The flattening of 1-bit FISTA's NMSE continues as well, with all measurements above 30 dB SNR between 74% and 78%. Noise is a more dominant source of error here as dynamic range increases. The error rates for FISTA-1b have a steeper slope than before, but still gradually increase as SNR is lowered from the minimum error of 23% at 50 dB SNR. The NMSE data is summarized in Table 5.12.

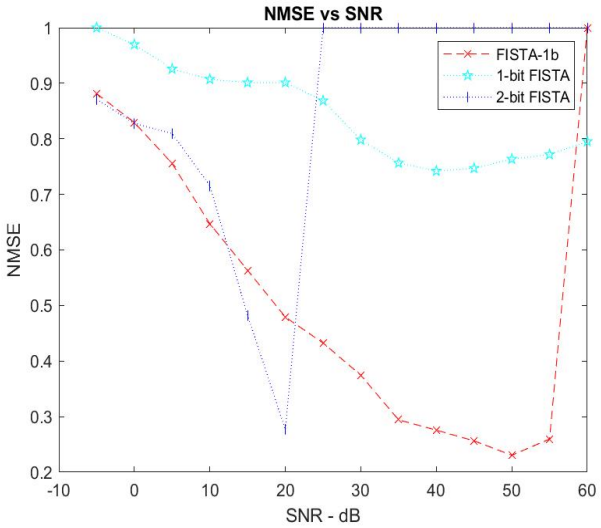

Figure 5.15: *NMSE for each approach at a dynamic range of 15 dB*

The ROC curves for this test are shown in Figure 5.16. In the top left the curves were

	FISTA-1b	1-bit FISTA	2-bit FISTA
Min NMSE	23.0%	74.2%	27.6%
SNR at Min	50 dB	40 dB	20 dB
Optimal Range	20 dB	30 dB	<5 dB
SNR Limits	[NA,55]	[0,NA]	[NA,20]
Total SNR Range	60<	60<	25<

Table 5.12: *Summary of NMSE data at 15 dB dynamic range. Optimal range is the width in dB where error rates are within* 10% *of min NMSE. Total SNR range is the width in dB that an algorithm is able to make detections.*

measured at 10 dB SNR, where the NMSE for 2-bit FISTA crosses higher than FISTA-1b. The top right shows curves taken at 20 dB where 2-bit FISTA has the lowest NMSE. The last set of curves in the bottom plot are at 50 dB SNR, where FISTA-1b has its minimum NMSE.

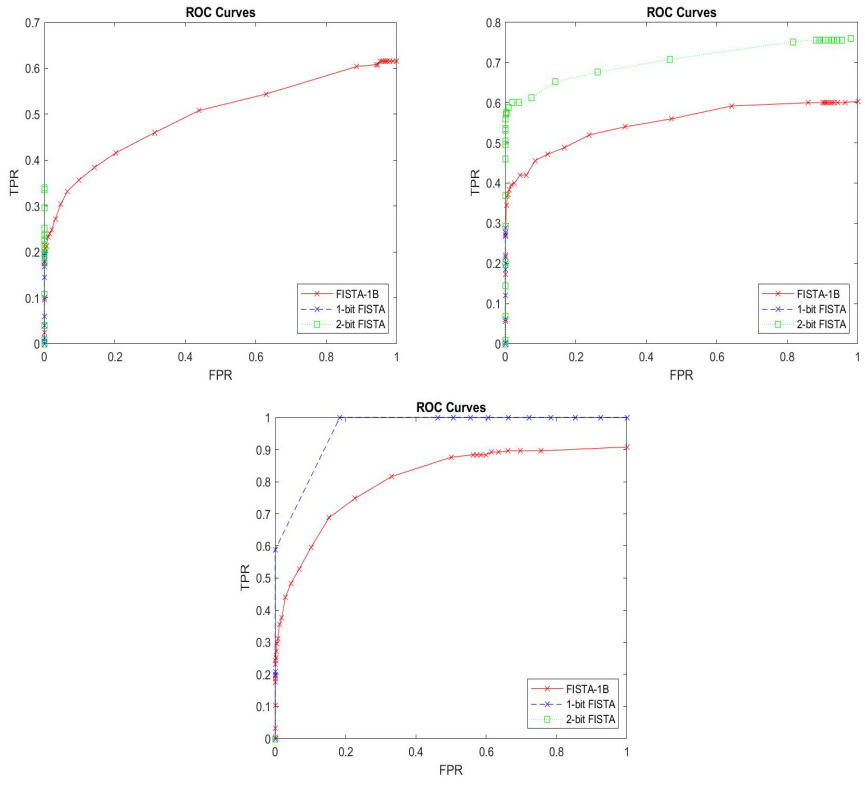

Figure 5.16: *ROC curves for dynamic range of 15 dB. SNR levels are 10 dB top left, 20 dB top right, 50 dB bottom.*

In all three plots the high dynamic range lowered detection rates. At low SNR 2-bit

FISTA was worse than FISTA-1b for the first time, with maximum TPR values of 35% and 62% respectively. Despite having the best maximum TPR, FISTA-1b only detected 20% of targets without any false positives, which was the same as 1-bit FISTA. At the mid-range SNR of 20 dB the detection rates still did not reach 100%. The best results were from 2-bit FISTA at 75%, followed by 60% for FISTA-1b, then 30% for 1-bit FISTA. Finally at 50 dB SNR 2-bit FISTA had stopped detecting as in previous tests. FISTA-1b was unable to detect all targets even though its NMSE was lowest here. The only good detection rate at this dynamic range came from 1-bit FISTA at high SNR. These results are reflected in the AUC values in Table 5.13.

	FISTA-1b	**1-bit FISTA**	**2-bit FISTA**
10 dB	0.49	0.20	0.35
20 dB	0.55	0.30	0.69
50 dB	0.80	0.96	0
Average	0.61	0.49	0.35

Table 5.13: *AUC of the ROC curves for dynamic range of 15 dB*

5.4 Final Conclusions, Limitations, and Future Work

The main goal of these experiments was to see if a 1-bit specific algorithm would provide better amplitude estimation than its few-bit equivalent for 1-bit measurements. Ultimately this was achieved, with the modified FISTA-1b performing much better than 1-bit FISTA. In the best case, the difference in minimum NMSE between the two was 35%. In the worst case it was 53%. The result is amplitude estimations with $84-93\%$ less error using FISTA-1b. 2-bit FISTA showed the same reduction in NMSE, but over a much smaller range of SNR values. Overall, FISTA-1b was the most robust to changing noise levels. For the tests with low dynamic range it stayed within 10% of its minimum NMSE over 2-5 times wider SNR range, with the best case covering a 55 dB span. At high dynamic range both 1-bit approaches maintain minimum NMSE over the same SNR width. 1-bit FISTA has over 60%

NMSE at that point, while FISTA-1b stays below 25%. 2-bit FISTA never operated near its minimum NMSE for more than a 15 dB span, and as dynamic range increased that was reduced to a single SNR level.

While FISTA-1b was great at amplitude estimation, it's detection accuracy tended to be lower than the original FISTA for each individual SNR value tested. The only time that it's individual performance was higher than the other two was at low SNR and the highest dynamic range. At high SNR it performed just below 1-bit FISTA, frequently $1-2\%$ lower based on the AUC values. At mid-range SNR it frequently didn't detect all targets, achieving $90-95\%$ at the maximum FPR. 2-bit FISTA was the best at these SNR values, which is expected since the 2-bit NMSE was lowest in that range. At low SNR none of the approaches detected all of the targets. FISTA-1b and 2-bit FISTA had TPR values within 10% of each other until the high dynamic range tests, but FISTA-1b would reach that rate with false positives. 1-bit FISTA did not perform well at low SNR, consistently detecting 25% or fewer targets. The real advantage of FISTA-1b was its operational range. It was able to make detections across the full span of SNR values, while 1-bit FISTA was weak to low SNR and 2-bit FISTA failed at high SNR. The average AUC values reflect the advantage in noise tolerance. FISTA-1b had $13-46\%$ higher average AUC than 1-bit FISTA and $48-74\%$ higher average AUC than 2-bit FISTA.

As for the tests themselves, the widely spaced target test was the best case scenario as intended. Aside from the low-SNR range, all three methods were able to detect all targets with few to no false positives. No significant change was observed when testing closely spaced targets. The only difference was a 5% lowering of the TPR for FISTA-1b at medium SNR. The best explanation for the lack of change is the Gaussian measurement matrix, which does not create the inter-symbol interference that would obscure adjacent target indices in a full PMCW radar simulation. The dynamic range tests clearly stresses all three methods, with NMSE rising and detection rates dropping as dynamic range was increased. There was a pattern in the NMSE plots of error curves rising and shifting towards

higher SNR as each dynamic range was tested. At 12 dB of dynamic range false positives became more common before reaching maximum TPR, and by 15 dB of dynamic range the only approach still reaching 100% TPR was 1-bit FISTA. This only happened at the highest SNR values.

One of the more interesting results was the rise in error rate for 1-bit FISTA as the SNR increased. The best culprit for this error is the quantization process. The main problem with using 1-bit depth in the original FISTA model is that the thresholds are $[-\infty, 0, \infty]$. All estimations will gravitate closer to zero in this case, since it will always be closer to the real signal amplitudes. This will keep the amplitude estimates low, raising the NMSE. When enough noise is present there is a better chance that a noisy estimate is moved closer to the correct amplitude, reducing NMSE. Eventually there is enough noise that it becomes the main source of estimation error, and the NMSE rises again.

There are a few limitations to the results obtained here. First, the ROC curves are a good way to show how accurate each approach is, but the impact of using sparse signal processing algorithms to create them needs to be explored further. Specifically, in classic radar detection each target index is relatively independent aside from interference from neighboring cells. In sparse signal processing all target indices are measured simultaneously and processed as a group. Another limitation is that only one sparse parameter tuning was used. Additional testing here was beyond the scope of the thesis but will be part of future development. Finally, both FISTA versions assume a known noise standard deviation. In practice this would have to be estimated, and mismatches of that estimate need to be explored.

CHAPTER 6

Conclusion and Future Work

This thesis explored how one-bit specific design impacts detection algorithms for ADAS radar. First it was established that there are benefits to PMCW radar replacing the current FMCW standard given improvements to both hardware implementations and future development. With that in mind the literature was reviewed to determine what the biggest challenges facing PMCW development are today. It was shown that the high frequencies used for ADAS radar push the capabilities of the ADCs in the receiver, specifically the balance of sample rate, power consumption, and resolution. The sampling rate could not be lowered, so a way to minimize the resolution bits was required to keep power consumption low.

Solutions were explored to find how many bits were required to prevent poor detection capabilities. The research showed that few and even 1-bit approaches were possible, but algorithms using 1-bit quantized data lost valuable amplitude information. This is because traditional 1-bit quantization only compares inputs to a single fixed threshold. A method was found to fix this problem, which came from the communications field. The 1-bit ADC sampling threshold was varied over time to capture more information about the signal amplitude. This resulted in better signal estimations despite only having one resolution bit.

To further explore this concept, a few-bit approach using FISTA was adapted to use 1-bit TVT measurements, called FISTA-1b. Both the original and the adapted model were then tested on their ability to preserve signal amplitudes, distinguish closely spaced targets, and

handle targets with large dynamic range. In trials measuring the NMSE of real vs estimated signal amplitudes, it was shown that FISTA-1b had significantly better amplitude estimation compared to the original method using 1-bit measurements. A reduction in NMSE $84-93\%$ was achieved depending on the SNR level. The original FISTA with 2-bit measurements had almost identical NMSE performance to FISTA-1b at low to medium SNR, but it was much more sensitive to the noise level and stopped detecting above 10 dB in most tests.

With regards to detection accuracy, 2-bit FISTA was consistently the best over the range it operated. At low SNR FISTA-1b had similar detection rates, but a higher number of false positives. At mid-range SNR 2-bit FISTA had perfect detections until it hit its upper SNR limit. At high SNR both 1-bit methods detected all targets with fewer than 5% FPR. While there wasn't an individual SNR level where FISTA-1b outperformed the original FISTA, it did achieve similar performance and over a much larger range of SNR values. The average AUC for each test shows how robust it is to noise, outperforming 1-bit FISTA by $13-46\%$ and 2-bit FISTA by $48-74\%$.

There were several opportunities and limitations discovered in the process of this work. First, the detection of targets that are closely spaced showed no performance drop, which is likely due to the ideal conditions of the Gaussian random measurement matrix. Part of the future work for this project is to implement a full PMCW radar transceiver simulation to generate the measurement matrix data. There were also trends in the NMSE plots that warrant further investigation, such as the source of the error that created the lower limit for 2-bit FISTA and FISTA-1b, and the error that affected 1-bit FISTA in high SNR cases. Finally, future tests will explore aspects related to sparsity, such as the number of targets and the tuning of the sparse regularization parameter. Altogether this work will help create a framework for comparing and evaluating few and 1-bit radar algorithms like FISTA to achieve the best detection capabilities with power efficient ADCs.

Bibliography

[1] W. Stark, M. Ali, and M. Maher, "Digital code modulation (dcm) radar for automotive application," 2020.

[2] B. Templeton, "Tesla in taiwan crashes directly into overturned truck, ignores pedestrian, with autopilot on," Forbes.com, Ed., 2020. [Online]. Available: https://www.forbes.com/sites/bradtempleton/2020/06/02/tesla-in-taiwan-crashes-directly-into-overturned-truck-ignores-pedestrian-with-autopilot-on/?sh=75c9d5f058e5.

[3] A. Gazis, E. Ioannou, and E. Katsiri, "Examining the sensors that enable self-driving vehicles," *IEEE Potentials*, vol. 39, no. 1, pp. 46–51, 2020. DOI: 10.1109/MPOT.2019.2941243.

[4] V. Giannini, D. Guermandi, Q. Shi, *et al.*, "A 79 ghz phase-modulated 4 ghz-bw cw radar transmitter in 28 nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 12, pp. 2925–2937, 2014. DOI: 10.1109/JSSC.2014.2355819.

[5] S. International, "Surface vehicle recommended practice, j3016-202104," 2021.

[6] F. Hafeez, U. U. Sheikh, N. Alkhaldi, H. Z. A. Garni, Z. A. Arfeen, and S. A. Khalid, "Insights and strategies for an autonomous vehicle with a sensor fusion innovation: A fictional outlook," *IEEE Access*, vol. 8, pp. 135 162–135 175, 2020. DOI: 10.1109/ACCESS.2020.3010940.

[7] R. Roriz, J. Cabral, and T. Gomes, "Automotive lidar technology: A survey," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, 2021. DOI: 10.1109/TITS.2021.3086804.

[8] T. Raj, F. Hashim, A. Huddin, M. F. Ibrahim, and A. Hussain, "A survey on lidar scanning mechanisms," *Electronics*, vol. 9, p. 741, Apr. 2020. DOI: 10.3390/electronics9050741.

[9] A. Swief and M. El-Habrouk, "A survey of automotive driving assistance systems technologies," in *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, 2018, pp. 1–12. DOI: 10.1109/IDAP.2018.8620826.

[10] C. Premebida, O. Ludwig, and U. Nunes, "Lidar and vision-based pedestrian detection system," *Journal of Field Robotics*, vol. 26, pp. 696–711, Sep. 2009. DOI: 10.1002/rob.20312.

[11] A. Deever, M. Kumar, and B. Pillman, "Digital camera image formation: Processing and storage," in *Digital Image Forensics: There is More to a Picture than Meets the Eye*, H. T. Sencar and N. Memon, Eds. New York, NY: Springer New York, 2013, pp. 45–77, ISBN: 978-1-4614-0757-7. DOI: 10.1007/978-1-4614-0757-7_2. [Online]. Available: https://doi.org/10.1007/978-1-4614-0757-7_2.

[12] M. Mody, N. Nandan, S. Dabral, *et al.*, "Image signal processing for front camera based automated driver assistance system," in *2015 IEEE 5th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, 2015, pp. 158–159. DOI: 10.1109/ICCE-Berlin.2015.7391221.

[13] S. Saponara, M. S. Greco, and F. Gini, "Radar-on-chip/in-package in autonomous driving vehicles and intelligent transport systems: Opportunities and challenges," *IEEE Signal Processing Magazine*, vol. 36, no. 5, pp. 71–84, 2019. DOI: 10.1109/MSP.2019.2909074.

[14] A. G. Venon, Y. Dupuis, P. Vasseur, and P. Merriaux, "Millimeter wave fmcw radars for perception, recognition and localization in automotive applications: A survey," *IEEE Transactions on Intelligent Vehicles*, pp. 1–1, 2022. DOI: 10.1109/TIV.2022.3167733.

[15] V. Ilic, M. Marijan, A. Mehmed, and M. Antlanger, "Development of sensor fusion based adas modules in virtual environments," in *2018 Zooming Innovation in Consumer Technologies Conference (ZINC)*, 2018, pp. 88–91. DOI: 10.1109/ZINC.2018.8448849.

[16] J. Kim, D. S. Han, and B. Senouci, "Radar and vision sensor fusion for object detection in autonomous vehicle surroundings," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2018, pp. 76–78. DOI: 10.1109/ICUFN.2018.8436959.

[17] S. L. Poczter and L. M. Jankovic, "The google car: Driving toward a better future?" *Journal of Business Case Studies*, vol. 10, no. 1, pp. 7–14, 2013. DOI: 10.19030/jbcs.v10i1.8324.

[18] D. L. Rosenband, "Inside waymo's self-driving car: My favorite transistors," in *2017 Symposium on VLSI Circuits*, 2017, pp. C20–C22. DOI: 10.23919/VLSIC.2017.8008500.

[19] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 22–35, 2017. DOI: 10.1109/MSP.2016.2628914.

[20] A. G. Venon, Y. Dupuis, P. Vasseur, and P. Merriaux, "Millimeter wave fmcw radars for perception, recognition and localization in automotive applications: A survey," *IEEE Transactions on Intelligent Vehicles*, pp. 1–1, 2022. DOI: 10.1109/TIV.2022.3167733.

[21] R. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 539–550, 1999. DOI: 10.1109/49.761034.

[22] H. Landau, "Sampling, data transmission, and the nyquist rate," *Proceedings of the IEEE*, vol. 55, no. 10, pp. 1701–1706, 1967. DOI: 10.1109/PROC.1967.5962.

[23] N. N. Çikan and M. Aksoy, "Analog to digital converters performance evaluation using figure of merits in industrial applications," in *2016 European Modelling Symposium (EMS)*, 2016, pp. 205–209. DOI: 10.1109/EMS.2016.043.

[24] J. Liu and S. Wright, "Robust dequantized compressive sensing," *Applied and Computational Harmonic Analysis*, vol. 37, Jul. 2012. DOI: 10.1016/j.acha.2013.12.006.

[25] E. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, Aug. 2006. DOI: 10.1002/cpa.20124.

[26] L. David, "Donoho. compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, pp. 1289–1306, Jan. 2006.

[27] A. Harms, W. U. Bajwa, and R. Calderbank, "Identification of linear time-varying systems through waveform diversity," *IEEE Transactions on Signal Processing*, vol. 63, no. 8, pp. 2070–2084, 2015. DOI: 10.1109/TSP.2015.2407319.

[28] U. Ahmad, D. Guermandi, A. Medra, W. Van Thillo, and A. Bourdoux, "Impact of even and odd order non-linearity on pmcw radars," in *2016 IEEE Radar Conference (RadarConf)*, 2016, pp. 1–5. DOI: 10.1109/RADAR.2016.7485135.

[29] F. Norouzian, A. A. Pirkani, E. Hoare, M. Cherniakov, and M. Gashinova, "Phenomenology of automotive radar interference," *IET Radar, Sonar Navigation*, vol. 15, Sep. 2021. DOI: 10.1049/rsn2.12096.

[30] A. Bourdoux, K. Parashar, and M. Bauduin, "Phenomenology of mutual interference of fmcw and pmcw automotive radars," in *2017 IEEE Radar Conference (RadarConf)*, 2017, pp. 1709–1714. DOI: 10.1109/RADAR.2017.7944482.

[31] J. Overdevest, F. Jansen, F. Laghezza, F. Uysal, and A. Yarovoy, "Uncorrelated interference in 79 ghz fmcw and pmcw automotive radar," in *2019 20th International Radar Symposium (IRS)*, 2019, pp. 1–8. DOI: 10.23919/IRS.2019.8768181.

[32] G. Brooker, "Understanding millimetre wave fmcw radars," *1st International Conference on Sensing Technology*, Jan. 2005.

[33] V. Giannini, D. Guermandi, Q. Shi, *et al.*, "A 79 ghz phase-modulated 4 ghz-bw cw radar transmitter in 28 nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 12, pp. 2925–2937, 2014. DOI: 10.1109/JSSC.2014.2355819.

[34] B. Murmann, "Adc performance survey 1997-2021," *Available: http://web.stanford.edu/ murmann/adcsurvey.html*, 2021.

[35] C. Aydogdu, N. Garcia, L. Hammarstrand, and H. Wymeersch, "Radar communications for combating mutual interference of fmcw radars," in *2019 IEEE Radar Conference (RadarConf)*, 2019, pp. 1–6. DOI: 10.1109/RADAR.2019.8835744.

[36] S. H. Dokhanchi, B. S. Mysore, K. V. Mishra, and B. Ottersten, "A mmwave automotive joint radar-communications system," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 3, pp. 1241–1260, 2019. DOI: 10.1109/TAES.2019.2899797.

[37] W. Van Thillo, V. Giannini, D. Guermandi, S. Brebels, and A. Bourdoux, "Impact of adc clipping and quantization on phase-modulated 79 ghz cmos radar," in *2014 11th European Radar Conference*, 2014, pp. 285–288. DOI: 10.1109/EuRAD.2014.6991263.

[38] J. Mo, P. Schniter, and R. W. Heath, "Channel estimation in broadband millimeter wave mimo systems with few-bit adcs," *IEEE Transactions on Signal Processing*, vol. 66, no. 5, pp. 1141–1154, 2018. DOI: 10.1109/TSP.2017.2781644.

[39] J. Liu and S. Wright, "Robust dequantized compressive sensing," *Applied and Computational Harmonic Analysis*, vol. 37, Jul. 2012. DOI: 10.1016/j.acha.2013.12.006.

[40] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sciences*, vol. 2, pp. 183–202, Jan. 2009. DOI: 10.1137/080716542.

[41] C.-Y. Wu, J. Li, and T. F. Wong, "Sparse parameter estimation for pmcw mimo radar using few-bit adcs," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 4415–4419. DOI: 10.1109/ICASSP39728.2021.9414267.

[42] C.-Y. Wu, T. Zhang, J. Li, and T. F. Wong, "Parameter estimation in pmcw mimo radar systems with few-bit quantized observations," *IEEE Transactions on Signal Processing*, vol. 70, pp. 810–821, 2022. DOI: 10.1109/TSP.2022.3146790.

[43] C. Gianelli, L. Xu, J. Li, and P. Stoica, "One-bit compressive sampling with time-varying thresholds for multiple sinusoids," in *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2017, pp. 1–5. DOI: 10.1109/CAMSAP.2017.8313172.

[44] X. Shang, H. Zhu, and J. Li, "Range-doppler imaging via one-bit pmcw radar," in *2020 IEEE 11th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2020, pp. 1–5. DOI: 10.1109/SAM48682.2020.9104290.

[45] X. Shang, J. Li, and P. Stoica, "Weighted spice algorithms for range-doppler imaging using one-bit automotive radar," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 4, pp. 1041–1054, 2021. DOI: 10.1109/JSTSP.2021.3071601.

APPENDIX A

Radar Simulation Code

This section contains all MATLAB code created by the author to produce the results of this thesis.

A.1 NMSE Testing Code

A.1.1 Main Code

```
%% nmse_main
%
%   Author: Ethan Triplett
%   Created: 5/31/22
%   Last Edited: 7/24/22
%
%   This code aims to test the amplitude estimation accuracy of all
    three
%   FISTA approaches.
%
clc;
%clf;
disp("Welcome to the NMSE FISTA radar program - " + datestr(now));

%% Global Variables
C = 127;                % Number of chips per pulse
Q = 8;                  % Number of pulses per coherent processing
    interval
N = 4064;               % Width of measurement matrix
M = Q*C;                % Number of samples [2]
sigma = 1.81;           % Noise variance
q_bits = 1;             % Number of bits for quantization
num_tvt = 2;           % Number of TVT values

%% Create sparse vector using target data

% % Closely Spaced Target Parameters
%amp1 = 3*exp(1i*0.7854);
%amp2 = 2.5*exp(1i*0.7821);
```

```
%target_data = [
%     2, amp2;
%     2485, amp1;
%     2500, amp2;
%     3250, amp2;
%     3251, amp1;];

% Widely Spaced Target Parameters
 amp1 = 3*exp(1i*0.7854);
 amp2 = 2.5*exp(1i*0.7821);
 target_data = [
      2, amp2;
      1400, amp1;
      2500, amp2;
      3250, 5*exp(1i*0.7854);     % amp2;%
      3750, amp1;];

% Create the sparse vector and measurement matrix
x = create_sparse_x(target_data,N);
x_tilde = [real(x); imag(x)];

A = randn(M,N) + 1i.*randn(M,N);
A_tilde = [real(A), -imag(A); real(A), imag(A)];

snr = 20*log(max(abs(target_data(:,2)))/(sigma));

% sigma = [13.44, 10.46, 8.14]; % low DR, -30--20 SNR
 sigma = [6.36, 4.96, 3.86 3.0, 2.33, 1.81, 1.42, 1.1, 0.86 ...]; % low DR
          0.67, 0.52, 0.406, 0.316, 0.246]; % low DR 30-50 SNR

% sigma = [10.6, 8.24, 6.42, 5.0, 3.88, 3.02, 2.36, 1.83, 1.43,...]; % 6dB DR
%      1.116, 0.868, 0.676, 0.526, 0.41]; % 6dB DR 30-50 SNR
%sigma = [0.32, 0.25]; % 6dB DR 55-60 SNR


%sigma = [16.5, 12.84, 10, 7.8, 6.06, 4.72, 3.68, 2.86, 2.24]; % 12 dB DR -10 to 30 SNR
%    1.73, 1.34, 1.05, 0.82, 0.64, 0.5]; % 12dB DR 35-60 SNR
%sigma = [0.64, 0.5, 0.39, 0.3]; % 12dB DR 55-70 SNR

%sigma =  [19.26, 15, 11.68, 9.1, 7.08, 5.5, 4.3, 3.34, 2.6,...
%     2.02, 1.58, 1.22, 0.96, 0.74]; % 15dB DR -5 to 60 SNR

%sigma = [0.96, 0.74, 0.58, 0.44]; % 15dB DR 55-70 SNR


trials = 50;
vars = length(sigma);
test_data_tvt = zeros([vars,trials]);
test_data_org1 = zeros([vars,trials]);
test_data_org2 = zeros([vars,trials]);

```

```
for j = 1:vars
    snr = 20*log(max(abs(target_data(:,2)))/sigma(j))
    for i = 1:trials

        noise = sigma(j).*(randn(M,1)+1i*randn(M,1));
        y = A*x + noise;
        yri = [real(y); imag(y)];
        [yq_tvt, hri, yq_org1, ud1, udplus1_1, yq_org2, ud2,
   udplus1_2] = ...
            get_quantized3(yri, q_bits, num_tvt);

        % Solve
        xhat1 = tvt_FISTA(A_tilde, yq_tvt, hri, sigma(j), 1e-5, 100,
     150);
        xhat2 = org_FISTA(A_tilde, ud1, udplus1_1, sigma(j), 1e-5, 1
   e3, 60);
        xhat3 = org_FISTA(A_tilde, ud2, udplus1_2, sigma(j), 1e-5, 1
   e3, 60);

        % Convert estimates to complex magnitudes
        xhatc1 = abs(xhat1(1:N) + 1i*xhat1(N+1:end));
        xhatc2 = abs(xhat2(1:N) + 1i*xhat2(N+1:end));
        xhatc3 = abs(xhat3(1:N) + 1i*xhat3(N+1:end));


        %display_real_v_2est(abs(x),abs(xhatc1),abs(xhatc2),abs(
   xhatc3));

        %Organize relevant amplitude data
        amplitudes = [abs(target_data(:,2)), ...
                        xhatc1(target_data(:,1)), ...
                        xhatc2(target_data(:,1)), ...
                       xhatc3(target_data(:,1))];


        % Calculate error in amplitudes
        test_data_tvt(j,i) = sum( ...
            (abs(target_data(:,2)) - xhatc1(target_data(:,1))).^2
   ...
                ./abs(target_data(:,2).^2));
        test_data_org1(j,i) = sum( ...
            (abs(target_data(:,2)) - xhatc2(target_data(:,1))).^2
   ...
                ./abs(target_data(:,2).^2));
        test_data_org2(j,i) = sum( ...
            (abs(target_data(:,2)) - xhatc3(target_data(:,1))).^2
   ...
                ./abs(target_data(:,2).^2));


    end
    disp("Trial batch complete - " + datestr(now));
end

```

```
125
126 % Calculate final error by summing rows and dividing by 6*100
127 tvt_nmse = (1/(5*trials)).*sum(test_data_tvt,2);
128 org1_nmse = (1/(5*trials)).*sum(test_data_org1,2);
129 org2_nmse = (1/(5*trials)).*sum(test_data_org2,2);
130
131
132 writematrix(test_data_tvt);
133 writematrix(test_data_org1);
134 writematrix(test_data_org2);
135
136 display_nmse(tvt_nmse, org1_nmse, org2_nmse);
```

Listing A.1: *ROCmain*

A.1.2 Supporting Functions

```
1  function display_nmse(tvt, org1, org2)
2  %% display_nmse
3  %
4  %   Author: Ethan Triplett
5  %   Created: 7/15/22
6  %   Last Edited: 7/24/22
7  %
8  %   This code displays the averaged NMSE for all three FISTA methods
9  %
10
11 figure1 = figure;
12 x_axis = -15:5:60;
13 tvt = plot(x_axis,tvt);
14 tvt.Color = 'red';
15 tvt.LineStyle = '--';
16 tvt.Marker = 'x';
17 hold("on")
18 org1 = plot(x_axis,org1);
19 org1.Color = 'cyan';
20 org1.LineStyle = ':';
21 org1.Marker = 'p';
22 hold("on")
23 org1 = plot(x_axis,org2);
24 org1.Color = 'blue';
25 org1.LineStyle = ':';
26 org1.Marker = '|';
27 hold("off")
28
29 title("NMSE vs SNR")
30 ylabel("NMSE")
31 xlabel("SNR - dB")
32 legend("FISTA-1b", "1-bit FISTA", "2-bit FISTA", ...
33         'location','northeast')
34 saveas(figure1,"nmse.jpg");
35 saveas(figure1,"nmse",'fig');
```

Listing A.2: *ROCmain*

A.2 Detection Accuracy Testing Code

This section contains the code used to measure detection accuracy. The main function calculates the error rates for all three FISTA methods then averages the results over all trials.

A.2.1 Main Code

```
%% ROC_main
%
%   Author: Ethan Triplett
%   Created: 7/15/22
%   Last Edited: 7/24/22
%
%   This code tests detection accuracy of each algorithm by producing ROC
%   curves. Curves are generated by classifying targets as either detected
%   or not based on a set of amplitude thresholds.
%
clc;
%clf;
disp("Welcome to the FISTA ROC curve program - " + datestr(now));

%% Global Variables
C = 127;                % Number of chips per pulse
Q = 8;                  % Number of pulses per coherent processing interval
N = 4064;               % Width of measurement matrix
M = Q*C;                % Number of samples [2]
q_bits = 1;             % Number of bits for quantization
num_tvt = 2;            % Number of TVT values

%% Create sparse vector using target data

% % Closely Spaced Target Parameters
% amp1 = 3*exp(-1i*0.349);
% amp2 = 2.5*exp(1i*0.7821);
% target_data = [
%       2, amp2;
%       2485, amp1;
%       2500, amp2;
%       3250, amp2;
%       3251, amp1;];

% Widely Spaced Target Parameters
amp1 = 3*exp(-1i*0.349);
amp2 = 2.5*exp(1i*0.7821);
target_data = [
     2, amp2;
     1400, amp1;
     2500, amp2;
     3250, 5*exp(-1i*0.349);   %amp2;%
```

```
     3750, amp1;];

% Create the sparse vector and measurement matrix
x = create_sparse_x(target_data,N);
x_tilde = [real(x); imag(x)];

A = randn(M,N) + 1i.*randn(M,N);
A_tilde = [real(A), -imag(A); real(A), imag(A)];

sigma = 10.6;          % Noise standard deviation
snr = 20*log(max(abs(target_data(:,2)))/(sigma));
snr_mean = 20*log(mean(abs(target_data(:,2)))/(sigma));

% Initialize data storage matrices
trials = 50;
vars = 40;
tp_tvt = zeros([vars,trials]);
fp_tvt = zeros([vars,trials]);
tp_org1 = zeros([vars,trials]);
fp_org1 = zeros([vars,trials]);
tp_org2 = zeros([vars,trials]);
fp_org2 = zeros([vars,trials]);

npv_tvt = zeros([vars,trials]);
ppv_tvt = zeros([vars,trials]);
npv_org1 = zeros([vars,trials]);
ppv_org1 = zeros([vars,trials]);
npv_org2 = zeros([vars,trials]);
ppv_org2 = zeros([vars,trials]);

% Perform trials
for i = 1:trials

    % Generate new noisy measurements
    noise = sigma.*(randn(M,1)+1i*randn(M,1));
    y = A*x + noise;
    yri = [real(y); imag(y)];
    [yq_tvt, hri, yq_org1, ud1, udplus1_1, yq_org2, ud2, udplus1_2]
   = ...
        get_quantized3(yri, q_bits, num_tvt);

    % Solve
    xhat1 = tvt_FISTA(A_tilde, yq_tvt, hri, sigma, 1e-5, 100, 150);
    xhat2 = org_FISTA(A_tilde, ud1, udplus1_1, sigma, 1e-5, 1e3, 60)
   ;
    xhat3 = org_FISTA(A_tilde, ud2, udplus1_2, sigma, 1e-5, 1e3, 60)
   ;

    % Convert estimates to complex magnitudes
    xhatc1 = xhat1(1:N) + 1i*xhat1(N+1:end);
    xhatc2 = xhat2(1:N) + 1i*xhat2(N+1:end);
    xhatc3 = xhat3(1:N) + 1i*xhat3(N+1:end);

```

```
    %display_real_v_2est(abs(x),abs(xhatc1),abs(xhatc2),abs(xhatc3),
    snr)

    for j = 1:vars

        % Get true positive rates and false positive rates
        [tp_tvt(j,i), fp_tvt(j,i), npv_tvt(j,i), ppv_tvt(j,i)] = ...
                                                get_error(x,xhatc1,j);
        [tp_org1(j,i), fp_org1(j,i), npv_org1(j,i), ppv_org1(j,i)] =
    ...
                                                get_error(x,xhatc2,j);
        [tp_org2(j,i), fp_org2(j,i), npv_org2(j,i), ppv_org2(j,i)] =
    ...
                                                get_error(x,xhatc3,j);

    end
    disp("Trial batch "+i+" complete - " + datestr(now));
end

% Average the TPR and FPR over all trials
tpr_tvt = mean(tp_tvt,2);
fpr_tvt = mean(fp_tvt,2);
tpr_org1 = mean(tp_org1,2);
fpr_org1 = mean(fp_org1,2);
tpr_org2 = mean(tp_org2,2);
fpr_org2 = mean(fp_org2,2);


% Display ROC curves
display_roc()

AUC = [trapz(flip(fpr_tvt), flip(tpr_tvt));
       trapz(flip(fpr_org1), flip(tpr_org1));
       trapz(flip(fpr_org2), flip(tpr_org2))];
```

Listing A.3: *ROCmain*

A.2.2 Supporting Functions

```
%% get_error
%
%   Author: Ethan Triplett
%   Created: 7/15/22
%   Last Edited: 7/20/22
%
%   This code takes a signal estimate and the original signal and
    produces
%   the rates of false positives and true positives based on a
    detection
%   threshold amps(j). All targets below the selected amplitude are
%   considered "not-detected".
%
function [tpr, fpr, npv, ppv] = get_error(x,xhat,j)

```

```
top = max([max(abs(x)), max(abs(xhat))]);
% Thresholds for classifying "detected" vs "not-detected"
amps = [0.000000001, linspace(0,0.01,10), linspace(0.02,1,16), ...
          1.3, 1.6, 2.0, 2.2, 2.6, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0,
    0.1+top];

abs_xhat = abs(xhat);

% Sort estimated values by detection state
p = find(abs_xhat > amps(j));  %
n = intersect(find(abs_xhat <= amps(j)),find(abs_xhat > 0));

% Find true targets
act = find(x);

% Calculate confusion matrix values
tp = sum(ismember(act,p));
fp = length(setdiff(p,act));
fn = 5-tp;%sum(ismember(act,n));
tn = length(setdiff(n,act));

% Calculate the rates of true positive and false positives
tpr = tp/(tp+fn+0.00001);
fpr = fp/(fp+tn+0.00001);

npv = fn/(fn+tp+0.00001);   % Actually NPV
ppv = tp/(tp+fp+0.00001);   % Actually PPV
```

Listing A.4: *ROCmain*

```
%% display_roc
%
%   Author: Ethan Triplett
%   Created: 7/15/22
%   Last Edited: 7/20/22
%
%   This code plots the ROC curves of three data sets. For each
    curve there
%   needs to be two variables representing the TPR and FPR. The rows
     should
%   represent each threshold level. The columns are the result of
    each
%   trial. Data sets are averaged across all trials before plotting.
%

disp("ROC curve - " + datestr(now));

% Display ROC curves
figure1 = figure;
xp = plot(fpr_tvt,tpr_tvt);
xp.Color = 'red';
xp.Marker = 'x';
hold('on')
xh1 = plot(fpr_org1,tpr_org1);
```

```
xh1.Color = 'blue';
xh1.LineStyle = '--';
xh1.Marker = 'x';
hold("on")
xh2 = plot(fpr_org2,tpr_org2);
xh2.Color = 'green';
xh2.LineStyle = ':';
xh2.Marker = 's';
hold("off")

title("ROC Curves")
xlabel("FPR")
ylabel("TPR")
legend("FISTA-1B", "1-bit FISTA","2-bit FISTA","Location","southeast
    ")
saveas(figure1,"ROC_curve.jpg");
saveas(figure1,"ROC_curve",'fig');
```

Listing A.5: *ROCmain*

A.3 FISTA Implementations

This section contains the original and TVT FISTA-1b implementations

A.3.1 Original FISTA

```
function xhat = org_FISTA(A, ud, udplus1, sigma, step,
    max_iterations, rho)
%% FISTA - Fast Iterative Shrinkage-Thresholding Algorithm
%
%   Author: Ethan Triplett
%   Created: 2/23/22
%   Last Edited: 4/12/22
%
%   Solves a group-sparse radar detection problem using FISTA.
%
%   Inputs:
%   A - the all-real dictionary matrix [R, -I; I, R]
%   ud - lower quantization threshold per measurement y
%   udplus1 - upper quantization threshold per measurement y
%   sigma - standard deviation of the additive noise
%
%   Outputs:
%   detections
%

%% Initialization
%disp("Initializing org_FISTA algorithm - " + datestr(now));
grid_length = length(A(1,:));
G = grid_length/2;
eta = 1;
%rho = 30;                    % Regularization parameter determining
    sparsity
```

```
xhat = zeros([grid_length, 1]);
z = zeros([grid_length, 1]);

% Stop conditions
min_change = 10^-6;


%% Main FISTA loop
%disp("Beginning FISTA iterations - " + datestr(now));
for i = 1:max_iterations
    prior_xhat = xhat;
    prior_eta = eta;

    mean = (A*z);
    pdf_up = normpdf((udplus1'-mean)./(sigma/sqrt(2)));
    pdf_down = normpdf((ud'-mean)./(sigma/sqrt(2)));
    cdf_up = normcdf((udplus1'-mean)./(sigma/sqrt(2)));
    cdf_down = normcdf((ud'-mean)./(sigma/sqrt(2)));
    e = (pdf_up - pdf_down)./(cdf_up - cdf_down);
    gradient = (sqrt(2)/sigma).*(A'*e);

    % Perform shrinkage operation
    h_z = z-step*gradient;
    normed_h = sqrt(h_z(1:G).^2 + h_z(G+1:end).^2);
    xhat(1:G) = (h_z(1:G)./normed_h).*max(0,normed_h-step*rho);
    xhat(G+1:end) = (h_z(G+1:end)./normed_h).*max(0,normed_h-step*
   rho);

    % Update FISTA parameters
    eta = (1+sqrt(1+4*eta^2))/2;
    z = xhat + ((prior_eta-1)/eta).*(xhat - prior_xhat);

    if ((max(xhat-prior_xhat)) < min_change)
        %disp("FISTA ended due to min_change restriction - " +
   datestr(now));
        break;
    elseif sum(isnan(xhat)) > 0
        %disp("ORG FISTA failed due to NaNs on iter - "+i+" - "+
   datestr(now));
        xhat = prior_xhat;
        break;
    end
end
%disp("Returning estimated xhat value - " + datestr(now));
end
```

Listing A.6: *ROCmain*

A.3.2 FISTA-1b

```
function xhat = tvt_FISTA(A_tilde, ysq, hri, sigma, step,
   max_iterations, rho)
%% FISTA - Fast Iterative Shrinkage-Thresholding Algorithm
%
```

```
%   Author: Ethan Triplett
%   Created: 6/29/22
%   Last Edited: 7/15/22
%
%   Solves a group-sparse radar detection problem using FISTA.
%
%   Inputs:
%   A - the all-real dictionary matrix [R, -I; I, R]
%   z - complex signs created from one-bit TVT quantization
%   hr - known real TVT thresholds
%   hi - known imaginary TVT thresholds
%   sigma - known noise variance
%
%   Outputs:
%   xhat - The estimated target locations
%

%% Initialization
%disp("Initializing tvt_FISTA algorithm - " + datestr(now));
grid_length = length(A_tilde(1,:));
G = grid_length/2;
eta = 1;
%rho = 140;                    % Regularization parameter
   determining sparsity
xhat = zeros([grid_length, 1]);
z = zeros([grid_length, 1]);
alpha = sqrt(2)/sigma;      % Replaces eta from MM paper

% Stop conditions
min_change = 10^-5;

%% Main FISTA loop
%disp("Beginning FISTA iterations - " + datestr(now));
for i = 1:max_iterations
    prior_xhat = xhat;
    prior_eta = eta;

    % Compute the gradient
    beta = alpha.*z;
    xri = ysq.*((A_tilde*beta)-(alpha.*hri));
    x_pdf = normpdf(xri);
    x_cdf = normcdf(xri);

    gradient = A_tilde'*-(x_pdf./x_cdf);

    % Perform shrinkage operation
    h_z = z - step.*gradient;
    normed_h = sqrt(h_z(1:G).^2 + h_z(G+1:end).^2);
    xhat(1:G) = (h_z(1:G)./normed_h).*max(0,normed_h-step*rho);
    xhat(G+1:end) = (h_z(G+1:end)./normed_h).*max(0,normed_h-step*
   rho);

    % Update the estimated sparse vector and check for convergence
    eta = (1+sqrt(1+4*eta^2))/2;
```

```
    z = xhat + ((prior_eta-1)/eta).*(xhat - prior_xhat);

    if ((max(xhat-prior_xhat)) < min_change)
        %disp("FISTA ended due to min_change restriction - " +
    datestr(now));
        break;
    elseif sum(isnan(xhat)) > 0
        %disp("FISTA failed due to NaNs - " + datestr(now));
        xhat = prior_xhat;
        break;
    end
end
%disp("Returning estimated xhat value - " + datestr(now));
end
```

Listing A.7: *ROCmain*

A.4 Generic Functions

This section contains all support functions not specific to the NMSE or ROC code.

```
function sparse_x = create_sparse_x(target_data,dim)
%% create_sparse_x
%
%   Creates a sparse vector of length dim, with non-zero indices
    provided
%   by the first column of target_data, and amplitudes provided by
    the
%   second column
%
if ~isscalar(dim)
    len = length(dim);
else
    len = dim;
end

sparse_x = zeros([len,1]);
sparse_x(target_data(:,1)) = target_data(:,2);
% The targets are directly on the grid for now, in the paper they
    are not
```

Listing A.8: *ROCmain*

```
%% get_quantized
%
%   Author: Ethan Triplett
%   Created: 6/31/22
%   Last Edited: 7/5/22
%
%   This code provides quantized versions of a measurement vector
    based on
%   the bit depth provided. One-bit depth is a special case that
    provides
%   values for both time varying threshold quantization and regular
    fixed
```

```
%   threshold quantization.
%
%


function [yq_tvt, h, yq_org1, ud1, udplus1_1, yq_org2, ud2,
    udplus1_2] = ...
                get_quantized3(y, q, num_tvt)

rmax = 0.95 * max(abs(y));
delta_s = rmax/(2^q);

tvt = [rmax*0.2, rmax*0.4, rmax*0.8];
h = tvt(randi([1 num_tvt] ,size(y)))';
yq_tvt = sign(y - h);

partition = [0];     % Only threshold without TVT
codebook = [-delta_s, delta_s];
[index,yq_org1] = quantiz(y,partition,codebook);

partition_w_inf = [-inf, partition, inf];
ud1 = partition_w_inf(index+1);
udplus1_1 = partition_w_inf(index+2);

q_bits = 2;
D = 1:(2^q_bits)-1;
partition = (-2^(q_bits-1)+D).*delta_s;
codebook = partition + (delta_s/2);
codebook = [-codebook(1,end) codebook];
[index,yq_org2] = quantiz(y,partition,codebook);

partition_w_inf = [-inf, partition, inf];
ud2 = partition_w_inf(index+1);
udplus1_2 = partition_w_inf(index+2);
```

Listing A.9: *ROCmain*

```
%% get_error
%
%   Author: Ethan Triplett
%   Created: 6/15/22
%   Last Edited: 7/20/22
%
%   This code displays the complex magnitudes of the original signal
    x vs
%   the three signal estimates. The SNR level is added to the
    filename
%   while saving plots.
%

function display_real_v_3est(x, xhat1, xhat2, xhat3,snr)

disp("Plotting real vs estimated data - " + datestr(now));
figure1 = figure;
```

```
xp = plot(x);
xp.Color = 'black';
xp.Marker = 'o';
hold('on')
xh1 = plot(xhat1);
xh1.Color = 'red';
xh1.LineStyle = '--';
xh1.Marker = 'x';
hold("on")
xh2 = plot(xhat3);
xh2.Color = 'green';
xh2.LineStyle = ':';
xh2.Marker = 's';
hold("on")
xh3 = plot(xhat2);
xh3.Color = 'blue';
xh3.LineStyle = '--';
xh3.Marker = 'p';
hold("off")

title("Signal Estimation vs Real Signal")
xlabel("Target Grid Location")
ylabel("Target Signal Amplitude")
legend("True Signal", "FISTA-1B", "2-bit FISTA","1-bit FISTA")

saveas(figure1,"output-"+round(snr)+".jpg");
saveas(figure1,"output-"+round(snr)+".fig");
```

Listing A.10: *ROCmain*