ABSTRACT

Design and Implementation of a Synthetic Aperture Radar (SAR) Imaging System
Using an Ultra-Wideband (UWB) Radar Sensor

Brian Thomas Ernzen, M.S.E.C.E.

Mentor: Yang Li, Ph.D.

This thesis documents the design and implementation of a radar imaging system. A rail system was constructed to control the location of an off-the-shelf ultra-wideband (UWB) radar along a linear path. Range imaging, cross-range imaging, and synthetic aperture radar (SAR) imaging algorithms were implemented in Matlab based on the data from the radar. Results are presented for a variety of target scenes. The radar imaging system can be used for future research of radar imaging algorithms as well as for classroom demonstration for a radar course at Baylor University.

Design and Implementation of a Synthetic Aperture Radar (SAR) Imaging System Using an Ultra-Wideband (UWB) Radar Sensor

by

Brian Thomas Ernzen, B.S.E.E.

A Thesis

Approved by the Department of Electrical and Computer Engineering

____

Kwang Y. Lee, Ph.D., Chairperson

Submitted to the Graduate Faculty of
Baylor University in Partial Fulfillment of the
Requirements for the Degree
of

Master of Science in Electrical and Computer Engineering

Approved by the Thesis Committee

____

Yang Li, Ph.D., Chairperson

____

B. Randall Jean, Ph.D.

____

Brian A. Garner, Ph.D.

Accepted by the Graduate School

May 2016

____

J. Larry Lyon, Ph.D., Dean

Page bearing signatures is kept on file in the Graduate School.

Copyright © 2016 by Brian Thomas Ernzen

All rights reserved

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

cm	Centimeter
dB	Decibel
dBm	Decibel-milliwatt
FCC	Federal Communications Commission
GHz	Gigahertz
ISAR	Inverse synthetic aperture radar
LPE	Lowpass equivalent
MHz	Megahertz
mm	Millimeter
ns	Nanosecond
PC	Personal computer
PII	Pulse integration index
RF	Radio frequency
RX	Receive
SAR	Synthetic aperture radar
SNR	Signal-to-noise ratio
TX	Transmit
USB	Universal Serial Bus
UWB	Ultra-wideband

ACKNOWLEDGMENTS

I would like to thank Dr. Yang Li, my advisor, for his support and guidance. Since I completed this work as a part time graduate student with a full time industry job, the progress was slow at times. I was not sure that I would ever finish, but Dr. Li was always confident that I would. I would like to thank my committee members, Dr. Randall Jean and Dr. Brian Garner, for their feedback and support. I would like to thank my employer, L-3, for providing a large portion of my tuition for my master's degree. I would like to thank my wife, Julie Ernzen, for providing some grammatical editing suggestions. Julie brought my widespread misuse of commas to my attention. I would like to thank Amanda Gardner for her assistance with formatting of this thesis and for providing some grammatical editing suggestions. I would like to thank Ph.D. candidate Willis Troy for valuable assistance during this project. Willis and I had many discussions about my project, and he helped with a USB communication problem and several other computer related problems that I encountered. I would like to thank Ph.D. candidates Wenlei Bai and Bin Xu for assistance with mapping out the target locations used to produce the results presented in this thesis. I would like to thank my parents, Tom and Jan Ernzen, for their help during the collection of the radar data used to produce the results presented. Also, my dad provided advice for the mechanical aspects of the design and helped with some of its construction. I would like to thank Brandon Ernzen, my brother, for loaning me his trailer to transport this project. I am very thankful for the many friendly and welcoming faculty members in the Engineering department that I have gotten to know during my time at Baylor. I am also very thankful for the many students and former students with whom I have formed friendships, including Josh Martin, Matthew Moldovan, Dr. Mohammad Reza Abedi, Mehdi Ebad, Josh Daniliuc, Wenlei Bai, Bin Xu, Willis Troy, David Moon, Bochuan Liu, and many others.

I would again like to thank my parents, Tom and Jan Ernzen, for their love and support. I would again like to thank my wife, Julie Ernzen, for her love and support. I am looking forward to spending more time with her and our daughter Emma.

DEDICATION

To my wife, Julie Ernzen, our daughter, Emma,

my parents, Tom and Jan Ernzen,

and

my friends who have gone before,

Robert J. "Bob" Province (1937-2015) and Earl Ray Snow (1948-2015)

CHAPTER ONE

Introduction

A pulsed radar is able to determine the range of objects from the radar by transmitting electromagnetic energy in the form of a pulse and then measuring the time delay of echoes from the objects. Synthetic aperture radar (SAR) uses the motion of a radar to produce additional spatial data about the objects, such as a 2-D image. As recounted in [1], SAR was first proposed by Carl Wiley in 1951. By 1953, aircraft based SAR was used to map a section of Florida [2]. Since then, SAR has been used on satellite and space shuttle platforms [2]. SAR has been used in many military applications such as detection of buildings, tanks, and aircraft [2]. Civilian applications include crop monitoring [2], [3] and imaging of Earth's surface [4]. SAR has several advantages over optical methods of image formation. SAR does not require that the object being imaged be illuminated by a light source [4]. Also, with the right choice of frequencies, SAR can be used in cloudy conditions [4]. Soumekh recounts in [3] that the theory on which SAR is based can be traced back to Gabor's paper [5] from 1948 which introduced the theory of wavefront reconstruction. It was not until the 1980s that fast computers and digital signal processing techniques existed to implement SAR with wavefront reconstruction [3]. Until then, SAR implementations had to rely upon approximation techniques such as optical processing using the Fresnel approximation and polar format processing using the plane wave approximation [3].

The radar used in this thesis is an ultra-wideband (UWB) pulse radar. In the United States, the Federal Communications Commission (FCC) regulates and defines unlicensed UWB transmission systems. The FCC regulations and definitions for these systems are found in [6]. The following definitions are from [6]. "UWB bandwidth. For the purpose of this subpart, the UWB bandwidth is the frequency band bounded by the points that are 10 dB below the highest radiated emission, as based on the complete transmission system including the antenna. The upper boundary is designated f_H and the lower boundary is designated f_L. The frequency at which the highest radiated emission occurs is designated f_M." "Center frequency. The center frequency, fc, equals $(f_H + f_L)/2$." "Fractional bandwidth. The fractional bandwidth equals $2(f_H - f_L)/ (f_H + f_L)$." "Ultra-wideband (UWB) transmitter. An intentional radiator that, at any point in time, has a fractional bandwidth equal to or greater than 0.20 or has a UWB bandwidth equal to or greater than 500 MHz, regardless of the fractional bandwidth." [6] UWB pulse radars have several advantages over non UWB radars. UWB radars have high range resolution since range resolution is inversely proportional to bandwidth [7]. Since the transmitted energy is spread over a wide frequency range, the UWB pulse has a low level of interference with other RF systems [7]. Also, UWB pulse radars have a high level of immunity to jamming [7]. The PulsON P410, the UWB radar used in this thesis, has been used by a number of other researchers for many interesting applications such as the detection of human breathing [8], detection of apnea in infants [9], ISAR imaging [10], [11], and SAR imaging using a drone [12].

This thesis documents the design and implementation of a radar imaging system. The system combines SAR and UWB capability by providing the capability to control the motion of an off-the-shelf UWB radar along a linear path. The system is controlled by a Matlab program executed on a PC. Data collected by the radar is saved by the Matlab program for later processing. The system was designed and implemented to be utilized for educational and research purposes at Baylor University. It is intended that the system will be used to provide classroom demonstration in the course ELC 5340 “Radar Engineering” offered in the Electrical Engineering Department at Baylor University. Students can participate in the collection of data using the radar imaging system. The data collected can then be used by the students so that they can implement radar signal processing algorithms such as those presented in this thesis. For research purposes, the system can be utilized for the development of advanced and novel radar imaging algorithms by Baylor University researchers.

This thesis is organized as follows: Chapter One presented an introduction to synthetic aperture radar and ultra-wideband technology and described the motivation for the work presented in this thesis. Chapter Two describes the radar imaging system including the hardware and software. Chapter Three describes a range imaging algorithm based on Chapter 1 of [3] and presents results from a Matlab implementation of the algorithm and data collected by the radar imaging system. Chapter Four describes a cross range imaging algorithm based on Chapter 2 of [3] and presents results from a Matlab implementation of the algorithm and data collected by the radar imaging system. Chapter Five describes a synthetic aperture radar imaging algorithm based on Chapter 4 of [3] and presents results from a Matlab implementation of the algorithm and data collected by the radar imaging system. Chapter Six presents ideas for the improvement of the radar imaging system, the improvement of the algorithms presented, or for small projects suitable for homework assignments for a radar engineering course.

CHAPTER TWO

Radar Imaging System

The radar imaging system of this thesis consists of a UWB radar and two antennas, a rail system to control the motion of the radar and the antennas, and Matlab software, executed from a PC, to interface with the radar and the rail system. The radar and the antennas, the rail system, and the software are described in Section 2.1, Section 2.2, and Section 2.3, respectively. In addition to the components of the rail system, Section 2.4 describes the targets used in target scenes to produce the results presented in Chapter Three, Chapter Four, and Chapter Five of this thesis.

2.1 PulsON 410 UWB Radar

The UWB radar used in this thesis is the PulsON 410 (P410) made by Time Domain. The P410, shown in Figure 2.1, has an operating band from 3.1 GHz to 5.3 GHz with a center frequency of 4.3 GHz [13]. The radar has a transmit power that is adjustable from -31.6 dBm to -12.64 dBm [13]. Two antenna ports are used to connect to a transmit antenna and a receive antenna [13]. In the work presented in this thesis, the transmit and receive antennas used are horn antennas.

Figure 2.1 P410 UWB Radar

The radar transmits an electromagnetic pulse from the transmit antenna and receives electromagnetic energy from the receive antenna. Figure 2.2 shows the P410 transmit pulse waveform based upon measurement data produced by Time Domain available at [14]. Although different antennas are used in the work of this thesis than were used by Time Domain, it is expected by the author that the pulse waveform of the radar in this thesis will be very similar to the waveform in Figure 2.2.

The radar has the ability to improve the signal-to-noise ratio (SNR) of the received signal by sending multiple pulses at a pulse repetition rate of 10.1 MHz. The radar then coherently integrates the received signals due to each pulse to form a received signal with a higher SNR than the received signals due to each individual pulse. The

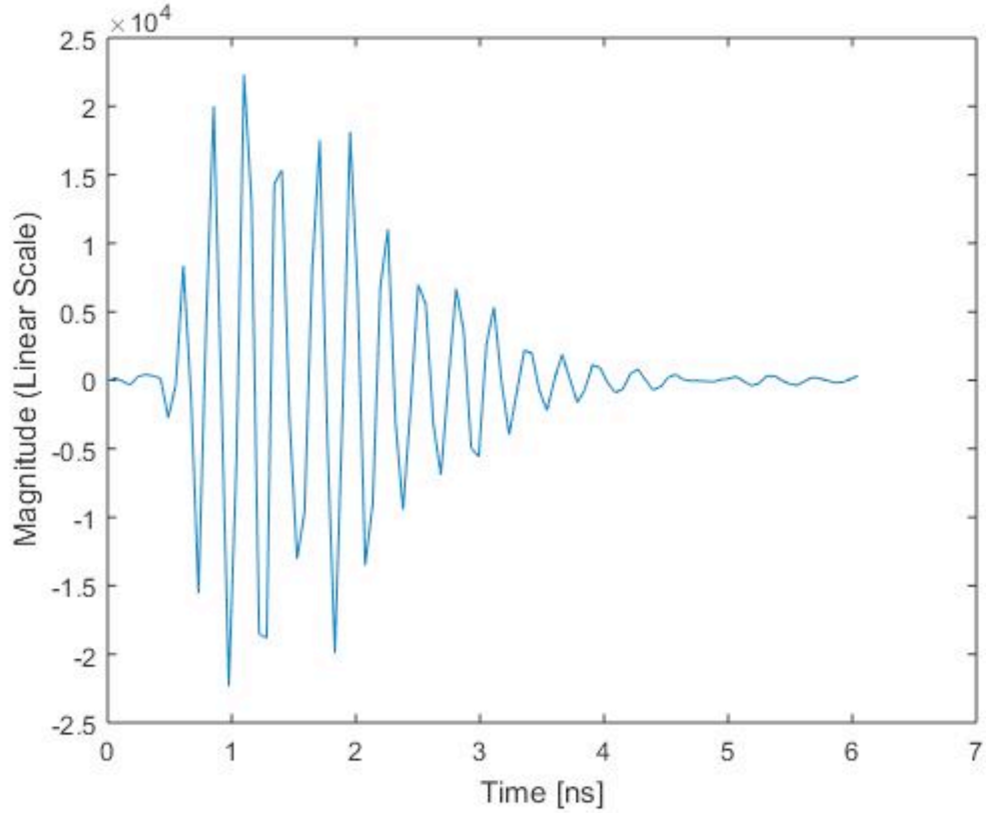

Figure 2.2 P410 Transmit Pulse Waveform

number of pulses that the P410 coherently integrates is given by (1), where PII is the pulse integration index.

$$Pulses = 2^{PII} \tag{1}$$

The P410 radar allows PII to be any integer from 6 to 15. Increasing the value of PII by one doubles the number of pulses that are coherently integrated and increases the received signal's SNR by 3dB [13], [15].

In the radar imaging system of this thesis, a Matlab program, Data_Collection.m, described in more detail in Section 2.3, is used to configure the desired parameters of the P410 and to command it to transmit pulses and receive return signals. The received electromagnetic signals are sampled by the P410 and the data is saved by the Matlab program as a data file for later processing.

2.2 Rail System

As described in Section 2.1, the P410 is capable of transmitting a series of pulses and coherently integrating the return signals from each pulse to form a signal with a high SNR. As described in Chapter Three, this return signal can be used with an appropriate signal processing algorithm to find the range of objects from the radar. In order to gain additional spatial information about the objects, such as cross range or a two dimensional synthetic aperture radar image, described in Chapter Four and Chapter Five respectively, it is required that the process of transmitting pulses and receiving return signals from the objects be performed at multiple radar locations along a linear path. To satisfy this requirement, a rail system was designed and implemented to provide accurate positioning of the radar and the antennas along a three meter length. This allows the radar to transmit pulses and acquire a coherently integrated receive signal at a particular radar position. The radar position can then be changed and the transmit/receive process can be repeated at the new position.

The rail system is shown in Figure 2.3. The major components of the rail system are a belt drive actuator, a stepper motor, a stepper drive, a power supply to provide power to the stepper drive, a cart on which the radar and the antennas are mounted, and an aluminum frame to support the system and to guide the rollers of the cart along the length of the system. The rail system was designed to provide three meters of total movement of the radar. Location markers on the aluminum frame designate -1.5 meters towards one end, 0 meters in the center, and +1.5 meters towards the other end. The general locations of these three markers are indicated in Figure 2.3.

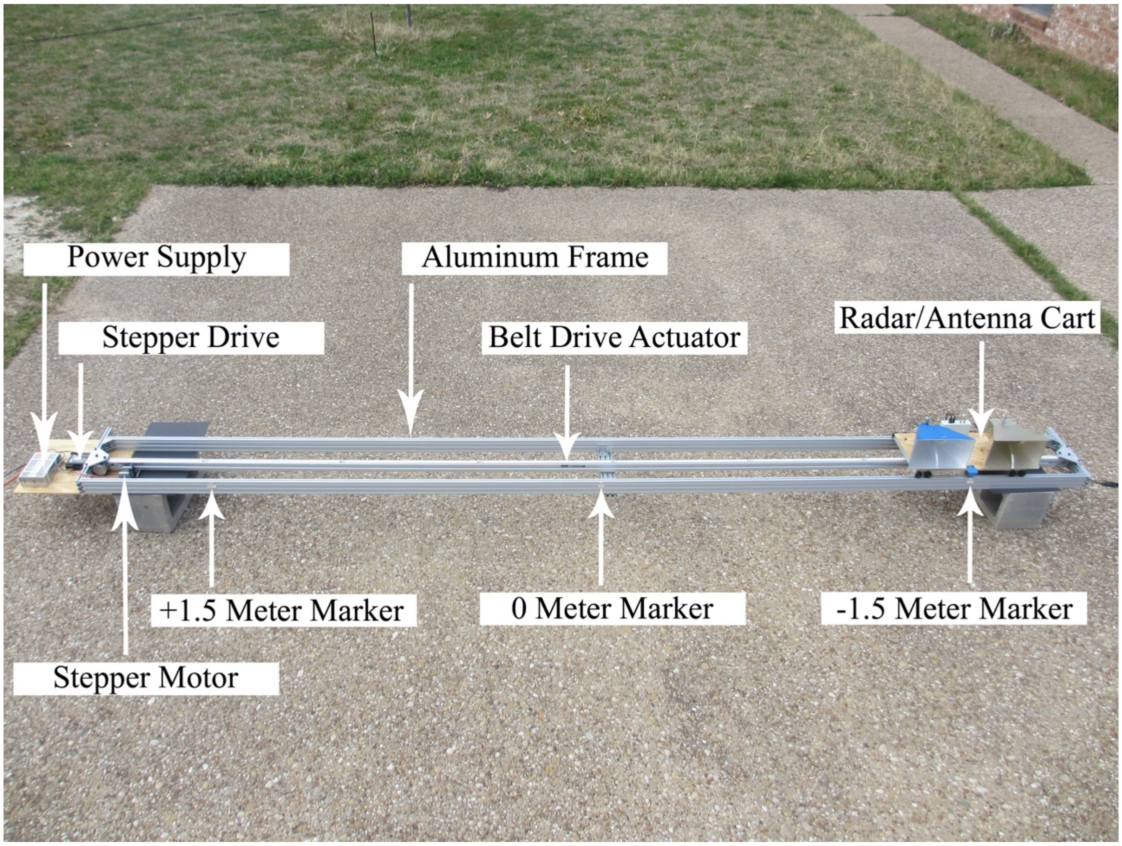

Figure 2.3 Rail System

The cart on which the radar and the antennas are mounted is detailed in Figure 2.4 and Figure 2.5. Figure 2.4 is the front view and Figure 2.5 is the rear view. The cart has rollers that are guided by the aluminum frame. The blue pointer on the front of the cart, shown in Figure 2.4, provides a way to manually align the cart at the -1.5 meter marker on the aluminum frame prior to use. The cart is attached to the belt on the belt drive actuator so that when the belt moves, the cart moves along with it.

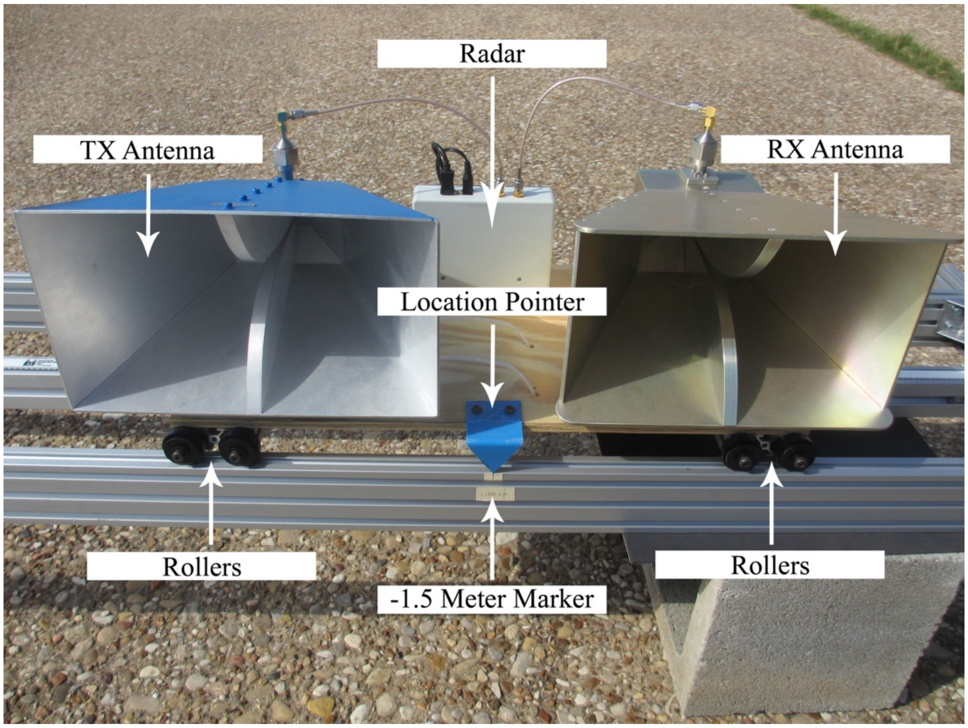

Figure 2.4 Front view of Cart for the Mounting of Radar and Antennas

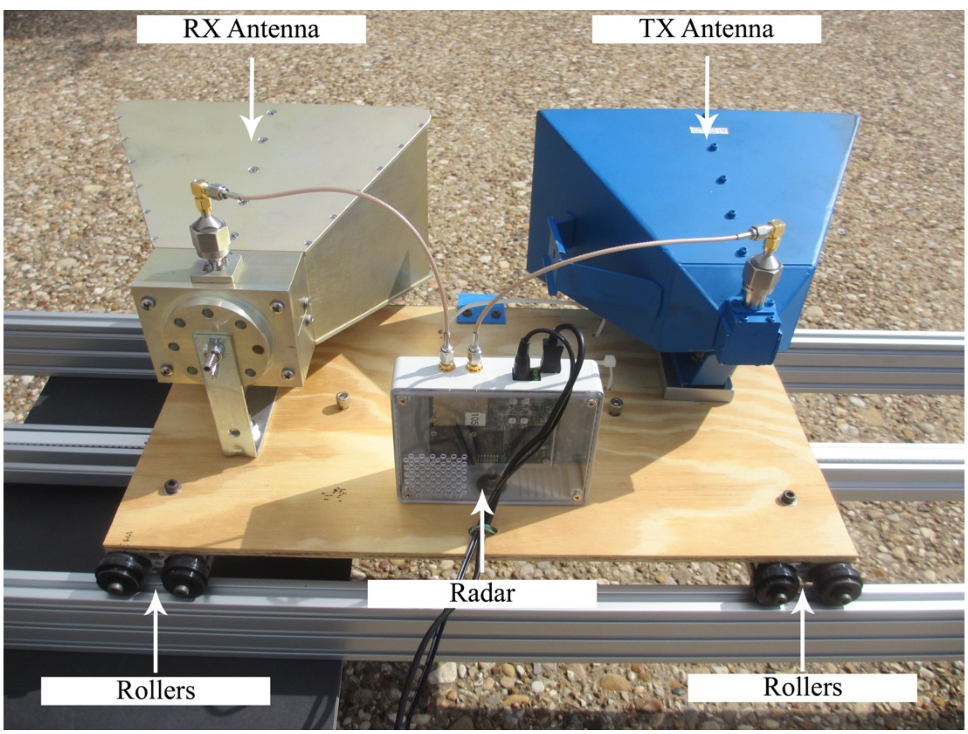

Figure 2.5 Rear View of Cart for the Mounting of the Radar and the Antennas

The stepper motor is the HT17-275 stepper motor from Applied Motion Products. The stepper motor and belt drive actuator are connected mechanically with two pulleys and a belt as detailed in Figure 2.6. Rotation of the stepper motor causes the belt drive actuator's belt to move which in turn moves the cart with it.

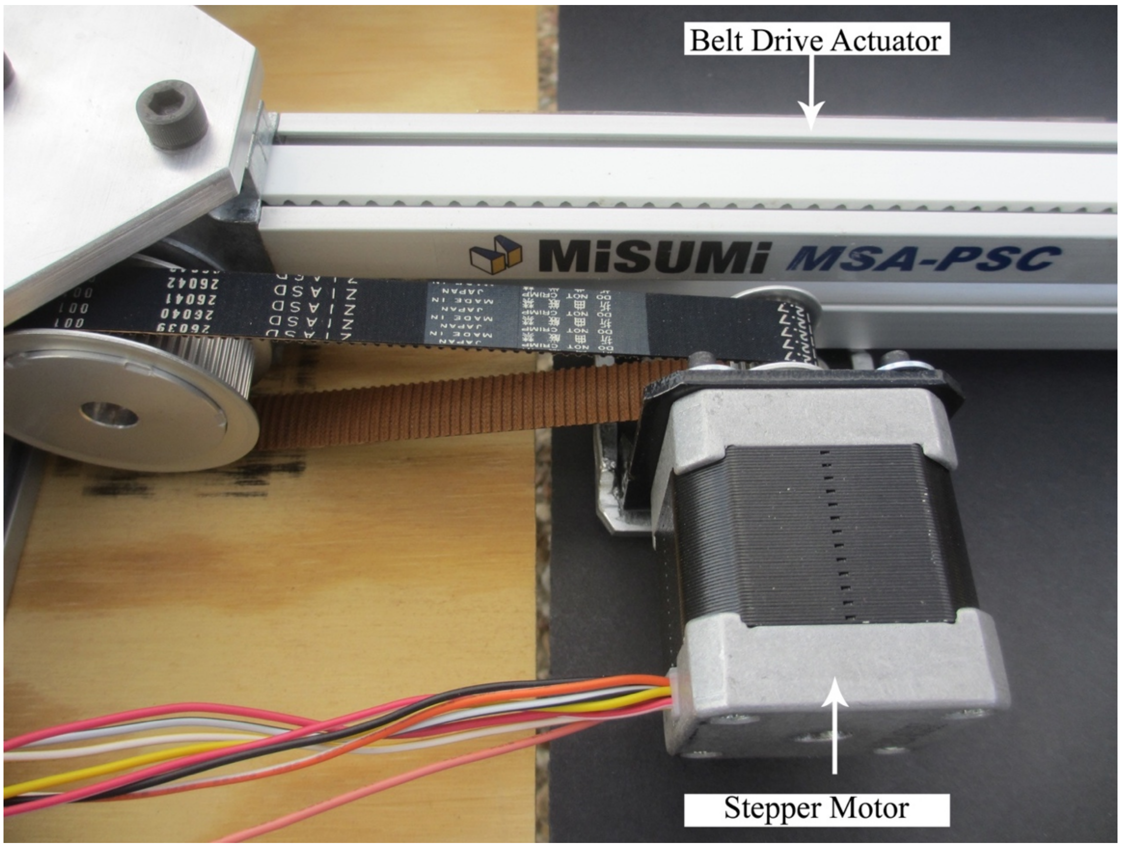

Figure 2.6 Stepper Motor and Belt Drive Actuator Connection with a Belt and Two Pulleys

The HT17-275 stepper motor is controlled by the ST5-Plus stepper drive from Applied Motion Products. The stepper drive receives power from the PS150A24 power supply from Applied Motion Products. The stepper drive and the power supply, along with the stepper motor, are shown detailed in Figure 2.7.

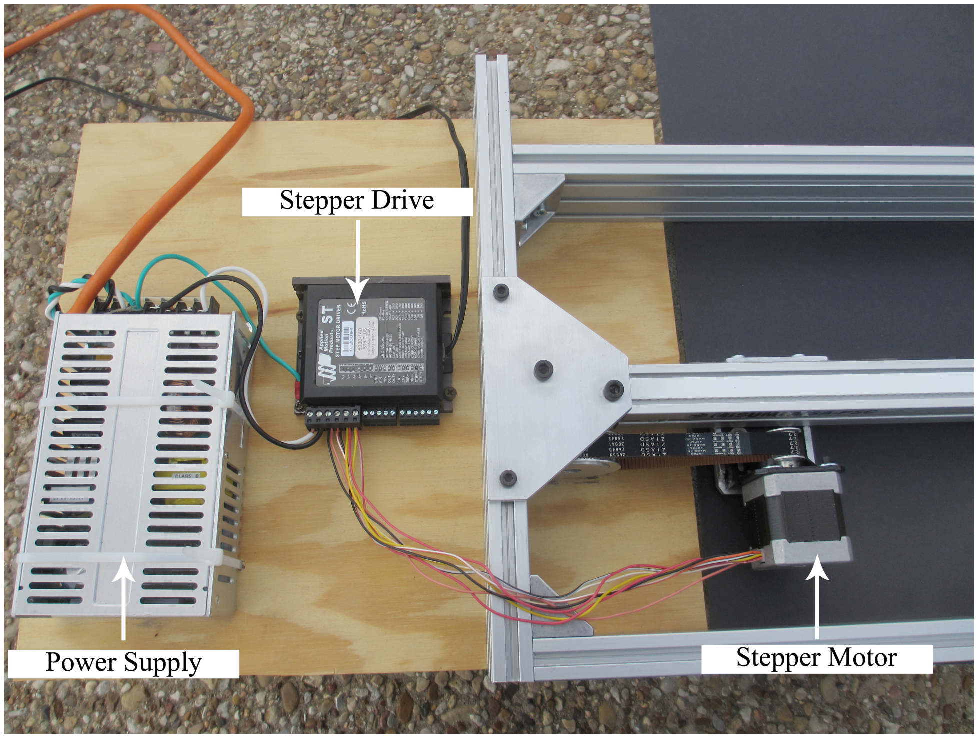

Figure 2.7 Stepper Motor, Stepper Drive, and Power Supply

The stepper drive receives high level commands through RS-232 communication from the Matlab program, Data_Collection.m, briefly mentioned in Section 2.1 and described in more detail in Section 2.3. The high level commands include commands to set the acceleration, the velocity, and the number of steps to rotate [16]. Since the PC used in this work did not have an RS-232 communication port, a USB to RS-232 serial adapter was used. The ST5-Plus is able to control the stepper motor to rotate in increments of 1 step where each revolution of the motor has 20,000 steps [16]. It was found experimentally that 1,788,000 motor steps are required to move the radar along the 3 meter travel distance. By dividing 3 meters by 1,788,000 motor steps, it is found that the radar can be moved in increments as small as a few micrometers by rotating the motor by one step. For the results presented in this thesis, the Matlab program, Data_Collection.m, was used to move the radar a total distance of 3 meters in 1 cm increments. By dividing 1,788,000 steps by 300, it is found that 5960 motor steps are required for a 1 cm movement. Data_Collection.m provides additional options to the user to move in increments of 1 mm, 0.5 mm, and 0.25 mm.

Table 2.1 summarizes the major components used for the rail system. Table 2.1 is not intended to be a complete parts lists since many items such as bolts, washers, and the structural aluminum for the rail frame are not listed.

Table 2.1 Major Components of Rail System

Item	Part Number or Model Number	Manufacturer
Belt Driven Actuator	MSA-PSC-0-GB-A1-04000	Misumi
Stepper Drive	ST5-Plus	Applied Motion Products
Stepper Motor	HT17-275	Applied Motion Products
Power Supply	PS150A24	Applied Motion Products
USB to RS232 serial adapter	3004-235	Applied Motion Products

2.3 Software

The Matlab program, Data_Collection.m, documented in Appendix A, was written to interface with the stepper drive and the P410 UWB for the collection of radar data. The data can be used for further signal processing by algorithms such as those described in Chapter Three, Chapter Four, and Chapter Five of this thesis. Prior to running this Matlab program, the radar should be set at the starting point by aligning the cart's blue pointer with the -1.5 meter marker on the aluminum frame.

The program has a number of variables to be adjusted by the user. The user should set the values of the variables radar_com_port_num and motor_com_port_num appropriately to reflect the communications ports used to communicate with the radar and the stepper drive, respectively. The values of the variables R1 and R2 should be set based upon the target scene. R1 and R2 set the desired minimum and maximum range for the returned radar data. The actual minimum and maximum ranges of the returned data will be slightly different than the requested values due to the way that the radar samples the received signal [15]. The user may need to experiment with these values to get the desired results. The variable Gtx sets the transmitted power. Gtx can be set to any integer from 0 to 63, where a value of 0 corresponds to the minimum transmit power of -31.6 dBm and a value of 63 corresponds to the maximum transmit power of -12.64 dBm [13], [15]. For the results presented in this thesis, a value of 63 was used for Gtx. The variable PII can be set to any integer value from 6 to 15 and sets the pulse integration index value that was discussed in Section 2.1. For the results presented in this thesis, a value of 10 was used for PII. The variable dT0 is used to calibrate the system based on the antennas and the length of coax cable connecting the radar and the antennas. To set this value, a target should be set at a known range from the radar. The value of dT0 in Data_Collection.m can then be adjusted so that the collected data reflects the correct range of the target. Several iterations of running Data_Collection.m may be needed to find the best value. For the antennas and coax cable used in this thesis, it was found that a dT0 value of 14.11 produced suitably accurate range results. This value is used for all of the results presented in this thesis. The variable Antenna is a string that can be set to reflect the antennas used. The string variable File_Name sets the desired name of the file to which Data_Collection.m will save data for later processing. The variables Ktry_max1, Ktry_max2, Ktry_max3, and Ktry_max4 control timeout limits for four USB communication instances between the radar and Matlab. A value of 1000 was found to be a reasonable value for these variables. The radar moves a total distance of 3 meters along the rail system stopping at evenly spaced intervals along the way. The variable Step_Size is used to set the distance in meters of these intervals. The results presented in this thesis were produced with a Step_Size value of 0.01. Acceptable values for Step_Size are 0.01, 0.001, 0.0005, and 0.00025.

Once the user adjustable variables are set, the program can be run. For the following explanation of program operation, a value for Step_Size of 0.01 is assumed. At the initial radar location of -1.5 meters on the rail system, Data_Collection.m commands the radar to transmit and receive electromagnetic signals as described in Section 2.1. The coherently integrated received signal is then stored by Data_Collection.m as the first row of a matrix variable called SCN_MATRIX. Data_Collection.m then commands the stepper drive to rotate the stepper motor by 5960 motor steps which causes the radar to move 1 cm along the rail system. After stopping at this location, Data_Collection.m again commands the radar to transmit and receive electromagnetic signals. The received signal from this location is stored by Data_Collection.m as the second row of the variable SCN_MATRIX. This process continues until the radar reaches the last location at +1.5 meters on the rail system. At this location, Data_Collection.m commands the radar to transmit and receive electromagnetic signals. The received signal from this last location is stored by Data_Collection.m as the last row of the variable SCN_MATRIX. SCN_MATRIX now has a total of 301 rows. Data_Collection.m then commands the stepper drive to rotate the stepper motor by 1,788,000 steps to move the radar 3 meters back to the -1.5 meter marker.

The last thing Data_Collection.m does is to save a Matlab data file with the name assigned to the string variable File_Name. This data file contains the user defined variables Gtx, PII, dT0, and Antenna. The file also contains SCN_MATRIX, Rbin, Taxis, and radar_position_column. Rbin is a row vector variable that is used as a range axis for the data of each row of SCN_MATRIX. Taxis is a row vector variable that is used as a time axis for the data of each row of SCN_MATRIX. Taxis is a scaled version of Rbin. The scaling between the time axis and the range axis is discussed in Chapter Three. The variable radar_position_column is a column vector with elements corresponding to each row of SCN_MATRIX. This column vector associates each row of SCN_MATRIX with a location of the radar along the rail system. The data file can be stored by the user for later use with signal processing algorithms such as the range imaging, cross range imaging, and synthetic aperture radar (SAR) imaging algorithms presented in Chapter Three, Chapter Four, and Chapter Five, respectively.

The Matlab program Data_Collection.m uses several functions written by Time Domain. The function open_com_port, written by Time Domain, is used to establish a USB port to communicate with the radar. Each time Data_Collection.m commands the radar to transmit and receive electromagnetic signals, it does so by executing the function plot_one_scn.m, written by Time Domain. Time Domain's Matlab function, plot_one_scn.m, is dependent on eight other Matlab functions written by Time Domain. Some of the functions written by Time Domain were modified in this work to allow all of the user adjustable variables to be created at the beginning of Data_Collection.m and to be passed as arguments from Data_Collection.m to Time Domain's Matlab functions as required. Some modifications were also made to Time Domains's Matlab functions to ensure that all required data was returned to Data_Collection.m

2.4 Radar Targets

In this thesis, objects for which it is desired to gain spatial information about, such as range information, will be referred to as targets. The area containing an arrangement of targets will be referred to as a target scene. The results presented in Chapter Three, Chapter Four, and Chapter Five of this thesis use the Matlab program Data_Collection.m to collect data from various target scenes. The targets used in this thesis are steel pipe with a length of 61 cm and an outer diameter of 7.3 cm. Figure 2.8 shows an example of one of these targets.

Figure 2.8 Steel Pipe Used as a Target

CHAPTER THREE

Range Imaging

A radar transmits electromagnetic energy which may impinge upon targets causing a portion of the transmitted energy to return to the radar as an echo signal. For transmitted energy in the form of a finite duration pulse, the range of the targets from the radar can be processed by measuring the time delay between the transmitted pulse and the received echo signals. Range refers to the distance of a target from the radar, regardless of direction. The process of determining the range of the targets in a target scene is range imaging. This chapter describes a range imaging algorithm from Chapter 1 of [3] and presents the results from the Matlab implementation of the range imaging algorithm for several target scenes. Appendix B documents the Matlab program Range_Imaging.m developed for this thesis. Range_Imaging.m uses data acquired with Data_Collection.m, described in Chapter Two of this thesis.

Recall from Chapter Two that Data_Collection.m collects radar returns at intervals along the length of the rail system. Figure 3.7, Figure 3.15, Figure 3.25, and Figure 3.33 of this chapter depict the radar at the initial position at the -1.5 meter marker on the rail system. In this chapter the target ranges refer to the ranges of targets from the radar when the radar is at the center position at the 0 meter marker on the rail system.

3.1 Range Imaging Basics

Consider Figure 3.1. In Figure 3.1, a target is located at an unknown range R from the radar. The radar has a transmit antenna and a receive antenna designated in

Figure 3.1 as TX and RX, respectively. In the work presented in this thesis, the transmit antenna and receive antenna are very close together such that they are considered to be co-located. This is a valid assumption if the range of the target is much larger than the distance between the antennas since the distance from each antenna to the target will be approximately the same.

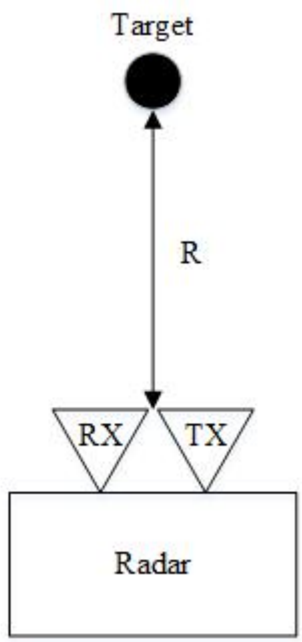

Figure 3.1 Range Imaging of a Target with a Range R

Figure 3.2 shows the transmitted signal p(t) and the received signal s(t) along a time axis. The transmitted signal p(t) contains a pulse that is transmitted by the TX antenna at t = 0. Then at time t_R, which occurs Δt later, the echo pulse is received by the RX antenna.

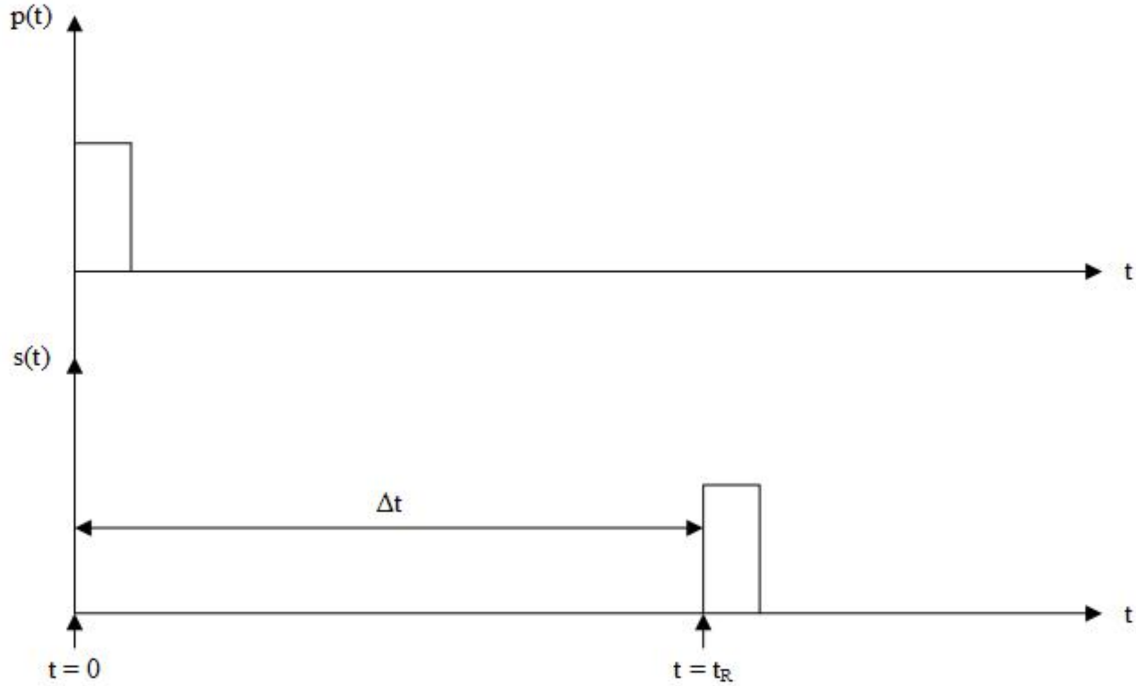

Figure 3.2 Range Imaging Transmitted Pulse and Echo Pulse with a Target at a Range R

Considering that the transmitted pulse and the echo pulse travel at the speed of light, c = $3*10^8$ m/s, and the sum of the distances traveled by the transmitted pulse and the echo pulse is 2*R, it can be shown that the range R of the target from the radar can be calculated by (1).

$$R = \frac{c\Delta t}{2} \tag{1}$$

Since the transmitted pulse occurs at t = 0, Δ t in (1) can be replaced t_R. Then, after dropping the subscript from t_R, the range R of the target from the radar can be calculated by (2).

$$R = \frac{ct}{2} \tag{2}$$

Due to the relationship between R and t given in (2), every value on the time axis of s(t) corresponds to a range. The signal s(t), from the bottom of Figure 3.2, can now be redrawn as Figure 3.3 with the exact same shape but with the axis relabeled in terms of the corresponding range.

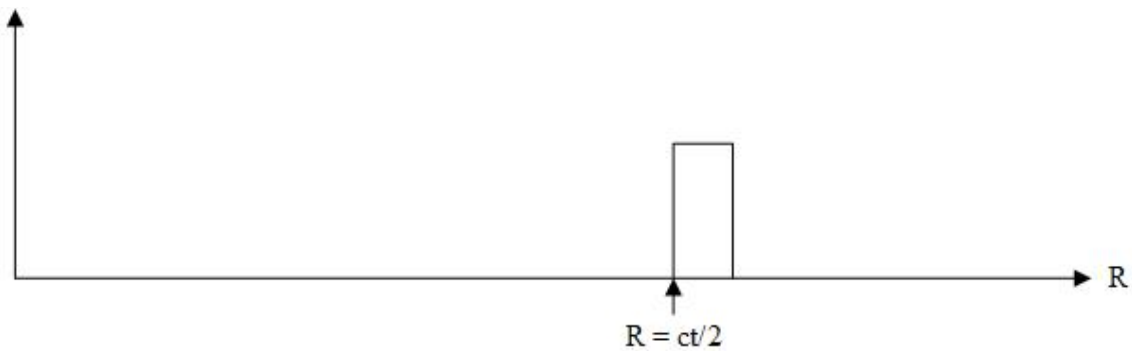

Figure 3.3 Range Imaging Echo Pulse in Terms of Range with a Target at a Range R

In Figure 3.3, the range of the target has been determined but not the direction of the target from the radar. The target could have been located at any point with a distance R from the antenna and the same results would be obtained, provided the target was within the beamwidths of the antennas.

In Figure 3.4, two targets, designated as Target 1 and Target 2, are shown at ranges R_1 and R_2 from the radar, respectively. In this case, a single transmitted pulse from the TX antenna would result in two echoes being received by the RX antenna. This is shown in Figure 3.5 and Figure 3.6, where t_1 and t_2 are the times of arrival of the echoes from Target 1 and Target 2, respectively.

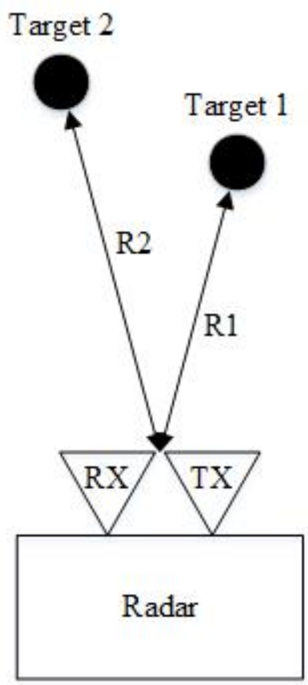

Figure 3.4 Range Imaging of Two Targets at Ranges of R_1 and R_2

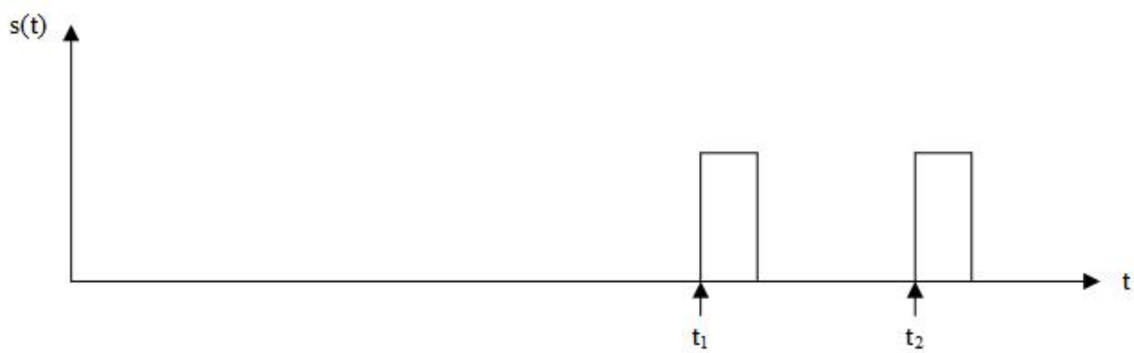

Figure 3.5 Echo Pulses with Two Targets at Ranges of R_1 and R_2

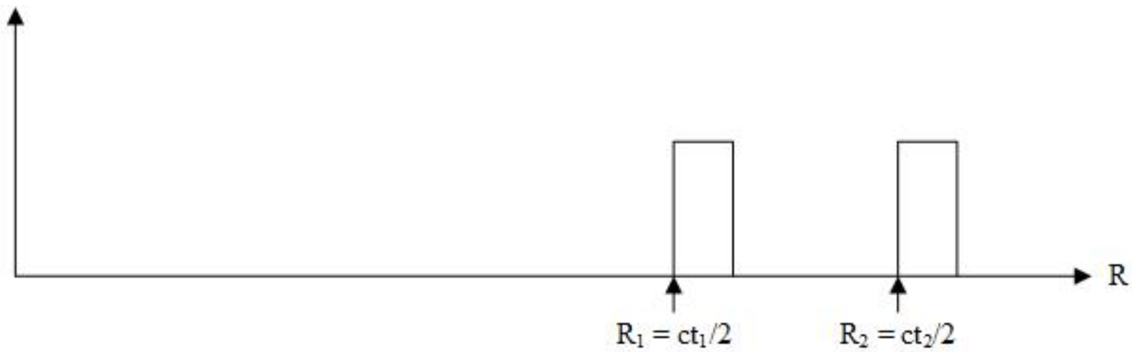

Figure 3.6 Echo Pulses in Terms of Range with Two Targets at Ranges of R_1 and R_2

The transmitted pulse of p(t) shown in Figure 3.2 is represented as a rectangle. This is an idealization. The rectangle represents the window of a real transmitted pulse. The return signals represented in Figure 3.2, Figure 3.3, Figure 3.5, and Figure 3.6 are also idealizations. The rectangles in those figures represent the windows of the echos from the targets. Section 3.2 will describe the range imaging algorithm in more detail. In doing so, the actual signals received by the PulsON 410 used in this thesis will be presented. An actual pulse transmitted by the PulsON 410 was shown in Figure 2.2 of Chapter Two.

3.2 Range Imaging Algorithm

Section 3.1 presented the basics of range imaging. This section describes the details of the range imaging algorithm by looking at the results of the range imaging algorithm for the target scene shown in Figure 3.7. In Figure 3.7, two targets are placed in the target scene. One target is placed at a range of 3.31 meters from the radar and the other is placed at a range of 4.21 meters from the radar. The goal of the range imaging algorithm is to determine the ranges of the two targets.

Figure 3.7 Range Imaging Target Scene with Targets Located at Ranges of 3.31 meters and 4.21 meters

The radar transmits a signal p(t) containing a pulse. A representative pulse waveform transmitted by the PulsON 410 was shown in Figure 2.2 of Chapter Two. The radar then receives a return signal s(t) containing return echoes from the targets. The return signal s(t), acquired by the PulsON 410, is shown in Figure 3.8. Figure 3.9 is the same waveform as shown in Figure 3.8 except the axis is relabeled in terms of range according to (2). The location of the targets can be observed in Figure 3.9 by observing the locations of large amplitude.

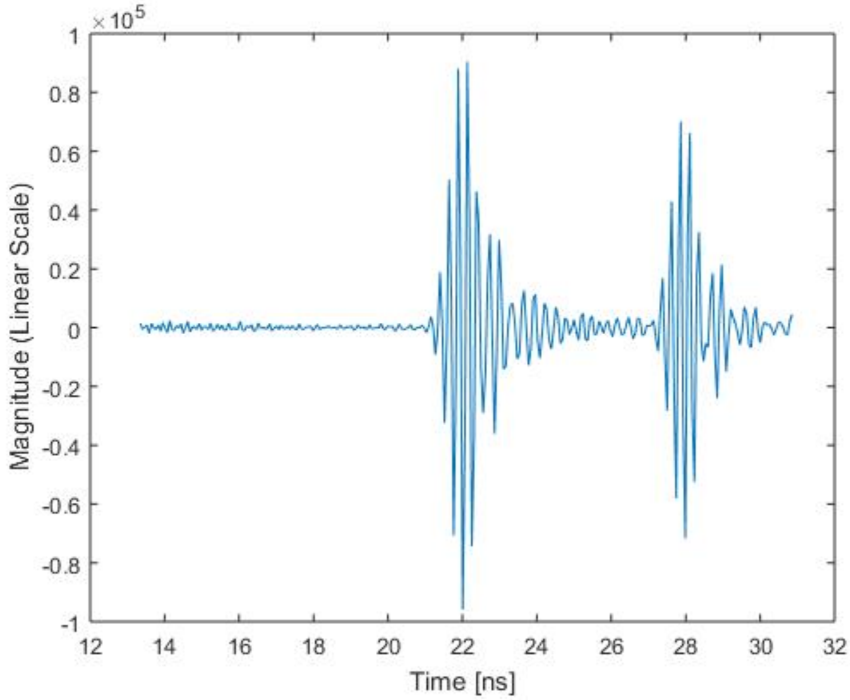

Figure 3.8 Range Imaging Signal s(t) from Target Scene with Targets Located at Ranges of 3.31 meters and 4.21 meters

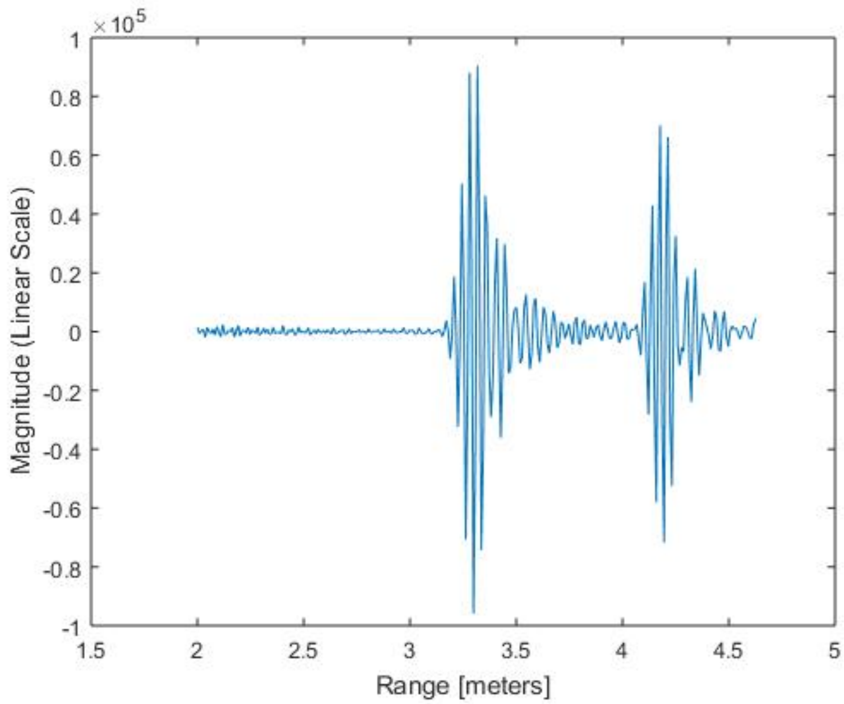

Figure 3.9 Range Imaging Signal s(t) in terms of Range from Target Scene with Targets Located at Ranges of 3.31 meters and 4.21 meters

The signal s(t) acquired by the PulsON 410 is a real valued bandpass signal. By taking the Fourier transform of the received signal s(t) in Figure 3.8, the spectrum of s(t) can be observed. Figure 3.10 shows the spectrum of the signal s(t) from Figure 3.8.

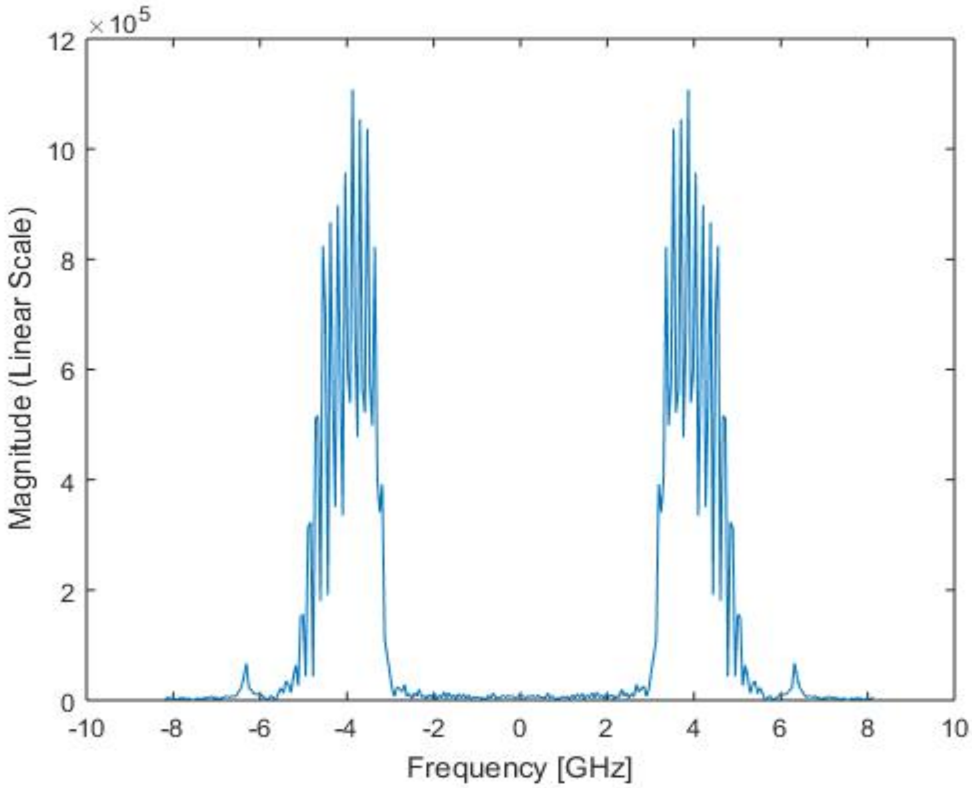

Figure 3.10 Spectrum of s(t) from Target Scene with Targets Located at 3.31 meters and 4.21 meters

Since the bandpass signal s(t) is a real valued signal, it contains no phase information. Changing the bandpass signal s(t) to an analytic signal is performed with the help of the Hilbert transform. The analytic version of s(t), denoted as $s_A(t)$ in this thesis, is found by (3) where ŝ(t) is the Hilbert transform of s(t) [17].

$$s_A(t) = s(t) + j\hat{s}(t) \tag{3}$$

The Matlab function hilbert is used to implement (3) in this thesis[1]. By taking the Fourier transform of $s_A(t)$, the spectrum of $s_A(t)$ can be observed. Figure 3.11 shows the spectrum of $s_A(t)$. By comparing Figure 3.10 and Figure 3.11, it is seen that the spectrum of $s_A(t)$ is similar to the spectrum of s(t) except that $s_A(t)$ does not have a negative frequency component.

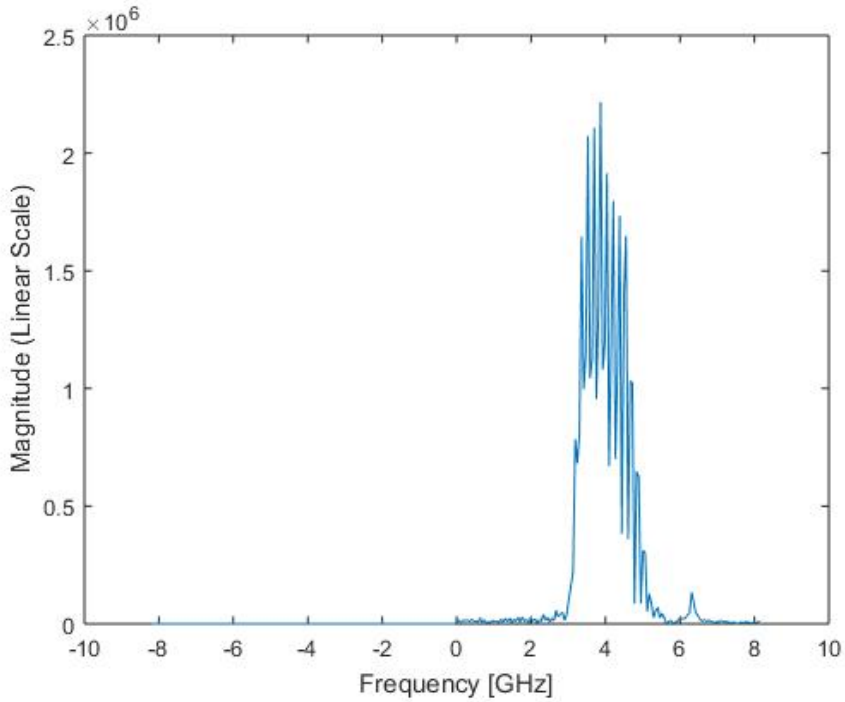

Figure 3.11 Spectrum of $s_A(t)$ from Target Scene with Targets Located at 3.31 meters and 4.21 meters

The baseband signal $s_b(t)$, also known as the lowpass equivalent (LPE) signal, can now be computed by (4) [17].

$$s_b(t) = s_A(t)e^{-j2\pi f_c t} \tag{4}$$

In (4), f_c = 4.3 GHz and is the center frequency of the P410 radar. Figure 3.12 shows

---

[1] The Matlab function hilbert directly transforms s(t) to $s_A(t)$ as opposed to $\hat{s}(t)$ [17].

the magnitude of the LPE signal $s_b(t)$. Recall that the bandpass signal of Figure 3.8 was redrawn as Figure 3.9 having the same shape as Figure 3.8, but the axis was relabeled in terms of range instead of time. In the same way, the magnitude of the LPE signal $s_b(t)$ of Figure 3.12 can be redrawn in Figure 3.13 with the axis in terms of range. The range of the targets can roughly be observed in Figure 3.13 by observing the locations of large amplitude. By taking the Fourier transform of $s_b(t)$, the spectrum of $s_b(t)$ can be observed. Figure 3.14 shows the spectrum of $s_b(t)$.

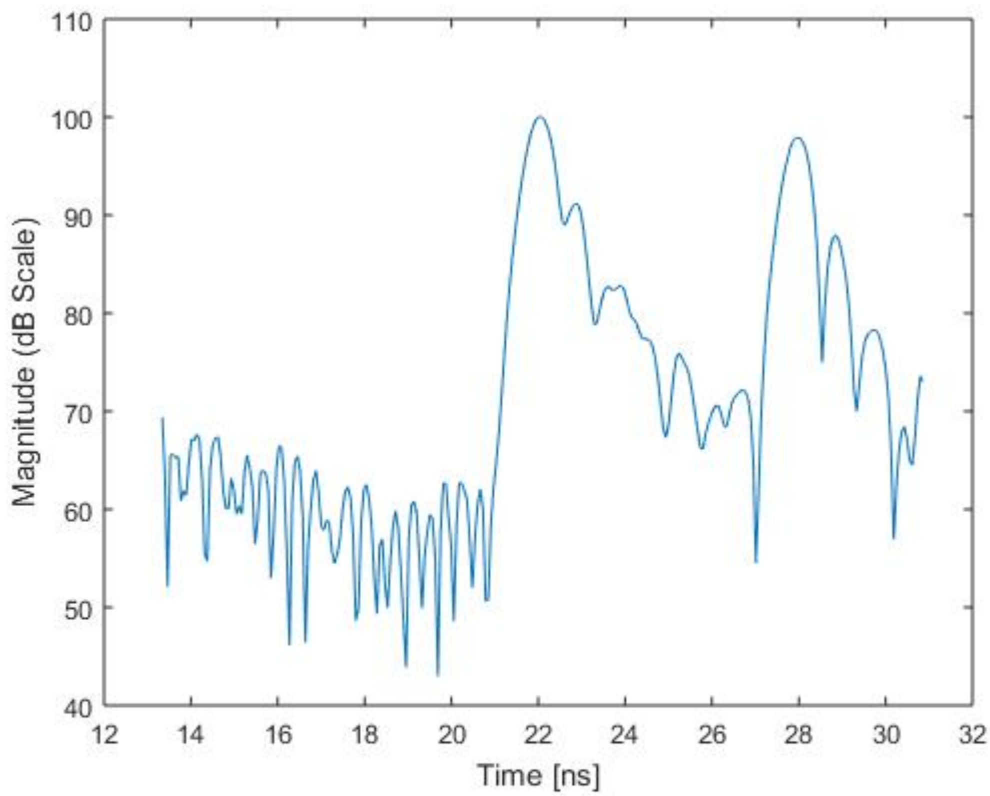

Figure 3.12 Range Imaging Signal $|s_b(t)|$ from Target Scene with Targets Located at Ranges of 3.31 meters and 4.21 meters

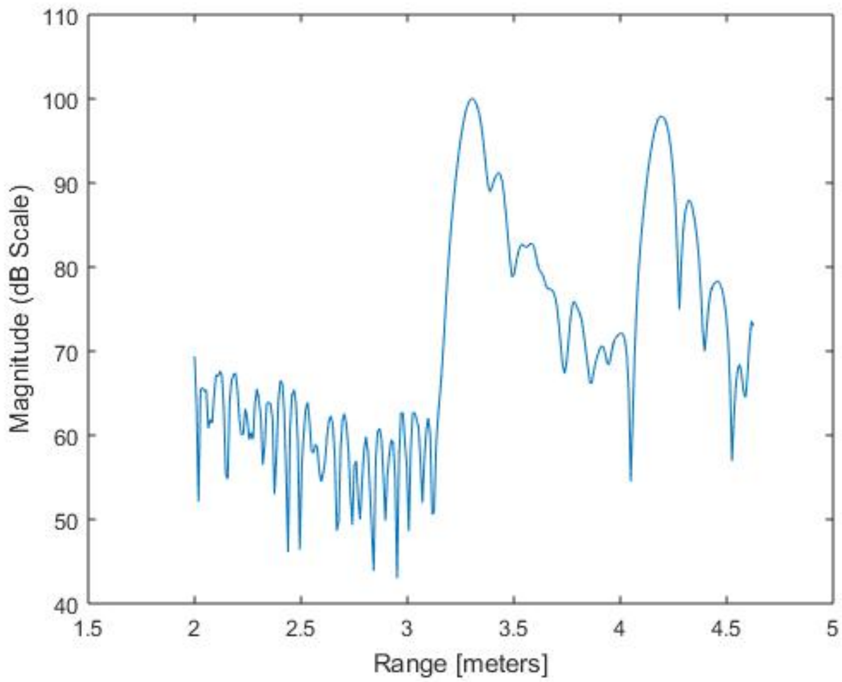

Figure 3.13 Range Imaging Signal $|s_b(t)|$ in Terms of Range from Target Scene with Targets Located at Ranges of 3.31 meters and 4.21 meters

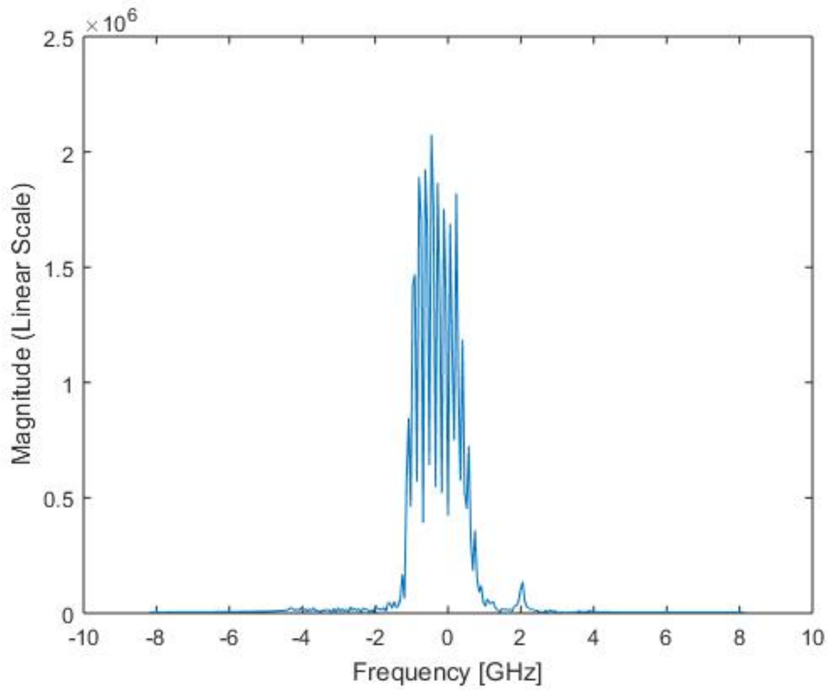

Figure 3.14 Spectrum of $s_b(t)$ from Target Scene with Targets Located at 3.31 meters and 4.21 meters

Previously, Figure 3.8 and Figure 3.9 showed the bandpass return signal s(t) for a target scene with targets at ranges of 3.31 meters and 4.21 meters. Figure 3.12 and Figure 3.13 showed the LPE return signal $s_b(t)$ for the same target scene. In Figure 3.12 and Figure 3.13, the high amplitude representing the range of the targets is much larger than the noise level presented in the signal. This will not always be the case. The range of the targets would become more difficult to distinguish for target echoes of lower magnitude. For a target of sufficiently low reflectivity or at a sufficiently large range, the echoes from the targets could have an amplitude that is nearly the same as the noise. One way to improve this situation is to increase the signal-to-noise ratio (SNR) of $s_b(t)$. This can be achieved with a technique known as matched filtering. Since the transmitted signal p(t) is known for the PulsON 410, as shown in Figure 2.2 in Chapter Two, a lowpass equivalent transmit signal $p_b(t)$ can be found for p(t) in the same way as the lowpass equivalent signal $s_b(t)$ was found for s(t). It is expected that the return echoes in s(t) will have the same shape as the transmitted waveform p(t) but delayed in time according to (2) and may have a different amplitude. The same reasoning applies to the lowpass equivalent signals $p_b(t)$ and $s_b(t)$. Matched filtering looks for a correlation between p(t) and the return signal s(t) or equivalently between $p_b(t)$ and $s_b(t)$. For the lowpass equivalent signals, the matched filter would be $p_b^*(-t)$, which is the time inverted, complex conjugate of $p_b(t)$. The output of the matched filter would be given by (5).

$$s_M(t) = s_b(t) \otimes p_b^*(-t) \tag{5}$$

The magnitude of $s_M(t)$ can then be plotted along the time axis and can also, from (2), be plotted along a range axis. By observing the location of large amplitude, the ranges of the targets can be observed.

In this thesis, an alternative form of matched filtering, known as reference signal matched filtering, is implemented. In reference signal matched filtering, a bandpass reference signal $s_0(t)$ is acquired from a reference target at a known range. With (3) and (4), a lowpass equivalent reference signal, $s_{0b}(t)$, can be created. This lowpass equivalent reference signal $s_{0b}(t)$ is used instead of the transmitted signal $p_b(t)$ to create the filter. In this case, the filter is $s_{0b}^*(-t)$, which is the time inverted, complex conjugate of $s_{0b}(t)$. The output of the reference signal matched filter is given by (6).

$$s_M(t) = s_b(t) \otimes s_{0b}^*(-t) \tag{6}$$

An additional step is now required that was not required with the matched filtering of (5). In matched filtering, the transmitted pulse occurs at t = 0. In the case of reference signal matched filtering, the reference signal $s_{0b}(t)$ is due to a target at range corresponding to a time not equal to 0. Due to this, the signal $s_M(t)$ from (6) needs to have its time axis shifted to compensate for the known range of the reference target. Then the magnitude of $s_M(t)$ can be plotted along the shifted time axis and also along a corresponding range axis according to (2). By observing the locations of large amplitude, the ranges of the targets can be observed. The Matlab function xcorr is used to implement the reference signal matched filter in this thesis.

To obtain a bandpass reference signal $s_0(t)$, a target is placed at a range of 3.31 meters as shown in Figure 3.15. The reference bandpass signal $s_0(t)$ is shown in Figure 3.16. Figure 3.17 is the same waveform as shown in Figure 3.16 except the axis is in terms of range according to (2). The location of the reference target can roughly be observed in Figure 3.16 and Figure 3.17 by the location of large amplitude. By taking the Fourier transform of the reference bandpass signal $s_0(t)$ in Figure 3.16, the spectrum of $s_0(t)$ can be observed. Figure 3.18 shows the spectrum of the signal $s_0(t)$.

Figure 3.15 Range Imaging Reference Target Located at a Ranges of 3.31 meters

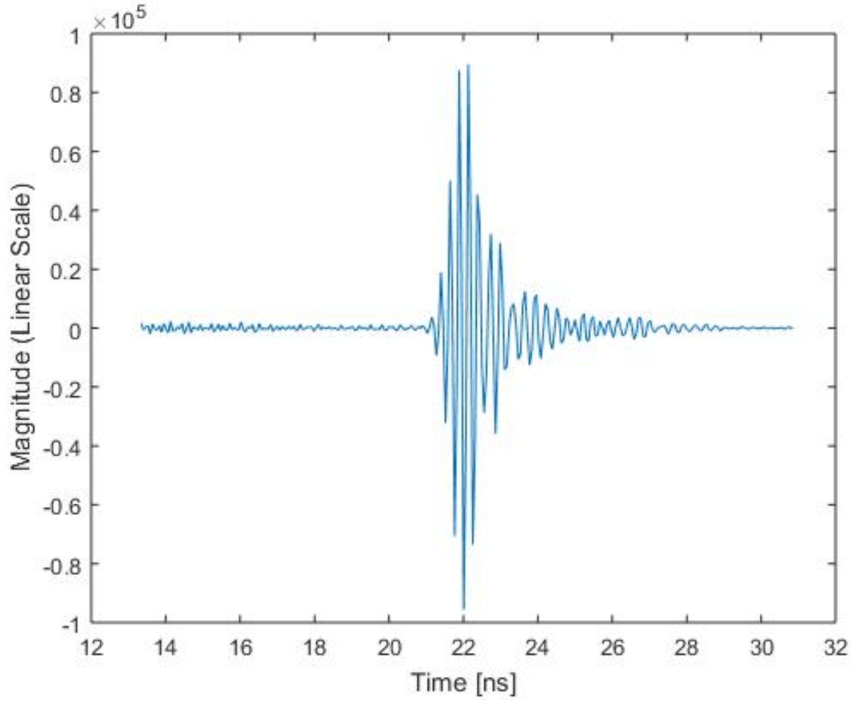

Figure 3.16 Range Imaging Signal $s_0(t)$ from Reference Target Located at a Range of 3.31 meters

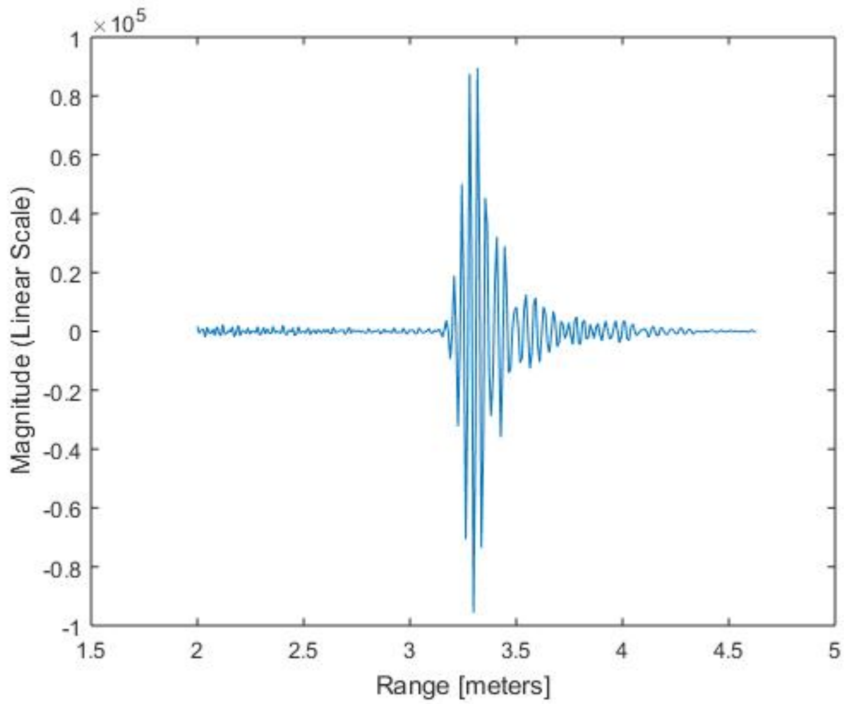

Figure 3.17 Range Imaging Signal $s_0(t)$ in terms of Range from Reference Target Located at a Range of 3.31 meters

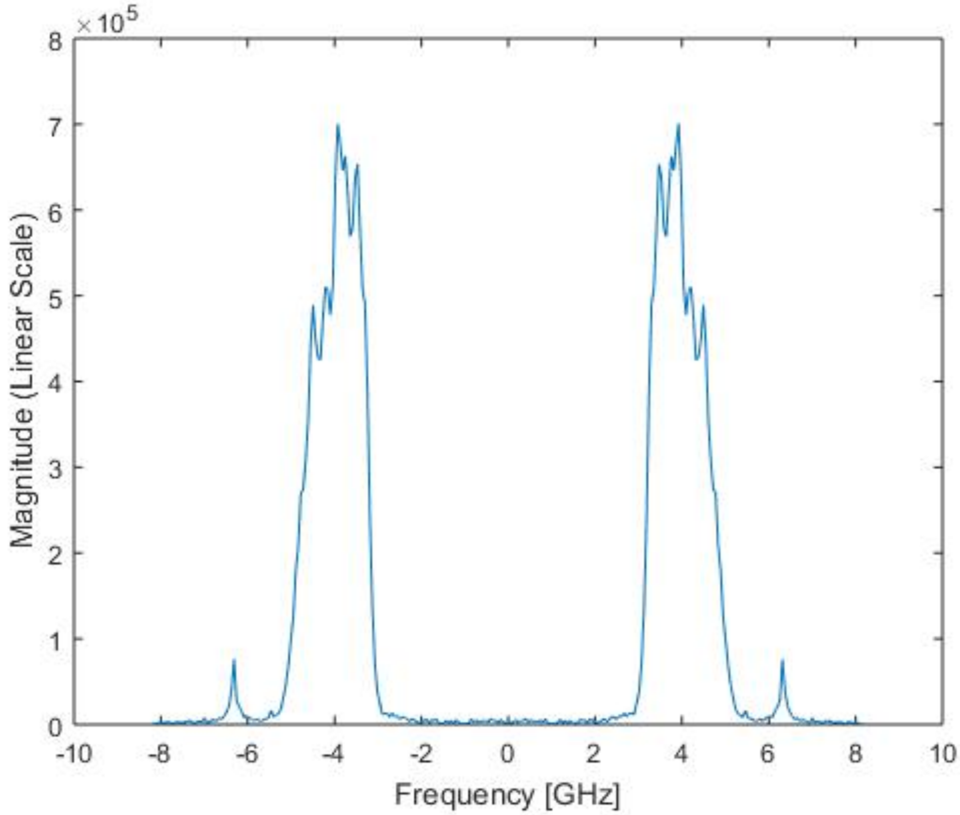

Figure 3.18 Spectrum of $s_0(t)$ from Reference Target Located at a Range of 3.31 meters

In a similar manner as described previously for the target scene with two targets, the analytic reference signal, $s_{0A}(t)$, can be obtained. By taking the Fourier transform of $s_{0A}(t)$, the spectrum can be observed. Figure 3.19 shows the spectrum of $s_{0A}(t)$.

In a similar manner as described previously for the target scene with two targets, the lowpass equivalent reference signal $s_{0b}(t)$ can be obtained. Its magnitude is shown in Figure 3.20. Figure 3.21 is the same waveform as shown in Figure 3.20 except the axis is in terms of range according to (2). The range of the reference target can be observed in Figure 3.21 by observing the location of large amplitude. By taking the Fourier transform of $s_{0b}(t)$, the spectrum of $s_{0b}(t)$ can be observed. Figure 3.22 shows the spectrum of $s_{0b}(t)$.

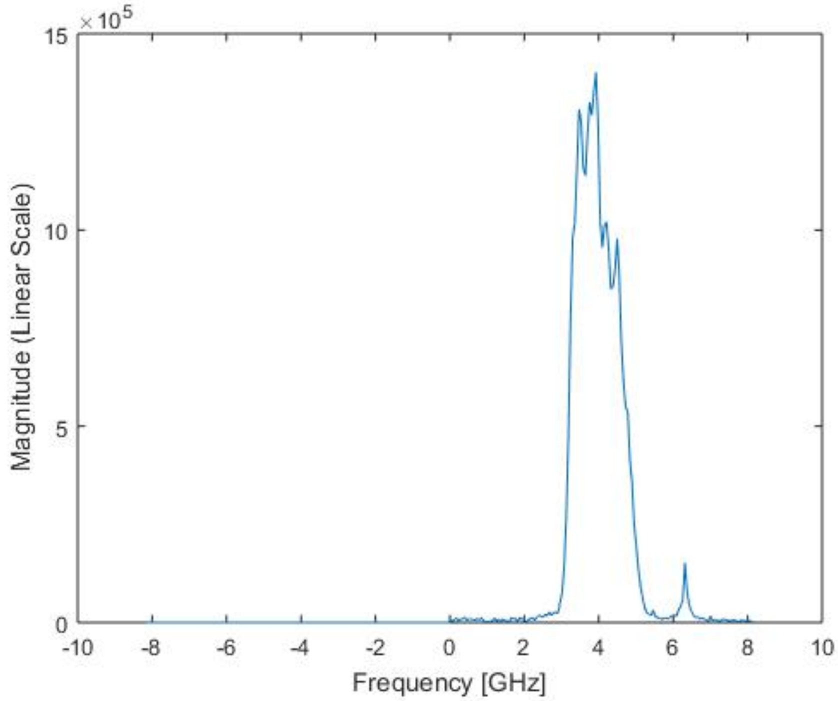

Figure 3.19 Spectrum of $s_{0A}(t)$ from Reference Target Located at a Range of 3.31 meters

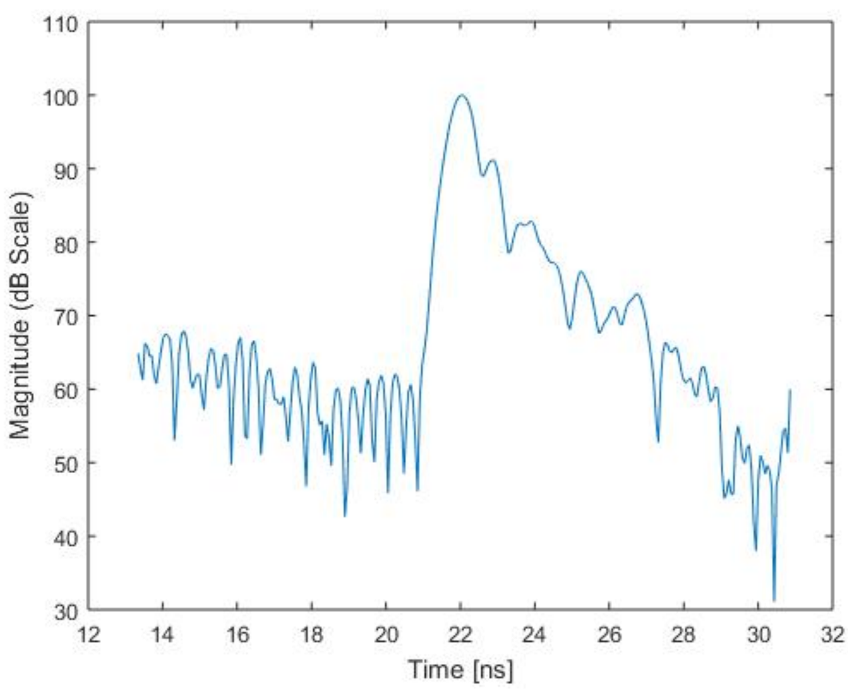

Figure 3.20 Range Imaging Signal $|s_{0b}(t)|$ from Reference Target Located at a Range of 3.31 meters

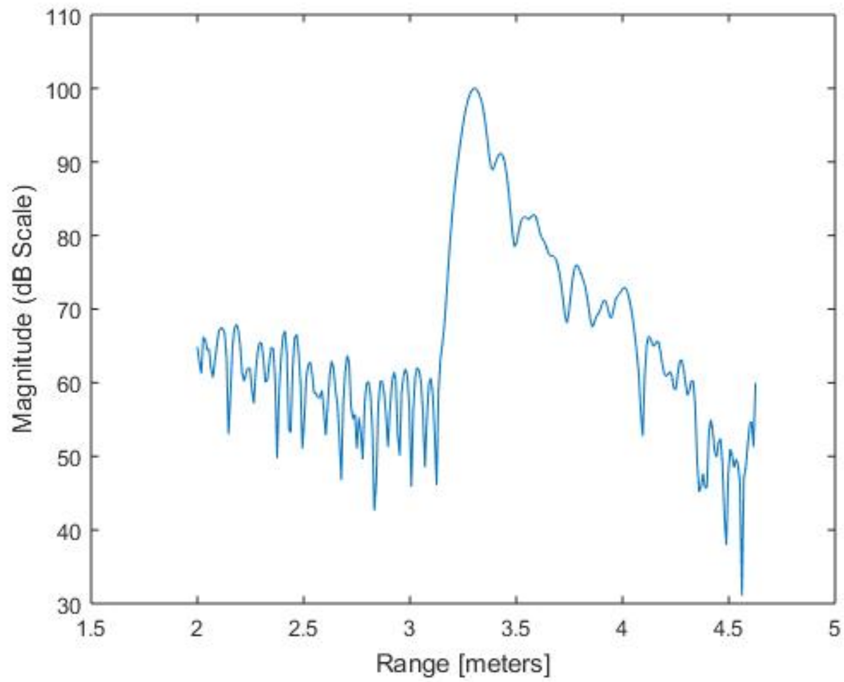

Figure 3.21 Range Imaging Signal |s0b(t)| in Terms of Range from a Reference Target Located at a Range of 3.31 meters

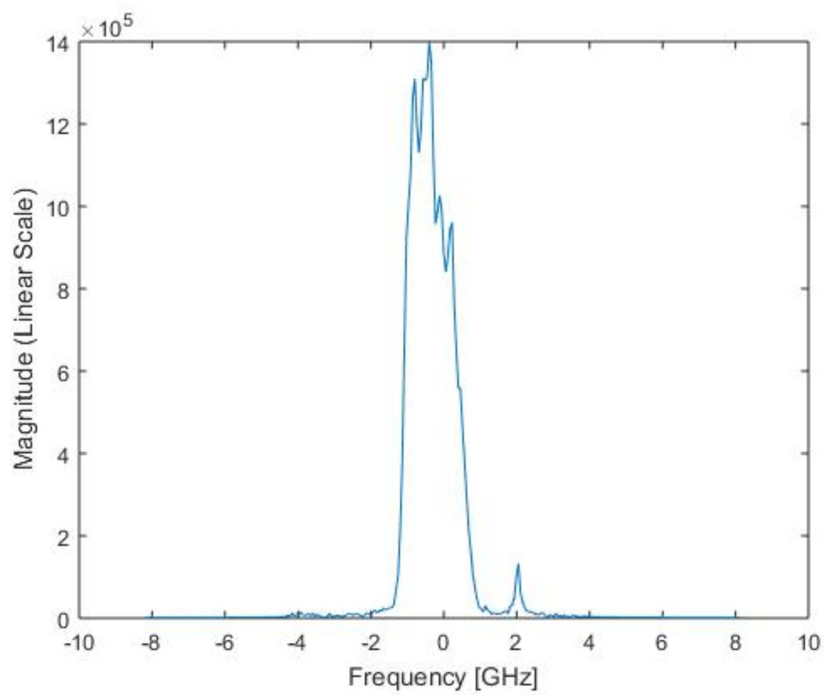

Figure 3.22 Spectrum of $s_{0b}(t)$ from Reference Target Located at a Range of 3.31 meters

The lowpass equivalent reference signal return, $s_{0b}(t)$ from Figure 3.20 is now used as a filter to implement reference signal matched filtering. After axis realignment, the magnitude of the output of the filter is shown in Figure 3.23. Figure 3.24 is the same signal as Figure 3.23 except it is in terms of range according to (2). This is the final result of the range imaging algorithm. The range of the objects can be found by observing the locations of large amplitude.

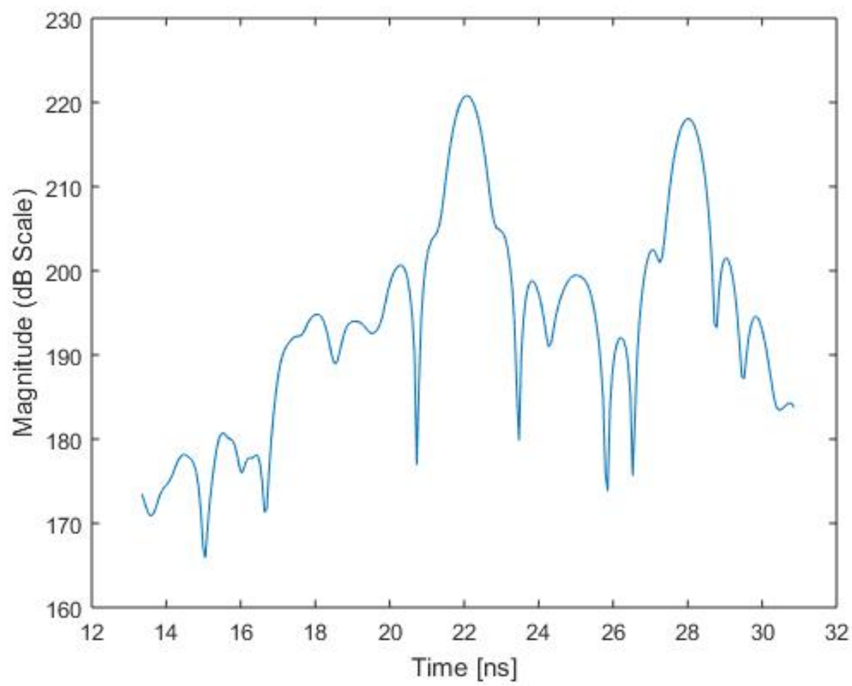

Figure 3.23 Range Imaging Signal $|s_M(t)|$ after axis realignment, for Target Scene with Targets Located at Ranges of 3.31 Meters and 4.21 Meters

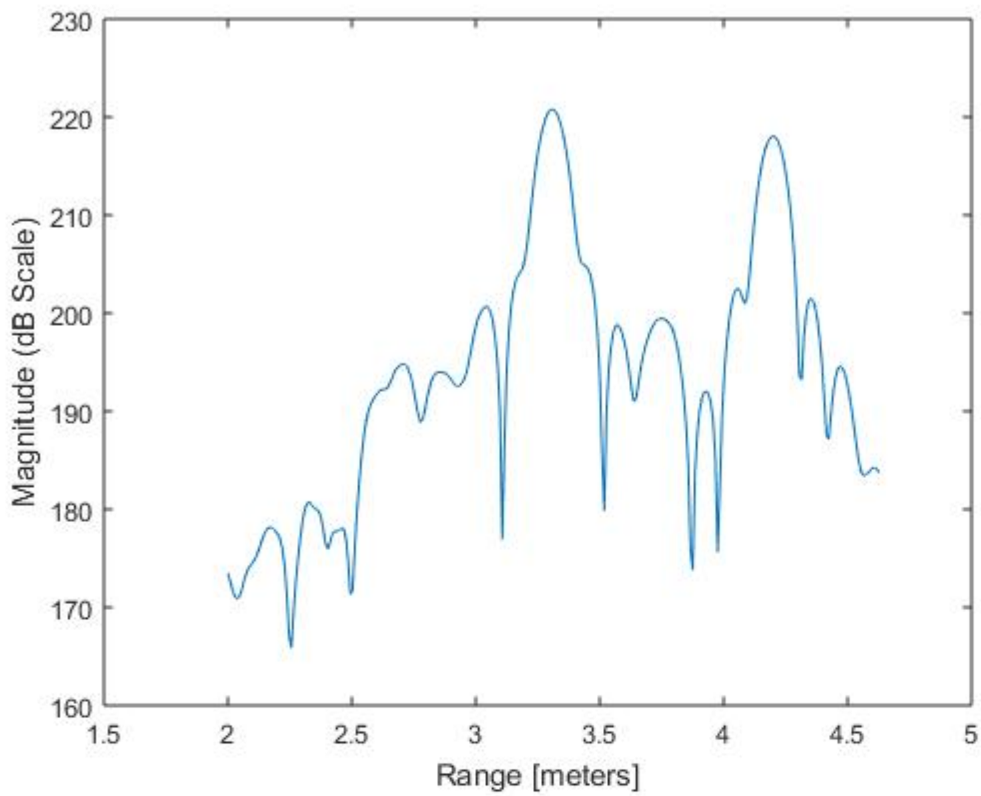

Figure 3.24 Range Imaging Signal $|s_M(t)|$ after axis realignment, in terms of Range for Target Scene with Targets Located at Ranges of 3.31 meters and 4.21 meters

3.3 Additional Range Imaging Results

Section 3.2 described the range imaging algorithm used in this thesis. Results from a target scene with targets at a range of 3.31 meters and 4.21 meters were used in the discussion. Section 3.3 shows the results for another target scene.

This target scene consists of two targets located at a range of 6.63 meters and 7.51 meters. The target scene is shown in Figure 3.25. The target bandpass signal, s(t), is shown in Figure 3.26. Figure 3.27 is the same waveform as shown in Figure 3.26 except the axis is relabeled in terms of range according to (2). The spectrum of the target bandpass signal, s(t), and the spectrum of the target analytic signal, $s_A(t)$, are shown in Figure 3.28 and Figure 3.29, respectively. The magnitude of the target lowpass equivalent signal, $s_b(t)$, is shown in Figure 3.30. Figure 3.31 is the same waveform as shown in Figure 3.30 except the axis is relabeled in terms of range according to (2). The spectrum of the target lowpass equivalent signal, $s_b(t)$, is shown in Figure 3.32.

The reference target, located at a range of 6.63 meters is shown in Figure 3.33. The reference bandpass signal, $s_0(t)$, is shown in Figure 3.34. Figure 3.35 is the same waveform as shown in Figure 3.34 except the axis is relabeled in terms of range according to (2). The spectrum of the reference bandpass signal, $s_0(t)$, and the spectrum of the reference analytic signal, $s_{0A}(t)$, are shown in Figure 3.36 and Figure 3.37, respectively. The magnitude of the reference lowpass equivalent signal, $s_{0b}(t)$, is shown in Figure 3.38. Figure 3.39 is the same waveform as shown in Figure 3.38 except the axis is relabeled in terms of range according to (2). The spectrum of the reference lowpass equivalent signal, $s_{0b}(t)$, is shown in Figure 3.40.

The magnitude of the output of the reference signal matched filter after axis realignment is shown in Figure 3.41. Figure 3.42 is the same waveform as shown in Figure 3.41 except the axis is relabeled in terms of range according to (2).

Figure 3.25 Range Imaging Target Scene with Targets Located at Ranges of 6.63 meters and 7.51 meters

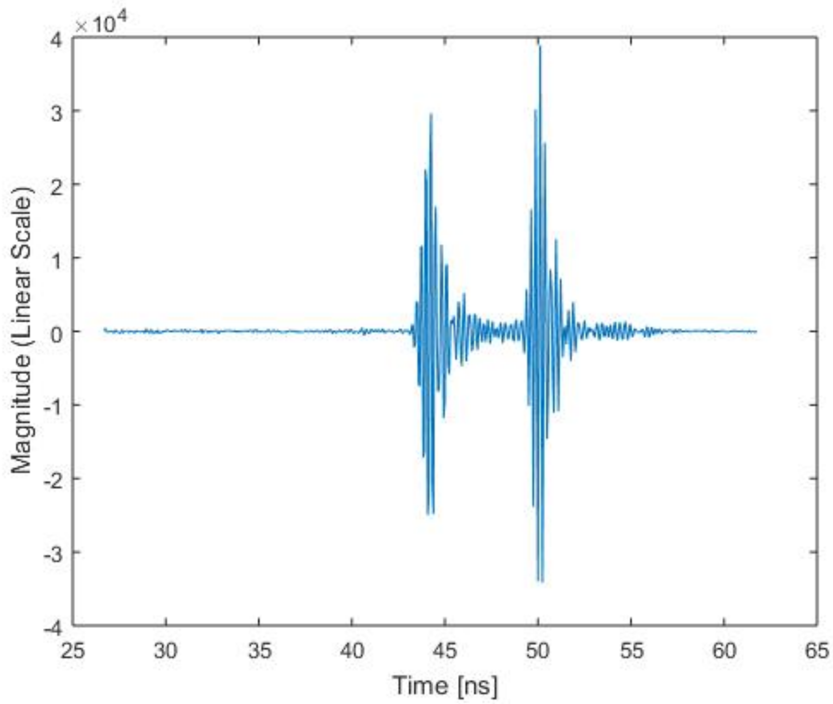

Figure 3.26 Range Imaging Signal s(t) from Target Scene with Targets Located at Ranges of 6.63 meters and 7.51 meters

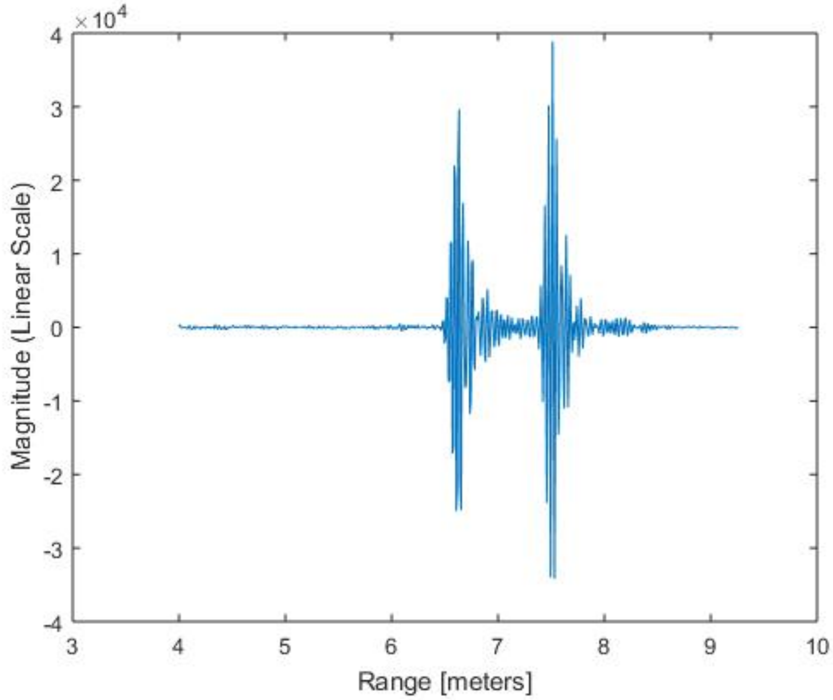

Figure 3.27 Range Imaging Signal s(t) in terms of Range from Target Scene with Targets Located at Ranges of 6.63 meters and 7.51 meters

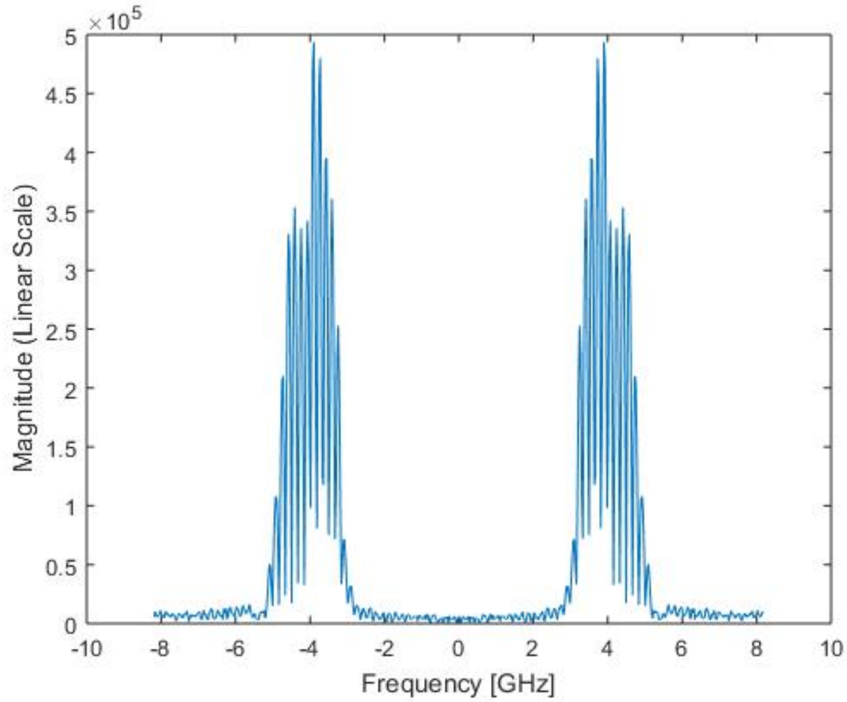

Figure 3.28 Spectrum of s(t) from Target Scene with Targets Located at 6.63 meters and 7.51 meters

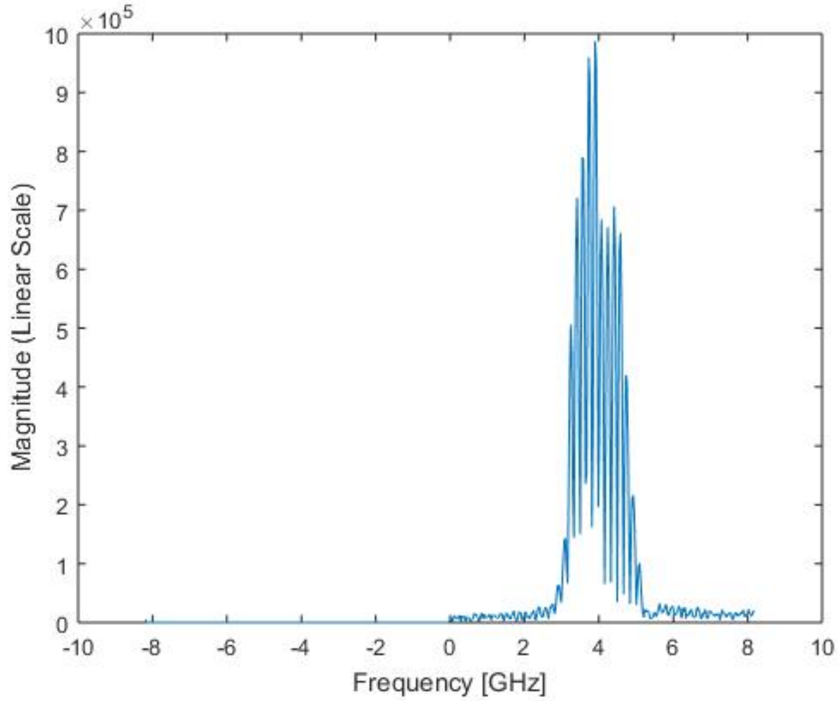

Figure 3.29 Spectrum of $s_A(t)$ from Target Scene with Targets Located at 6.63 meters and 7.51 meters

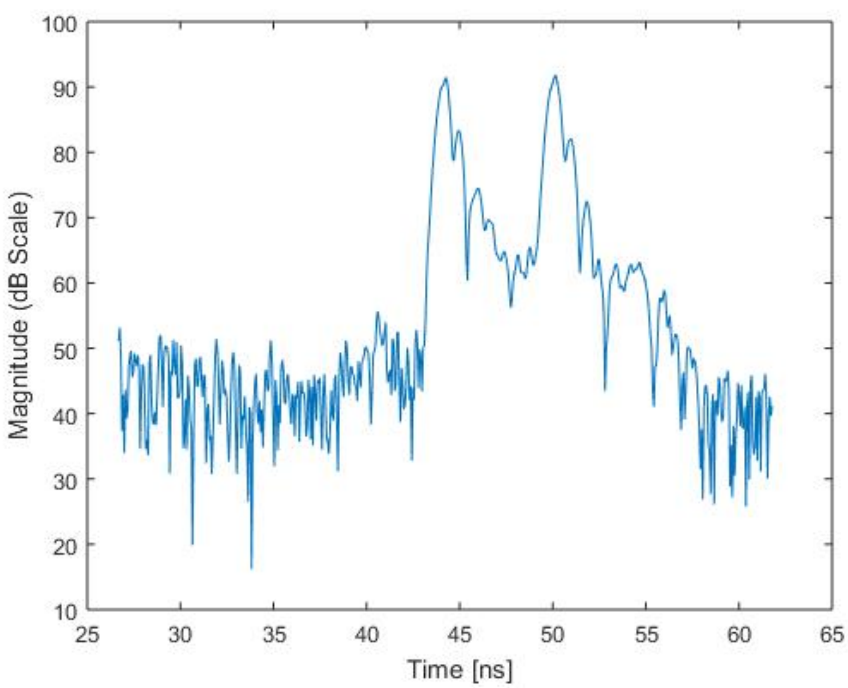

Figure 3.30 Range Imaging Signal $|s_b(t)|$ from Target Scene with Targets Located at Ranges of 6.63 meters and 7.51 meters

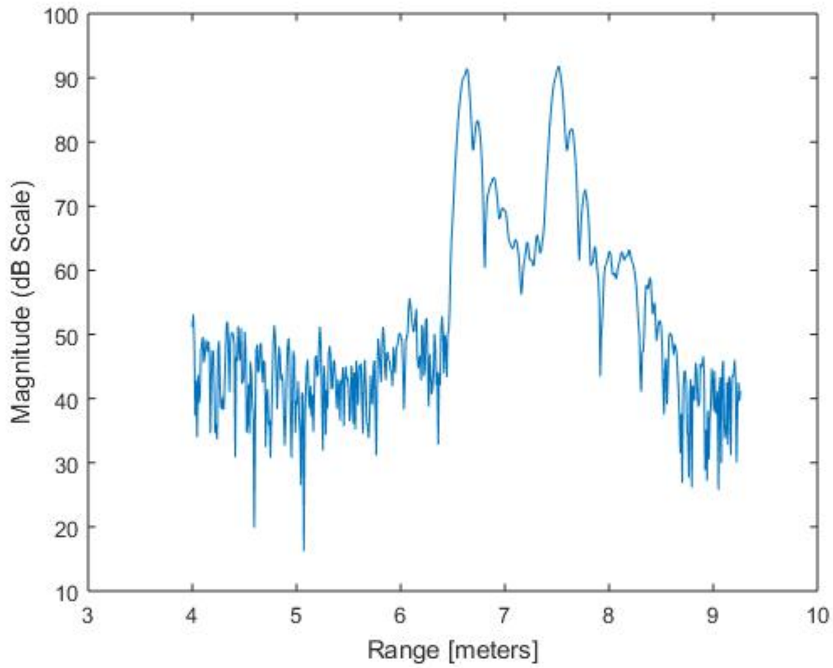

Figure 3.31 Range Imaging Signal $|s_b(t)|$ in Terms of Range from Target Scene with Targets Located at Ranges of 6.63 meters and 7.51 meters

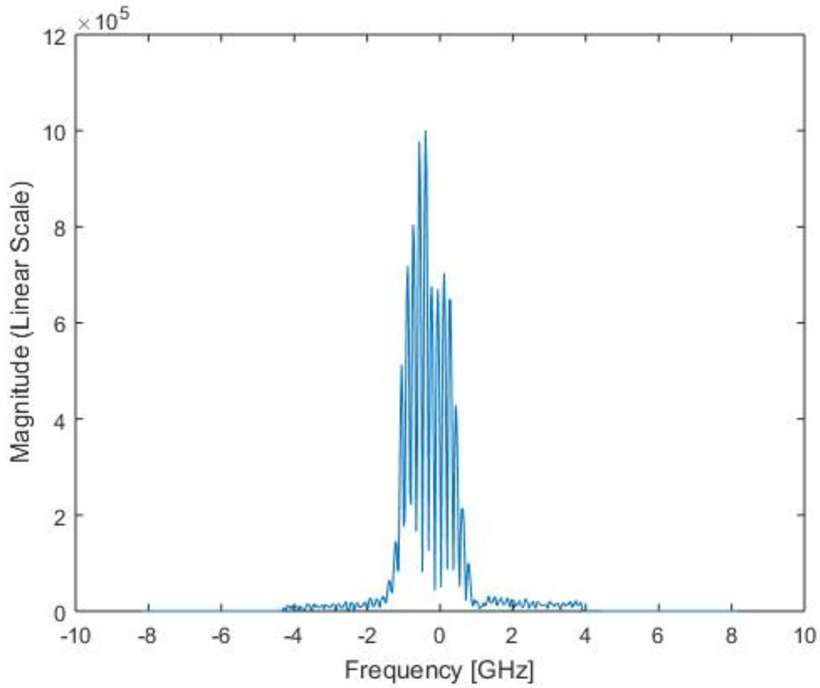

Figure 3.32 Spectrum of $s_b(t)$ from Target Scene with Targets Located at 6.63 meters and 7.51 meters

Figure 3.33 Range Imaging Reference Target Located at a Range of 6.63 meters

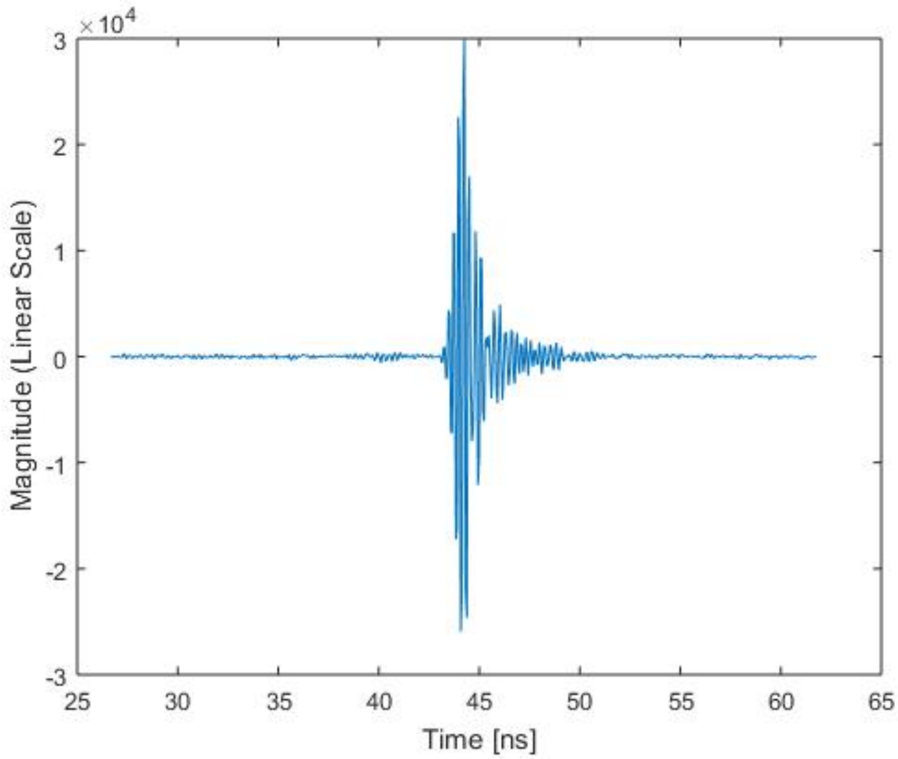

Figure 3.34 Range Imaging Signal $s_0(t)$ from Reference Target Located at a Range of 6.63 meters

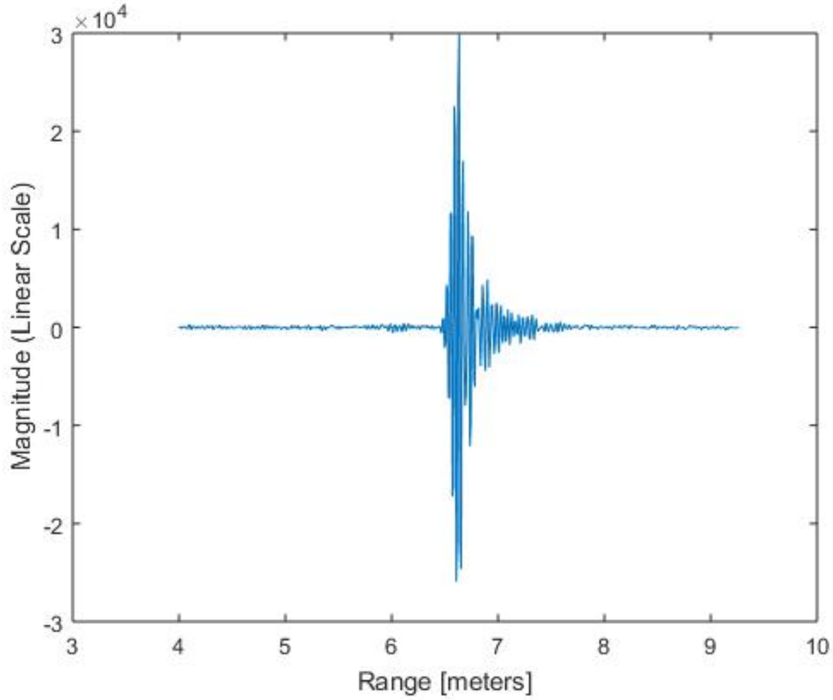

Figure 3.35 Range Imaging Signal $s_0(t)$ in terms of Range from Reference Target Located at a Range of 6.63 meters

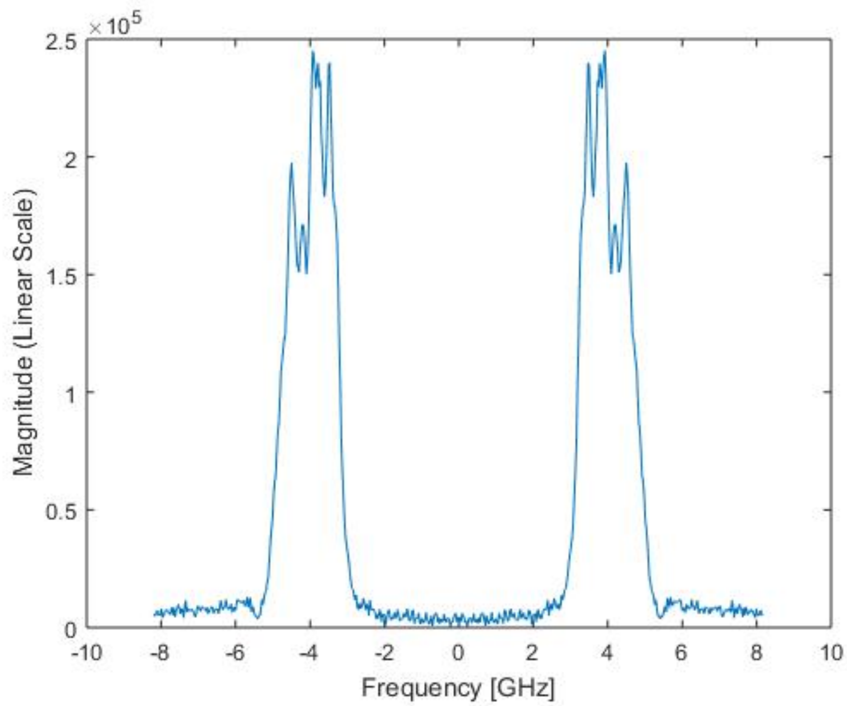

Figure 3.36 Spectrum of $s_0(t)$ from Reference Target Located at a Range of 6.63 meters

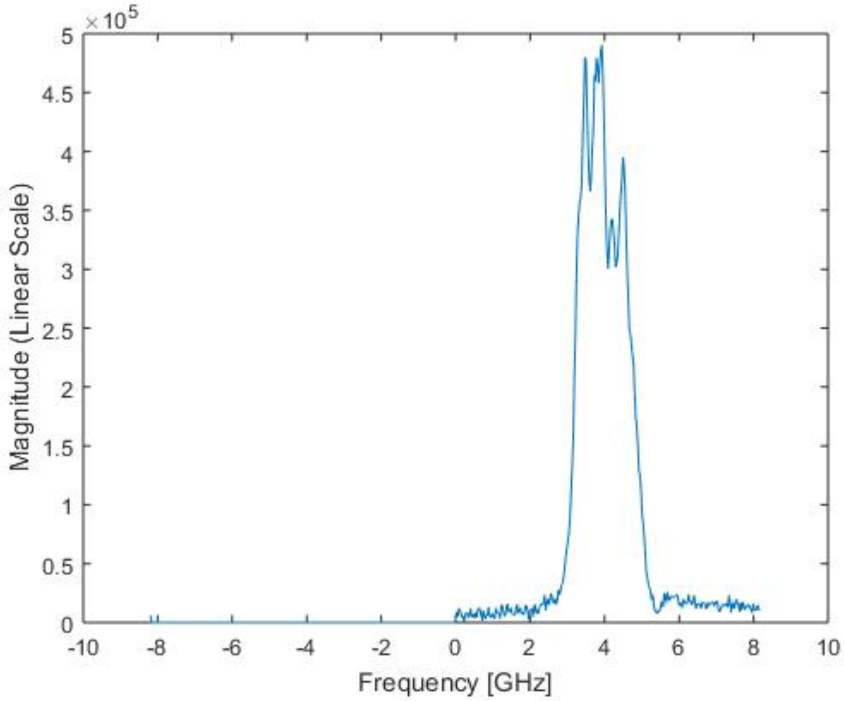

Figure 3.37 Spectrum of $s_{0A}(t)$ from Reference Target Located at a Range of 6.63 meters

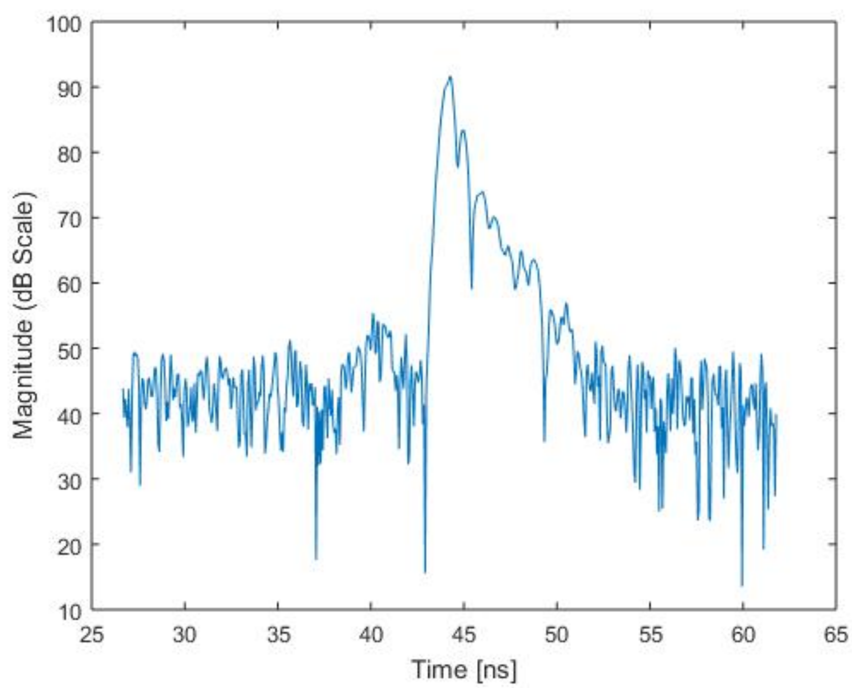

Figure 3.38 Range Imaging Signal $|s_{0b}(t)|$ from Reference Target Located at a Range of 6.63 meters

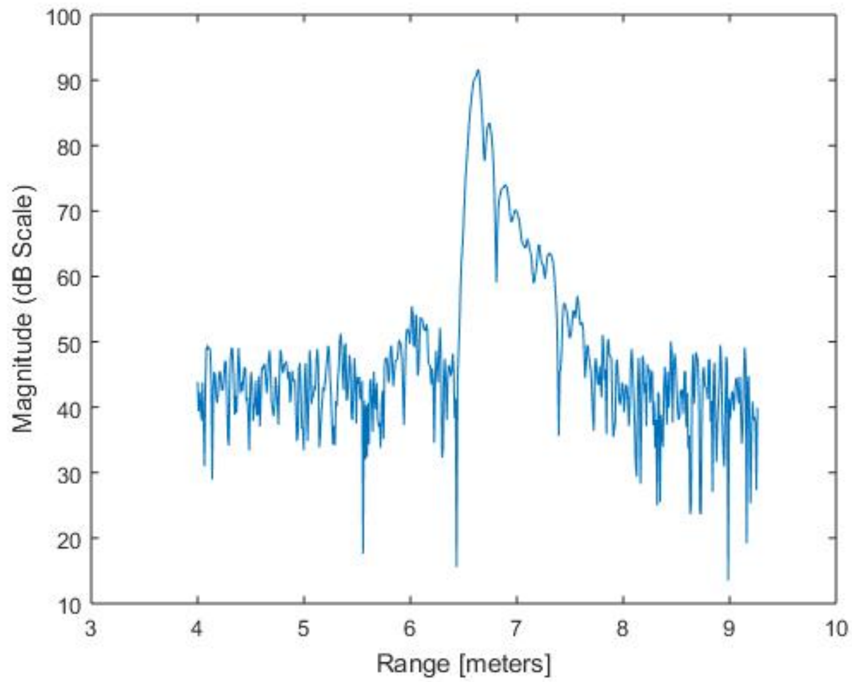

Figure 3.39 Range Imaging Signal $|s_{0b}(t)|$ in Terms of Range from a Reference Target Located at a Range of 6.63 meters

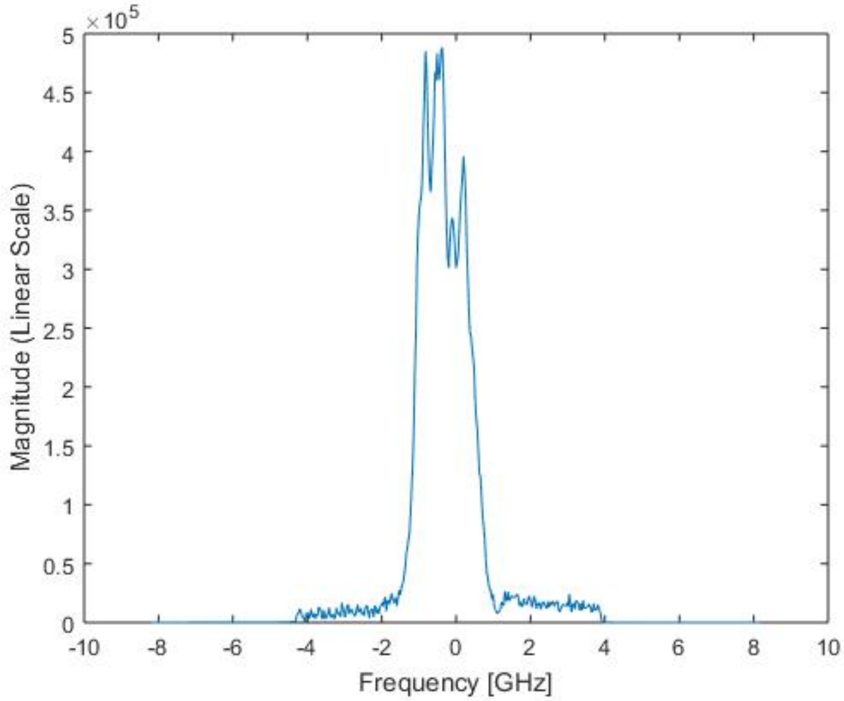

Figure 3.40 Spectrum of $s_{0b}(t)$ from Reference Target Located at a Range of 6.63 meters

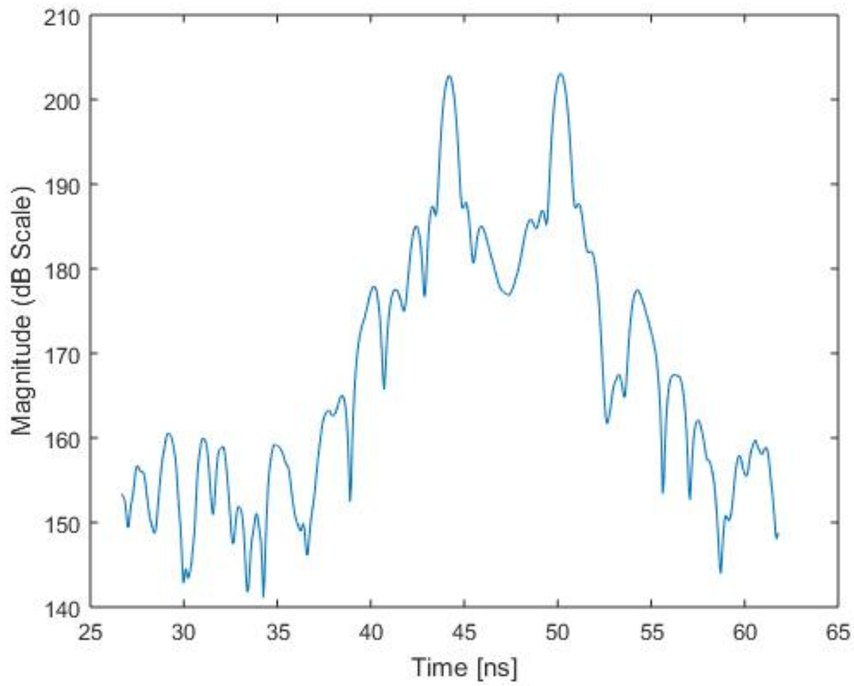

Figure 3.41 Range Imaging Signal $|s_M(t)|$ after Axis Realignment, for Target Scene with Targets Located at Ranges of 6.63 meters and 7.51 meters

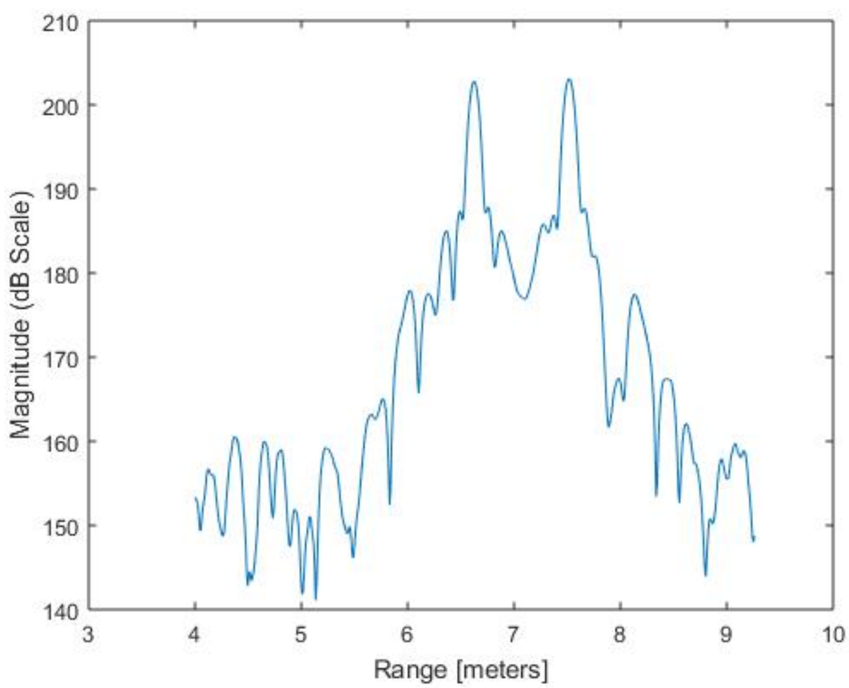

Figure 3.42 Range Imaging Signal $|s_M(t)|$ after Axis Realignment, in terms of Range, for Target Scene with Targets Located at Ranges of 6.63 meters and 7.51 meters

CHAPTER FOUR

Cross Range Imaging

Chapter Three presented an algorithm for determining the range of targets from the radar. The range imaging algorithm was able to determine the range, but not the direction of the targets from the radar. This chapter presents an algorithm for determining a different spatial property of the targets in a target scene. This is called cross range. This chapter describes a cross range imagng algorithm from Chapter 2 of [3] and presents the results from the Matlab implementation of the cross range imaging algorithm for several target scenes. Appendix C documents the Matlab program Cross_Range_Imaging.m developed for this thesis. Cross_Range_Imaging.m uses data acquired with Data_Collection.m, described in Chapter Two of this thesis.

4.1 Cross Range Imaging Basics

Consider Figure 4.1. In Figure 4.1, an axis is defined for the target scene. Two targets are located at the same known x coordinate and at unknown y coordinates. The goal of the cross range imaging algorithm presented in this chapter is to determine the y coordinates of targets placed at the same known x coordinates and at unknown y coordinates.

As described in Chapter Three, the radar transmits an electromagnetic pulse s(t) and the RX antenna acquires an electromagnetic signal s(t) containing echoes from the targets. With knowledge of the echo return time, the range of the targets can be determined.

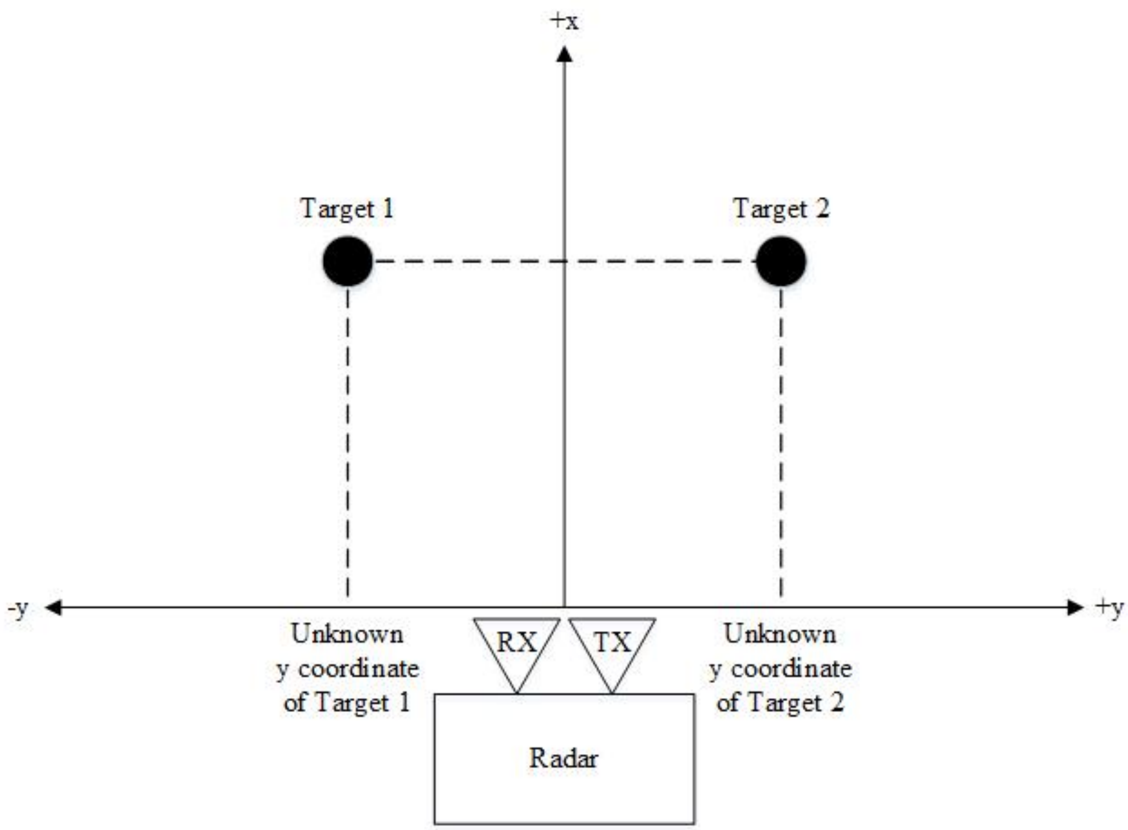

Figure 4.1 Cross Range Imaging Axis Definition

In cross range imaging, the location of the radar is allowed to vary. The radar moves along the y axis at x = 0. The location of the radar can be described be the coordinate (0,u), where u is the y coordinate of the radar. The radar starts at location (0,-L). At this location, the radar transmits p(t) and receives back s(t). The radar then moves a short distance to another location. At this new location, the radar transmits p(t) again and receives back another s(t). The received s(t) and this new location will be different than the s(t) received at the first location since the relative positions of the targets has changed in relationship to the radar. This process repeats until the last p(t) is transmitted and the last s(t) is acquired when the radar is at (0,+L). The return signals, s(t), from each location are bandpass signals; each signal can be changed into a baseband (lowpass equivalent) form, $s_b(t)$, in the same way as described in Chapter Three. The lowpass equivalent return signal $s_b(t)$ from each radar position can now can be combined to form a two dimensional signal $s_b(t,u)$. With the signal $s_b(t,u)$, the cross range imaging algorithm determines the cross range (y coordinates) of the targets at the same known x coordinate.

4.2 Cross Range Imaging Algorithm

Section 4.1 presented the basics of cross range imaging. This section describes the details of the cross range imaging algorithm by looking at results of the cross range imaging algorithm for the target scene shown in Figure 4.2. In Figure 4.2, two targets are placed in the target scene. One target is placed at (3.31,0) and the other is placed at (3.31,-1). The units are in meters. The x coordinates are known values. The goal of the cross range imaging algorithm is to determine the y coordinates of the two targets.

Figure 4.2 Cross Range Imaging Target Scene with Targets Located at (3.31,-1) and (3.31,0)

The radar starts at u = -1.50 meters. At this location, the radar transmits a signal p(t) and receives a return signal s(t) in the same way described in Chapter Three for range imaging. The return signal is then converted into a baseband (lowpass equivalent) signal, $s_b(t)$ as described in Chapter Three. Now, instead of just referring to the baseband return signal as $s_b(t)$, it will be referred to as $s_b(t,-1.50)$ The radar then moves 1 cm to the right to u = -1.49 meters. Again, at this location, the radar transmits a signal p(t) and receives a return signal which will be called s(t,-1.49). The signal s(t,-1.49) is converted to the signal $s_b(t,-1.49)$. The signal $s_b(t,-1.49)$ will be different than $s_b(t,-1.50)$ since the relative positions of the targets to the radar are different. This process continues with the radar moving in 1 cm increments to the right and at each location transmitting a signal p(t), receiving a signal s(t,u), and converting the received signal to a lowpass equivalent signal $s_b(t,u)$. The last transmitted signal, p(t) is sent at the radar position u = +1.50 meters. At this location, the radar receives the signal s(t,+1.50) and converts it to the lowpass equivalent signal $s_b(t,+1.50)$.

In Chapter Three, the signal $s_b(t)$ was processed in Matlab as a row vector of length N. In cross-range imaging, the two dimensional return signal, $s_b(t,u)$ is formed where the lowpass equivalent return signal acquired at each location forms a row of $s_b(t,u)$. In Matlab, $s_b(t,u)$ is processed as an M x N matrix, where M is the number of rows and N is the number of columns. In the current case being described, M = 301 since the radar moved from u = -1.50 meters to u = +1.5 meters in 1 cm increments. An image of the magnitude of $s_b(t,u)$ is shown in Figure 4.3.

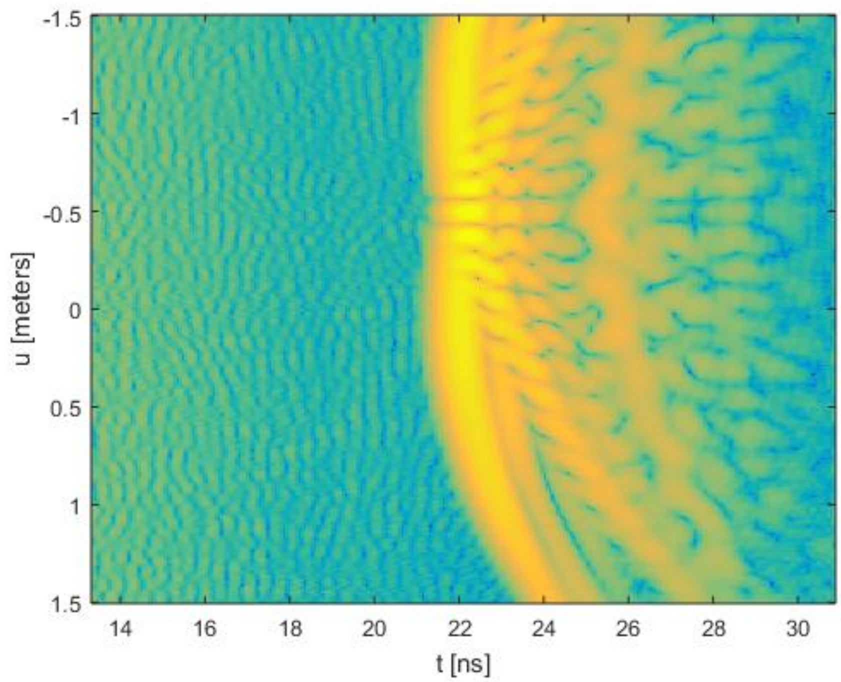

Figure 4.3 Cross Range Imaging Signal $|s_b(t,u)|$ from Target Scene with Targets Located at (3.31,-1) and (3.31,0)

Recall that the targets are both located at x = 3.31 meters. The range of a target from the radar will be greater than the target's x coordinate value except when the radar is at the same y coordinate as the target, in which case, the target's range from the radar and the target's x coordinate value will be the same[2]. From (2), a range of 3.31 meters corresponds to a time of 22.07 ns. In Figure 4.3, two arcs can be observed, each corresponding to one of the targets. Each arc has a minimum time at approximately 22.07 ns corresponding to an x coordinate value of 3.31 meters.

---

[2] This is not true for targets located at a y coordinate $y > +L$ or $y < -L$. In this case, the range of the target from the radar will always be greater than the target's x coordinate value. This case is not considered for the work in this thesis.

For $s_b(t,u)$, by allowing t = 22.07 ns, which corresponds to a range of 3.31 meters, the one dimensional signal $s_b(t=22.07,u)$ is formed. This signal is a column vector in Matlab. The signal $s_b(t=22.07,u)$ is the result of both targets. The real part of $s_b(t=22.07,u)$ is shown in Figure 4.4.

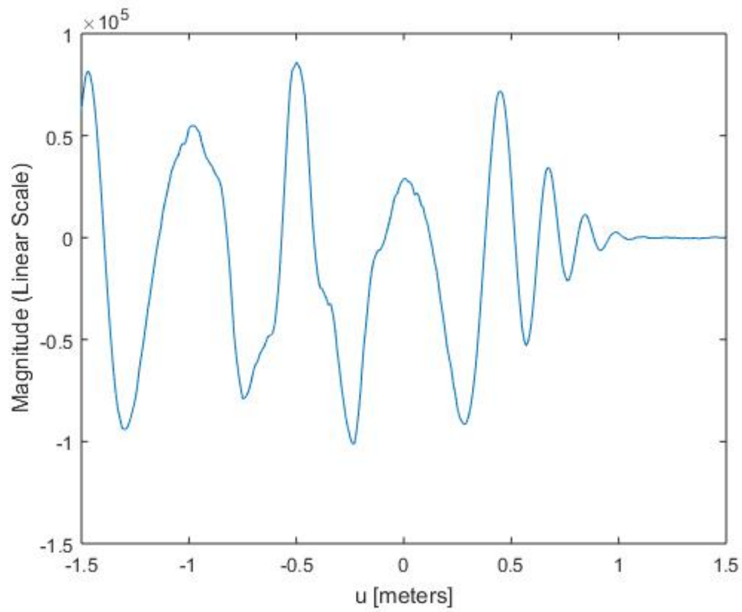

Figure 4.4 Cross Range Imaging Signal $Re[s_b(22.07,u)]$ from Target Scene with Targets Located at (3.31,-1) and (3.31,0)

In Chapter Three, a signal s(t) was acquired for a target scene and converted to the lowpass equivalent $s_b(t)$. Reference signal matched filtering was then performed based upon a reference signal $s_{0b}(t)$ from a reference target at a known range. In a similar way, the cross range imaging algorithm presented in this chapter uses reference signal matched filtering. A reference target located at x = 3.31 meters and at a known y coordinate, y = 0 meters, is shown in Figure 4.5.

Figure 4.5 Cross Range Imaging Reference Target Located at (3.31,0)

Using the same process described previously for the target scene to obtain the two dimensional signal $s_b(t,u)$, a two dimensional reference signal $s_{0b}(t,u)$ can be formed. An image of the magnitude of $s_{0b}(t,u)$ is shown in Figure 4.6. For $s_{0b}(t,u)$, by allowing t = 22.07 ns, the one dimensional signal $s_{0b}(t=22.07,u)$ is formed. This signal is processed as a column vector in Matlab. The real part of $s_{0b}(t=22.07,u)$ is shown in Figure 4.7.

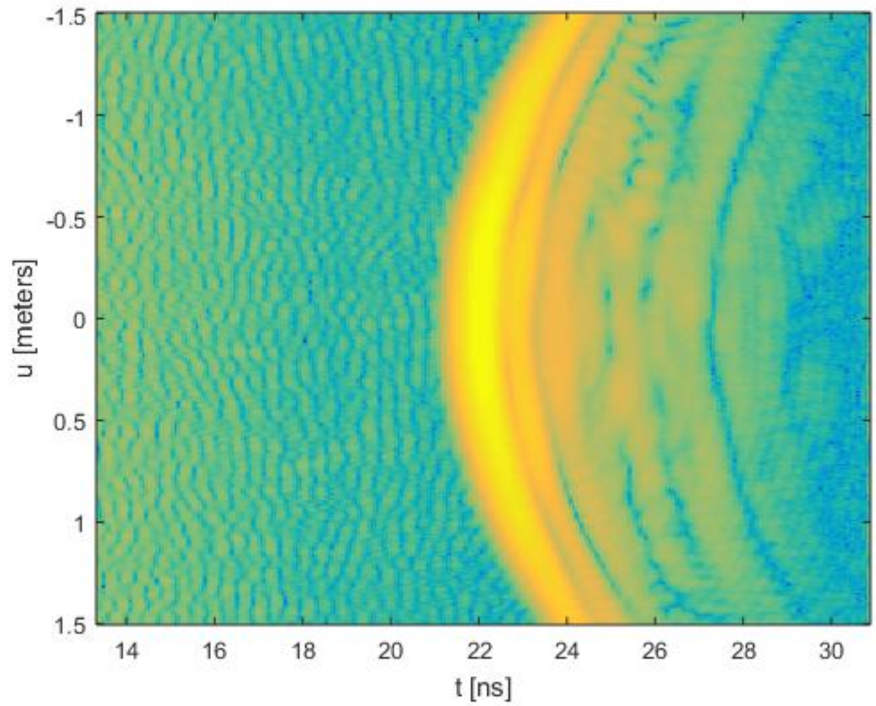

Figure 4.6 Cross Range Imaging Signal $|s_{0b}(t,u)|$ from Reference Target Located at (3.31,0)

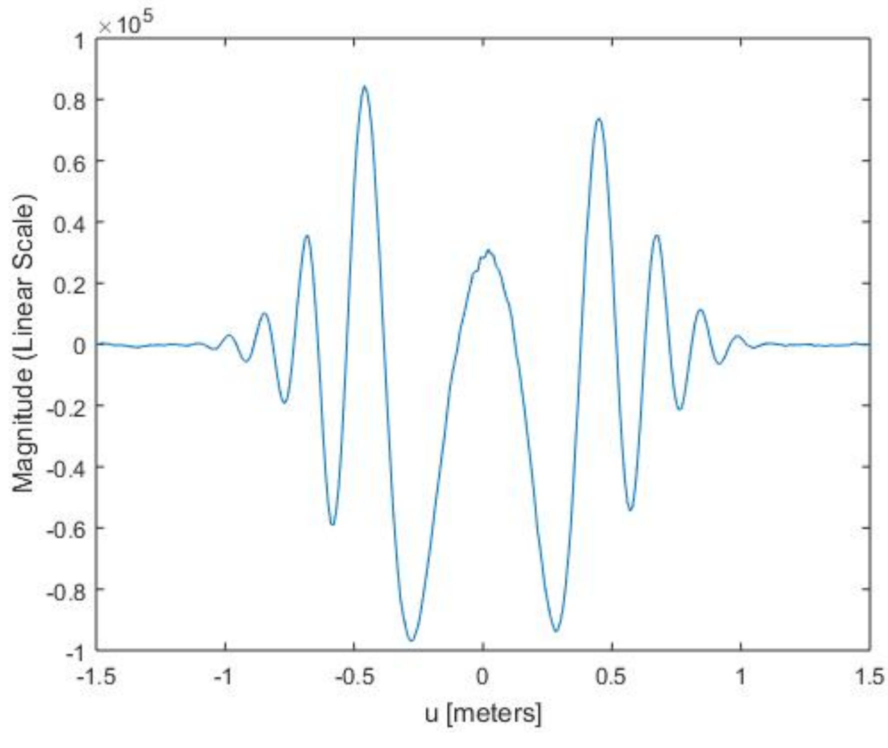

Figure 4.7 Cross Range Imaging Signal $Re[s_{0b}(22.07,u)]$ from Reference Target Located at (3.31,0)

For reference signal matched filtering for cross range imaging, the signal $s_{0b}(22.07,u)$ is used to create the filter. The filter is $s_{0b}^*(22.07,-u)$ which is the u inverted, complex conjugate of $s_{0b}(22.07,u)$. The output is given by (7), where $t_{x\text{-coordinate}} = 22.07$.

$$s_M(u) = s_b(t_{x-cooridnate}, u) \otimes s_{0b}^*(t_{x-coordinate}, -u) \tag{7}$$

This filter uses $s_{0b}(t_{x\text{-coordinate}},u)$, represented as a column vector in Matlab, as a correlator for the signal $s_b(t_{x\text{-coordinate}},u)$, represented as a column vector in Matlab. The matched filter was implemented with the Matlab function xcorr. Similar to the reference signal matched filtering of range imaging, it is now required that the u axis of the output of the filter, $s_M(u)$ be shifted to account for the known y coordinate of the reference target. Since u = y, the time shifted output of the filter can be written as $s_M(y)$. The magnitude of $s_M(y)$ is shown in Figure 4.8. By observing the locations of large amplitude, the y coordinates (cross range) of the targets can be observed.

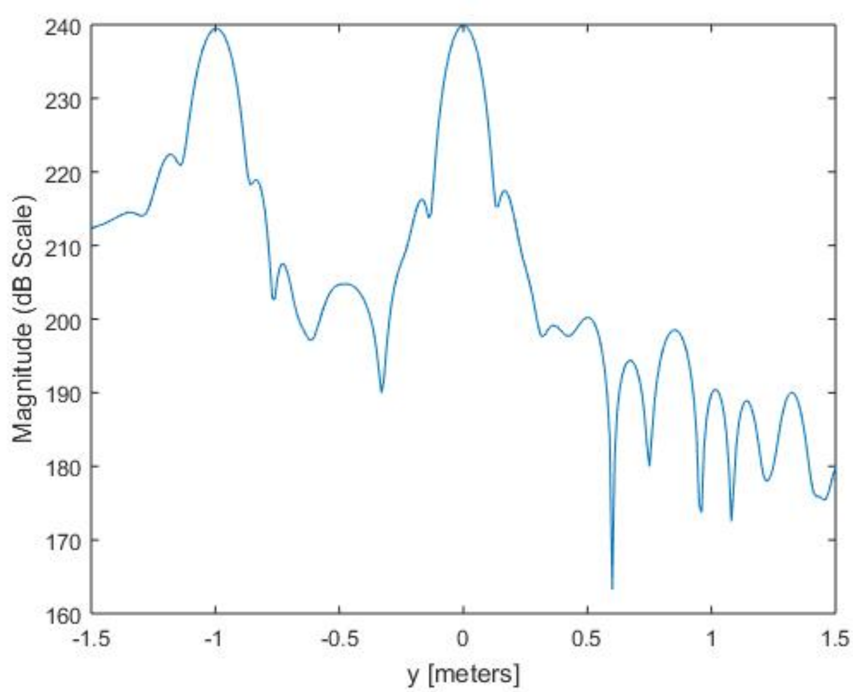

Figure 4.8 Cross Range Image of Target Scene with Targets at (3.31,-1) and (3.31,0)

4.3 Additional Cross Range Imaging Results

Section 4.2 described the cross range imaging algorithm used in this thesis. Results from a target scene with targets at y coordinates of -1 and 0 meters were used in the discussion. Both targets had an x coordinate of 3.31 meters. Section 4.3 presents the results for another target scene.

This target scene consists of targets located at the coordinates (6.63,0) and (6.63,-1). The target scene and the image of the magnitude of $s_b(t,u)$ produced from it are shown in Figure 4.9 and Figure 4.10, respectively. The real part of the signal $s_b(44.20,u)$, where t = 44.20 ns corresponds to a range of 6.63 meters, is shown in Figure 4.11. The reference target located at (6.63,0) and the image of the magnitude of the reference signal $s_{0b}(t,u)$ produced from the reference target are shown in Figure 4.12 and Figure 4.13 respectively. The real part of $s_{0b}(44.20,u)$, where t = 44.20 ns corresponds to a range of 6.63 meters, is shown in Figure 4.14. The magnitude of $s_M(y)$ is shown in Figure 4.15.

Figure 4.9 Cross Range Imaging Target Scene with Targets Located at (6.63,-1) and (6.63,0)

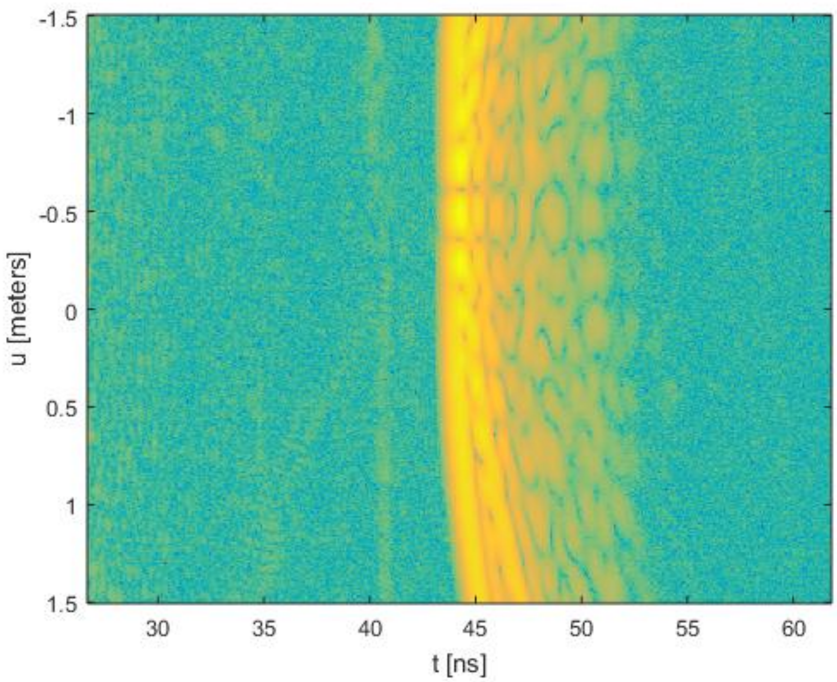

Figure 4.10 Cross Range Imaging Signal $|s_b(t,u)|$ from Target Scene with Targets Located at (6.63,-1) and (6.63,0)

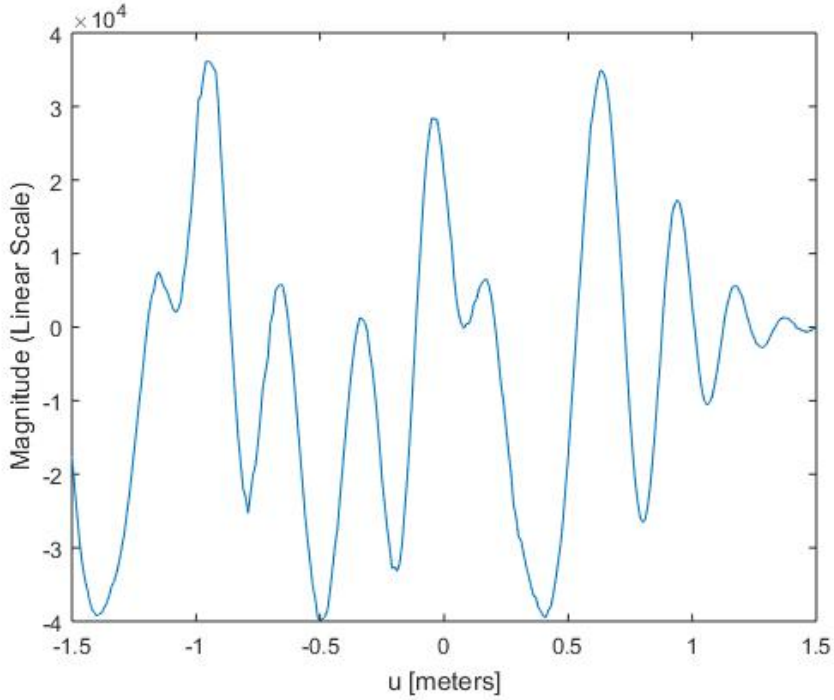

Figure 4.11 Cross Range Imaging Signal $Re[s_b(44.20,u)]$ from Target Scene with Targets Located at (6.63,-1) and (6.63,0)

Figure 4.12 Cross Range Imaging Reference Target Located at (6.63,0)

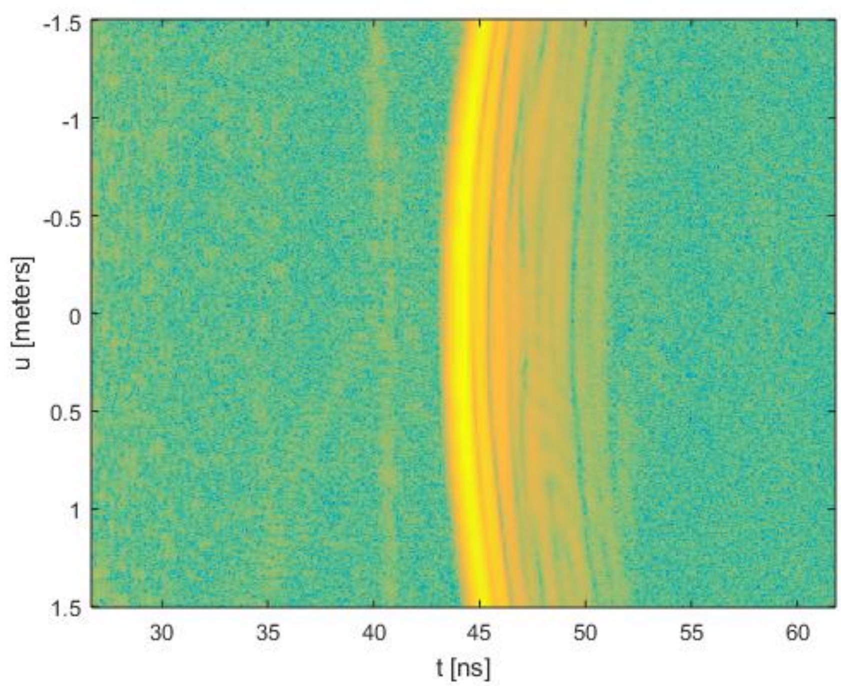

Figure 4.13 Cross Range Imaging Signal $s_{0b}(t,u)$ from Reference Target Located at (6.63,0)

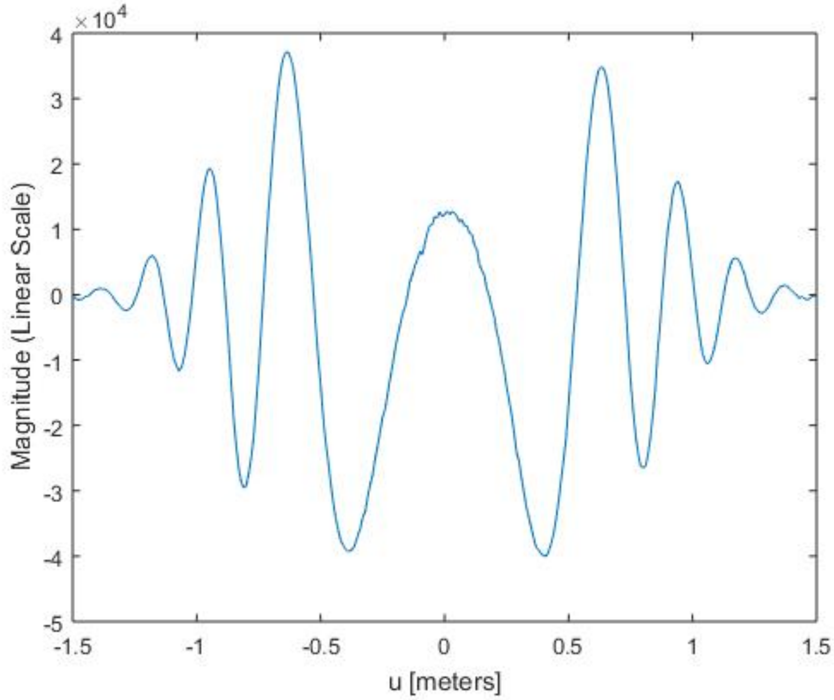

Figure 4.14 Cross Range Imaging Signal Re[s_{0b}(44.20,u)] from Reference Target Located at (6.63,0)

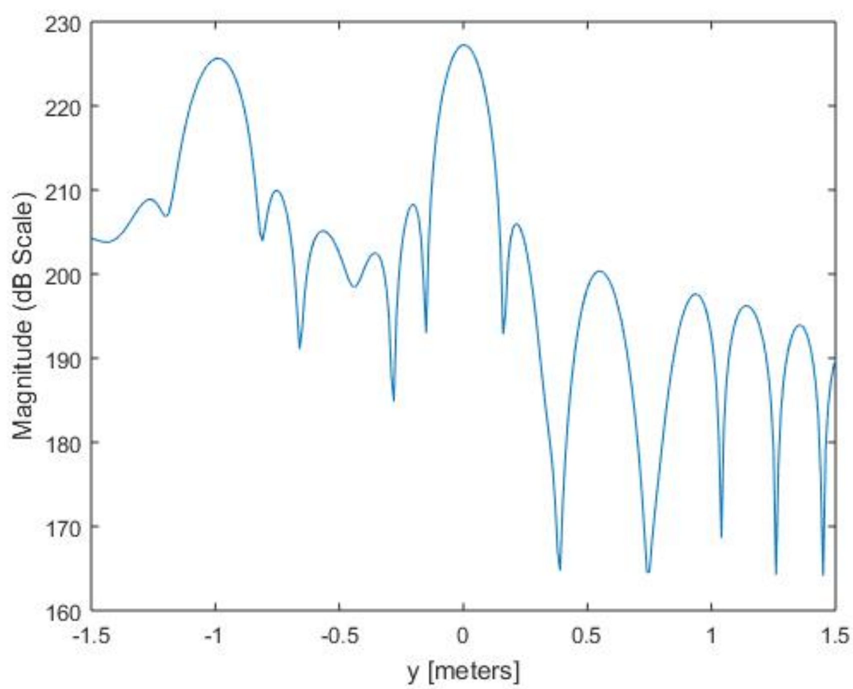

Figure 4.15 Cross Range Image of Target Scene with Targets at (6.63,-1) and (6.63,0)

CHAPTER FIVE

Synthetic Aperture Radar (SAR) Imaging

Chapter Three presented an algorithm for determining the range of targets from the radar. The range imaging algorithm was able to determine the range, but not the direction of the targets from the radar. Chapter Four presented an algorithm for determining the y coordinates (cross range) of targets all located at the same known x coordinate. This chapter presents an algorithm for the imaging of targets arranged in an unknown two dimensional alignment in a target scene. This is called synthetic aperture radar (SAR) imaging. This chapter describes a SAR imaging algorithm from Chapter 4 of [3] and presents the results from the Matlab implementation of the SAR imaging algorithm for several target scenes. Appendix D documents the Matlab program SAR_Imaging.m developed for this thesis. SAR_Imaging.m uses data acquired with Data_Collection.m, described in Chapter Two of this thesis.

5.1 SAR Imaging Basics

In Chapter Four, Figure 4.1 defined an axis definition used for cross range imaging. In this chapter, SAR imaging will use the same axis definition as was defined for cross range imaging. The axis is reshown in Figure 5.1 with five targets. The targets are not all at the same x coordinate as was the case in cross range imaging. The goal of the SAR imaging algorithm presented in this chapter is to produce an image of the target scene from which the x and y coordinates of the targets can be determined.

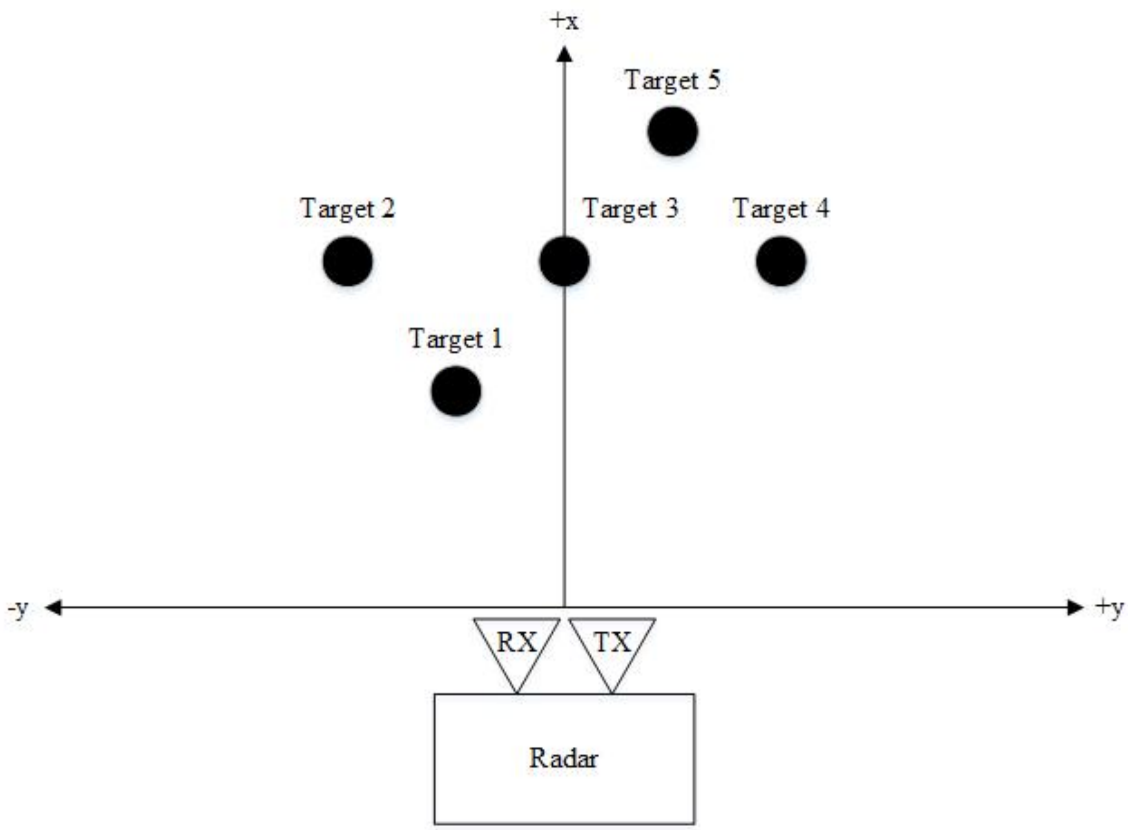

Figure 5.1 Synthetic Aperture Radar Axis Definition

As was the case with cross range imaging, for SAR imaging the location of the radar is allowed to vary. The process described in Section 4.1 for cross range imaging is used for SAR to form the two dimensional signal, $s_b(t,u)$. Based upon, $s_b(t,u)$, the SAR imaging algorithm produces an imaging of the target scene from which the x and y coordinates of the targets can be determined.

5.2 SAR Imaging Algorithm

Section 5.1 presented the basics of SAR imaging. This section describes the details of the SAR imaging algorithm by looking at results of the SAR imaging algorithm for the target scene shown in Figure 5.2. In Figure 5.2, five targets are placed in the target scene. The five targets are placed at the coordinates (2.44,-0.5), (3.31,-1), (3.31,0), (3.31,1), and (4.18,0.5). The units are in meters. The goal of the SAR imaging algorithm is to produce an image of the target scene from which the x and y coordinates of the five targets can be determined.

Figure 5.2 SAR Target Scene with Targets Located at (2.44,-0.5), (3.31,-1), (3.31,0), (3.31,1), and (4.18,0.5)

The radar starts at u = -1.50 meters. At this location, the radar transmits a signal p(t) and receives a return signal s(t) in the same way described in Chapter Three for range imaging. The return signal is then converted into a baseband (lowpass equivalent) signal, $s_b(t)$ as described in Chapter Three. Now, instead of just referring to the baseband return signal as $s_b(t)$, it will be referred to as $s_b(t,-1.50)$. The radar then moves 1 cm to the right to u = -1.49 meters. Again, at this location, the radar transmits a signal p(t) and receives a return signal which will be called s(t,-1.49). The signal s(t,-1.49) is converted to the signal s_b(t,-1.49). The signal s_b(t,-1.49) will be different than s_b(t,-1.50) since the relative positions of the targets to the radar are different. This process continues with the radar moving in 1 cm increments to the right and at each location, transmitting a signal p(t), receiving a signal s(t,u), and converting the received signal to a lowpass equivalent signal s_b(t,u). The last transmitted signal, p(t) is sent at the radar position u = +1.50 meters. At this location, the radar receives the signal s(t,+1.50) and converts it to the lowpass equivalent signal s_b(t,+1.50).

The two dimensional return signal, s_b(t,u) is formed where the lowpass equivalent return signal acquired at each location forms a row of s_b(t,u). In Matlab, s_b(t,u) is processed as an M x N matrix, where M is the number of rows and N is the number of columns. In the current case being described, M = 301 since the radar moved from u = -1.50 meters to u = +1.5 meters in 1 cm increments. An image of the magnitude of s_b(t,u) is shown in Figure 5.3.

In Chapter Three and Chapter Four, reference signal matched filtering was used in the range imaging algorithm and the cross range imaging algorithm, respectively. In the reference signal matched filtering of range imaging, the signal s_{0b}(t) from the reference target, a row vector in Matlab, was used as a correlator for the signal s_b(t), a row vector in Matlab. In the reference signal matched filtering of cross range imaging, the appropriate column vector from s_{0b}(t,u), from the reference target, was used as a correlator for the appropriate column vector from s_b(t,u), from the target scene. The SAR imaging algorithm presented in this chapter uses a two dimensional version of reference signal matched filtering.

A reference target located at (3.31,0) is shown in Figure 5.4. Using the same process that was used to obtain the two dimensional signal $s_b(t,u)$ of Figure 5.3, a two dimensional reference signal $s_{0b}(t,u)$ is formed from the reference target. An image of the magnitude of $s_{0b}(t,u)$ is shown in Figure 5.5.

For reference signal matched filtering for SAR imaging, the reference signal $s_{0b}(t,u)$ is used to create the filter for the target scene signal, $s_b(t,u)$. The Matlab function xcorr2 is used which is a two dimensional version of the Matlab function xcorr. The output of the xcorr2 function now requires axis realignment in both the time and u directions. After realignment, the time axis can be relabeled in terms of the x coordinate according to (2) and the u axis can be relabeled as the y axis since u = y. The image of the magnitude of the signal after axis realignment and relabeling is shown in Figure 5.6.

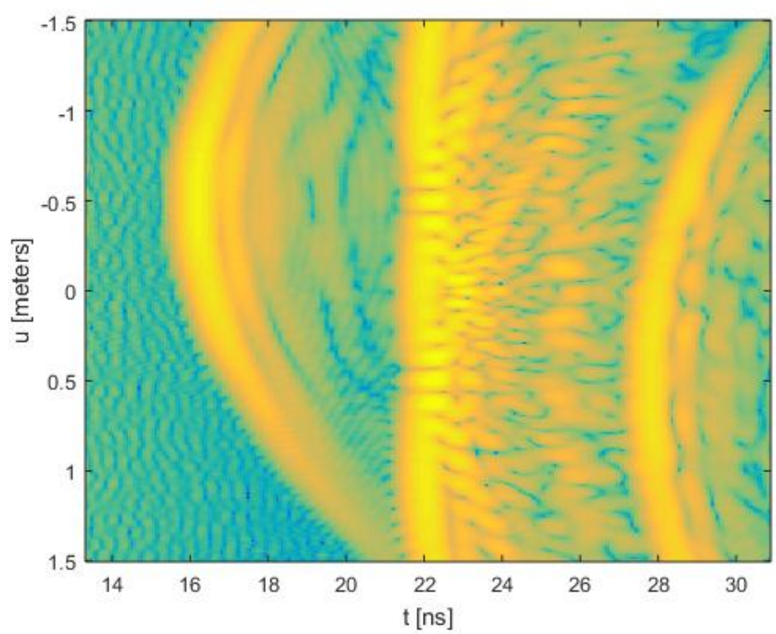

Figure 5.3 SAR Signal $|s_b(t,u)|$ from Target Scene with Targets Located at (2.44,-0.5), (3.31,-1), (3.31,0), (3.31,1), and (4.18,0.5)

Figure 5.4 SAR Reference Target Located at (3.31,0)

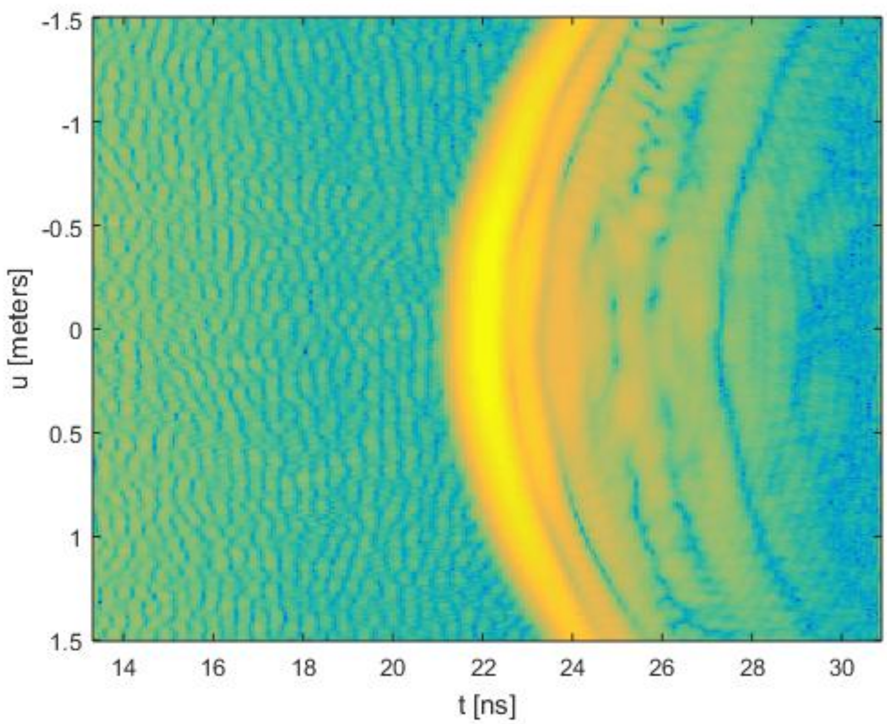

Figure 5.5 SAR Reference Signal $|s_{0b}(t,u)|$ from Reference Target Located at (3.31,0)

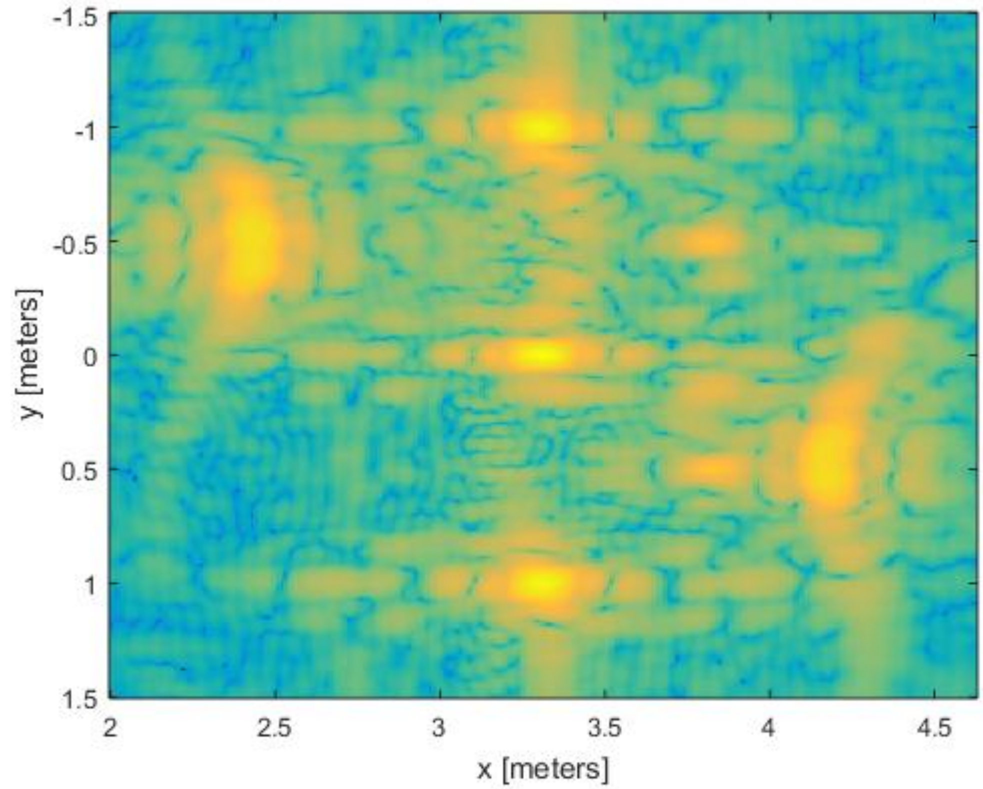

Figure 5.6 SAR Image of Target Scene with Targets at (2.44,-0.5), (3.31,-1), (3.31,0), (3.31,1), and (4.18,0.5)

5.3 Additional SAR Imaging Results

Section 5.2 described the SAR imaging algorithm used in this thesis. Results from a target scene with targets placed at the coordinates (2.44,-0.5), (3.31,-1), (3.31,0), (3.31,1), and (4.18,0.5) were used in the discussion. Section 5.3 presents the results for another target scene.

This target scene consists of targets located at the coordinates (5.76,-0.5), (6.63,-1), (6.63,0), (6.63,1), and (7.50,0.5). The target scene and the image of the magnitude of the signal $s_b(t,u)$ produced from the target scene are shown in Figure 5.7 and Figure 5.8 respectively. The reference target located at (6.63,0) and the image of the magnitude of the reference signal $s_{0b}(t,u)$ produced from the reference target are shown in Figure 5.9 and Figure 5.10 respectively. The image of the magnitude of the signal after reference signal matched filtering, axis realignment, and axis relabeling is shown in Figure 5.11.

Figure 5.7 SAR Target Scene with Targets Located at (5.76,-0.5), (6.63,-1), (6.63,0), (6.63,1), and (7.50,0.5)

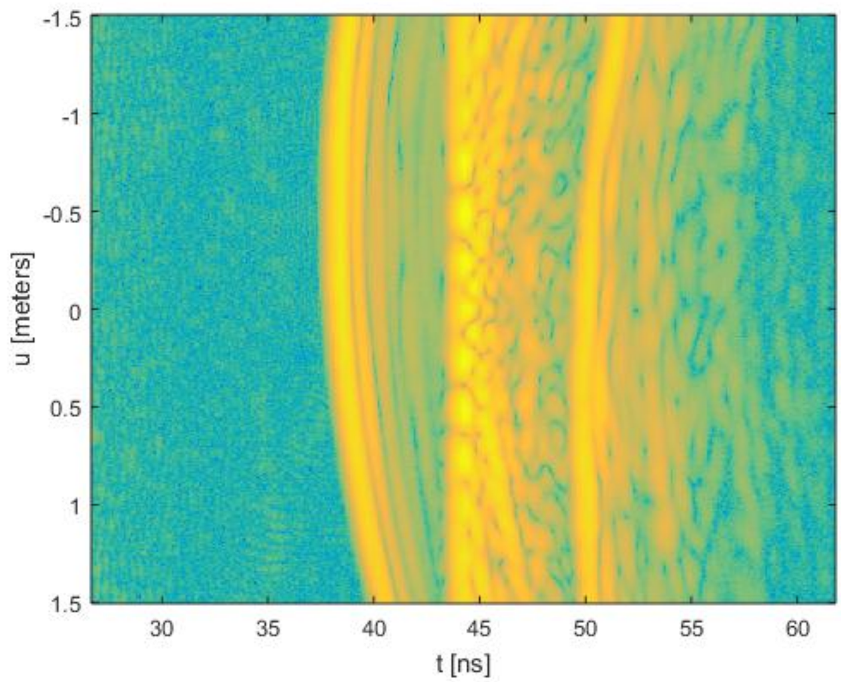

Figure 5.8 SAR Signal $|s_b(t,u)|$ from Target Scene with Targets Located at (5.76,-0.5), (6.63,-1), (6.63,0), (6.63,1), and (7.50,0.5)

Figure 5.9 SAR Reference Target Located at (6.63,0)

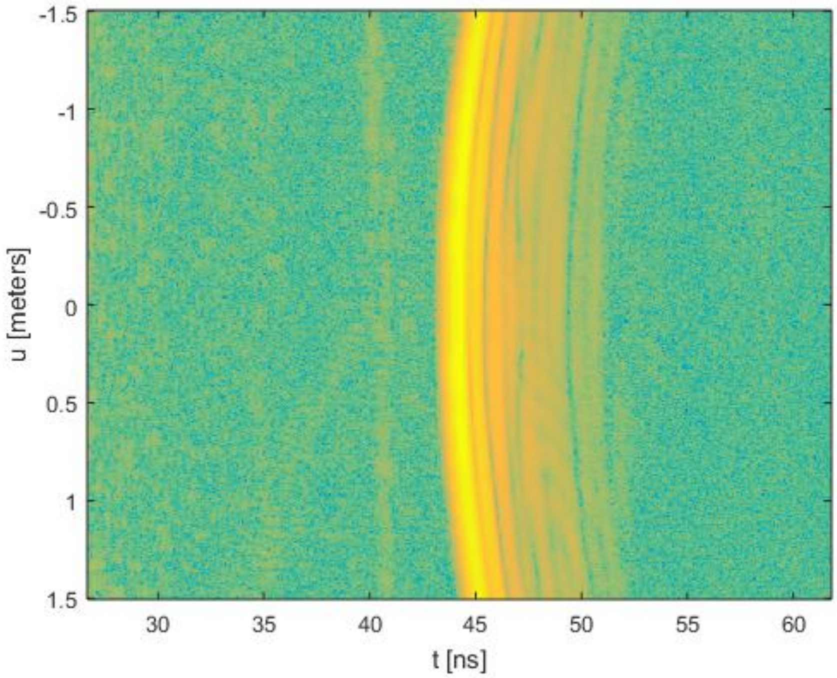

Figure 5.10 SAR Reference Signal $|s_{0b}(t,u)|$ from Reference Target Located at (6.63,0)

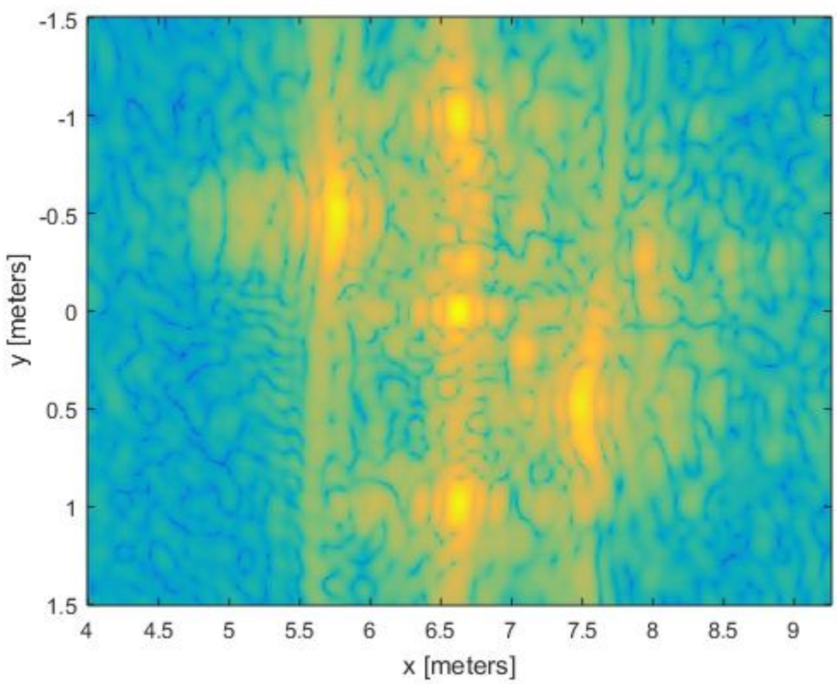

Figure 5.11 SAR Image of Target Scene with Targets at (5.76,-0.5), (6.63,-1), (6.63,0), (6.63,1), and (7.50,0.5)

CHAPTER SIX

Conclusion

6.1 Summary

This thesis documented the design and implementation of a radar imaging system. A rail system was built that could control the location of an off-the-shelf ultra-wideband (UWB) radar along a linear path. Matlab was used to command the radar and to control a stepper motor to move the radar. It was also used to implement the range imaging, cross range imaging, and synthetic aperture radar (SAR) imaging algorithms. The radar imaging system can be used for future research at Baylor University. Future researchers can set up their own target scenes of interest and implement the algorithms presented in this thesis, other established algorithms from the literature, or their own novel algorithms. The system can also be used for classroom demonstration in the course ELC 5340 "Radar Engineering" at Baylor University. Students in the course can participate in the data acquisition process using the radar imaging system. The data collected can then be distributed to the students so that they can implement algorithms such as those presented in this thesis.

6.2 Future Work

This section presents ideas that may be used for either the improvement of the radar imaging system presented or the improvement of the algorithms presented. The ideas in this section may also be used for radar course projects or homework assignments using data acquired by the radar imaging system.

The reference signal matched filtering, presented in Chapter Three, Chapter Four, and Chapter Five, for range imaging, cross range imaging, and SAR imaging, respectively, uses a convolution based form of reference signal matched filtering using the Matlab functions xcorr and xcorr2. The reference signal matched filtering could be implemented without the use of the Matlab functions mentioned. This would require a lower level implementation of the convolution involved. The reference signal matched filtering could also be implemented by the use of fast Fourier transforms, multiplication, and inverse fast Fourier transforms.

For range imaging, the band pass return signals s(t) and thus the low pass equivalent return signals $s_b(t)$ had echoes from the targets that were much larger than the signal noise. Because of this, the benefit of reference signal matched filtering is not clear in the data presented. By lowering the pulse integration index (PII), using targets with lower reflectivity, or setting the targets at a further range, the SNR of the return signal would become smaller. Using reference signal matched filtering for this scenario would more clearly demonstrate its utility.

The reference signal matched filtering of range imaging could possibly be improved in the following way. The reference signal matched filter is created from return signals from a reference target at a known range. A window could be applied to the data in the time domain, such that the amplitude of the data that is not associated with the echo from the reference target is lowered. This could improve the SNR of the result after reference signal matched filtering, making the ranges of the targets more distinct. It believed by the author that this windowing technique applied in the u domain for the reference signal matched filtering for cross range imaging would be detrimental to the final results. This technique could also prove useful for the reference signal matched filter for SAR imaging if applied to the matrix $s_b(t,u)$ in the time direction. For the same reasoning as mentioned for cross range imaging, it is believed by the author that windowing in the u direction for reference signal matched filtering for SAR imaging would be detrimental to the final results.

Section 4.2 described the details of the cross range imaging algorithm by looking at results of the cross range imaging algorithm for the target scene shown in Figure 4.2. The reference target, located at (3.31,0) was shown in Figure 4.5. The real part of the reference signal $s_{0b}(22.07,u)$, where t = 22.07 ns was the time corresponding to the x coordinate (x = 3.31 meters) of both the targets in the target scene and the reference target according to (2), was shown in in Figure 4.7. Section 4.3.2 presented results similar to that of Section 4.2 except the targets and reference target presented in Section 4.2 were at an x coordinate of 6.63 meters, corresponding to t = 44.20 ns. The real part of the reference signal $s_{0b}(44.20,u)$ was shown in Figure 4.14. By comparing Figure 4.7 and Figure 4.14, it can be observed that Figure 4.14 is a stretched version of Figure 4.17. This demonstrates that there is a variation of the real part of the reference signals from reference targets located at different x coordinates. The imaginary part of the reference signals and the magnitude of the reference signal was not discussed in this thesis, but it is believed by the author that these would vary also. Even without analyzing the magnitudes and the imaginary parts, the variation of the real part is sufficient to show that there is a variation of the reference signal $s_{0b}(t_{x\text{-coordiante}},u)$ as the reference target x coordinate varies. This was not a problem in cross range imaging, since the reference signal matched filter was always created from a reference target at the same x coordinate as the targets. In the two dimensional reference signal matched filtering of SAR imaging, the variation was assumed to be sufficiently small. By creating multiple reference signal matched filters from reference targets located at different x coordinates, it may be possible to improve the two dimensional signal matched filtering of SAR imaging. For a given time value in the signal $s_b(t,u)$, the "closest fit" reference signal matched filter could be used.

The radar imaging system described in this thesis moves the radar in small steps along the rail. At each position, the radar stops, transmits an electromagnetic signal, and receives an electromagnetic signal. Since the radar is not moving during the time it is transmitting and receiving, there is not a Doppler frequency associated with the return signal. Many SAR implementations, such as airborne SAR systems, use continuous radar movement, and thus have Doppler frequencies in their received signals. If the radar imaging system was modified such that the radar moved continuously along the length of the rail, this would allow the user to develop algorithms to take into account the Doppler frequencies involved.

One major difficulty of the project was finding a desirable location to perform the experimental work to produce the results presented in this thesis. Desirable features of the location found include a flat surface such that the targets could be accurately placed and a location without unintentional targets near the target scene. Flat surfaces were difficult to locate at outside locations. Locations without unintentional targets near the target scene were difficult to locate at indoor location since most indoor locations have walls and other reflective features. A desirable location was eventually located and the results of this thesis were produced at that location. It is not expected in future work, that such a desirable location will always be available. It may be possible to use an indoor location by manually removing the data from the bandpass return signals that corresponds to ranges greater than the maximum range of the target scene.

.

APPENDICES

APPENDIX A

Data_Collection.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%File Name: Data_Collection.m
%Author: Brian Ernzen
%Current Revision: - (Initial Release)  Released 03/06/2016

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
clc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%




%%%%% ADJUSTABLE PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
radar_com_port_num = 5;
motor_com_port_num = 'COM6';
R1 = 4.5;   %[meters] 2 is default value
R2 = 6.5;  %[meters] 12 is default value
Gtx = 63; % 63 is default value
PII = 10;  % 7 is default value
dT0 = 10; %[ns] 10 is default value. This value is adjusted for
          %different antennas and cable lengths.

Antenna = 'Horn'; %This text should reflect the antenna used
                  %during the test.

File_Name = 'SCN_08092015_1.mat';  %This text will be the name
                                   %of the output file.

Ktry_max1 = 1000; % 10 is default value. This value controls timeout
                  %limit for one of the USB communication instances.
Ktry_max2 = 1000; % 10 is default value. This value controls timeout
                  %limit for one of the USB communication instances.
Ktry_max3 = 1000; % 10 is default value. This value controls timeout
                  %limit for one of the USB communication instances.
Ktry_max4 = 1000; % 10 is default value. This value controls timeout
                  %limit for one of the USB communication instances.


Step_Size = 0.01; %[meter] Chose from the following
                  %values: 0.01, 0.001, 0.0005, 0.00025
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Number_of_Scans = (3/Step_Size)+1; %Since radar movement distance is
                                   %3 meters
                                   %(-1.5 meters to +1.5 meters)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%% GIVE 10 SECONDS FOR USER TO MOVE AWAY FROM TEST SETUP %%%%%%%%%%%%
disp('Move away from the test')
disp('Radar scanning will commence in 10 seconds')
pause(1)
disp('Radar scanning will commence in 9 seconds')
pause(1)
disp('Radar scanning will commence in 8 seconds')
pause(1)
disp('Radar scanning will commence in 7 seconds')
pause(1)
disp('Radar scanning will commence in 6 seconds')
pause(1)
disp('Radar scanning will commence in 5 seconds')
pause(1)
disp('Radar scanning will commence in 4 seconds')
pause(1)
disp('Radar scanning will commence in 3 seconds')
pause(1)
disp('Radar scanning will commence in 2 seconds')
pause(1)
disp('Radar scanning will commence in 1 second')
pause(1)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%% MOTOR SETUP %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s = serial(motor_com_port_num);
set(s, 'terminator', 'CR');
fopen(s)

       %Per motor manufacturer: 20,000 steps gives one
       %motor revolution
       %It was empirically found by BTE that 1,788,000 steps
       %moves 3 meters
       %Then 596,000 steps -> 1 meter
       %        5960 steps -> 1 cm = 0.01 meters
       %         596 steps -> 1mm = 0.001 meters
       %         298 steps -> 0.5mm = 0.0005 meters
       %         149 steps -> 0.25mm = 0.00025 meters
```

```
if Step_Size == 0.01
    motor_steps_per_move = 'DI5960'

elseif Step_Size ==0.001
    motor_steps_per_move = 'DI596'

elseif Step_Size == 0.0005
    motor_steps_per_move = 'DI298'

elseif Step_Size == 0.00025
    motor_steps_per_move = 'DI149'

else
    motor_steps_per_move = 'DI1'
end

fprintf(s, motor_steps_per_move) %Sets number of motor steps per move
fprintf(s, 'AC2')   %Sets motor acceleration rate
fprintf(s, 'DE2')   %Sets motor deceleration rate
fprintf(s, 'VE.5')  %Sets motor speed

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%% Radar COM PORT SETUP %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
srl = open_com_port(radar_com_port_num)
%srl = instrfind

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%% TAKE RADAR SCANS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for n = 1:Number_of_Scans

     if n > 1
          fprintf(s, 'FL')        %Move motor before each scan except
                                  %before first scan
          pause(2)                %Allow time for motor to reach
                                  %next position
     end

     [Rbin,SCN,Taxis] = ...
         plot_one_scn(...
         srl,R1,R2,Gtx,PII,dT0,...
         Ktry_max1,Ktry_max2,Ktry_max3,Ktry_max4);

     if n == 1
         SCN_MATRIX = zeros(...
             Number_of_Scans,size(SCN,2));  %Preallocate SCN_MATRIX
     end
```

```
    SCN_MATRIX(n,:) = SCN;
end

radar_position_column = ...
    zeros(Number_of_Scans,1);  %Preallocate radar_position_column
for n = 1:Number_of_Scans
    radar_position_column(n,:) = ...
        (n-(ceil(Number_of_Scans/2)))*Step_Size;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%% MOVE RADAR BACK TO START AND CLOSE COMMUNICATION WITH MOTOR %%%%%
fprintf(s, 'DI-1788000')  %Move radar back to the
                          %beginning (3 meter movement)
fprintf(s, 'FL')
fclose(s)
delete(s)
clear s

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%% CLOSE COMMUNICATION WITH RADAR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fclose(srl);
delete(srl)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%% OUTPUTS OF THIS PROGRAM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SCN_MATRIX;
Rbin; %[meters]
Taxis; %[ns]
radar_position_column; %[meters]
Gtx;
PII;
dT0;
Antenna;

save(File_Name, 'SCN_MATRIX', 'Rbin', 'Taxis', ...
    'radar_position_column', 'Gtx', 'PII', 'dT0', 'Antenna');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX B

Range_Imaging.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%File Name: Range_Imaging.m
%Author: Brian Ernzen
%Current Revision: - (Initial Release)  Released 03/06/2016

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
clc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% ADJUSTABLE PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Reference_Data_Filename = 'SCN_01182016_20';
Reference_Position_X_Value = 6.63; %[meters]
Reference_Position_Y_Value = 0; %[meters]

Target_Data_Filename = 'SCN_01182016_21';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% DEFINED CONSTANTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fc = 4.3; %[GHz] Center frequency

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% LOAD THE REFERENCE DATA FROM FILE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Reference_Data = load(Reference_Data_Filename);

Reference_Rbin = Reference_Data.Rbin; %[meters]
Reference_Taxis = Reference_Data.Taxis; %[ns]
Reference_radar_position_column = ...
    Reference_Data.radar_position_column; %[meters]
Reference_SCN = Reference_Data.SCN_MATRIX(find(...
    Reference_radar_position_column==0),:);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% LOAD THE TARGET DATA FROM FILE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Target_Data = load(Target_Data_Filename);

Target_Rbin = Target_Data.Rbin; %[meters]
Target_Taxis = Target_Data.Taxis; %[ns]
Target_radar_position_column = ...
    Target_Data.radar_position_column; %[meters]
Target_SCN = Target_Data.SCN_MATRIX(find(...
    Target_radar_position_column==0),:);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% CREATE REFERENCE BANDPASS, ANALYTIC, AND LPE SCANS %%%%%%%%%%%%%%%
Reference_Bandpass_SCN = Reference_SCN;

Reference_Analytic_SCN = hilbert(Reference_Bandpass_SCN);

for n = 1:size(Reference_Analytic_SCN,2)
    Reference_LPE_SCN(n) = ...
        Reference_Analytic_SCN(n) * ...
        exp(-1 * 1i * 2 * pi * fc * Reference_Taxis(n));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% CREATE TARGET BANDPASS, ANALYTIC, AND LPE SCANS %%%%%%%%%%%%%%%%%%
Target_Bandpass_SCN = Target_SCN;

Target_Analytic_SCN = hilbert(Target_Bandpass_SCN);

for n = 1:size(Target_Analytic_SCN,2)
    Target_LPE_SCN(n) = ...
        Target_Analytic_SCN(n) * ...
        exp(-1 * 1i * 2 * pi * fc * Target_Taxis(n));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% TAKE FFT OF REFERENCE BANDPASS, ANALYTIC, AND LPE SCANS %%%%%%%%%%
FFT_Reference_Bandpass_SCN = fftshift(fft(Reference_Bandpass_SCN));

FFT_Reference_Analytic_SCN = fftshift(fft(Reference_Analytic_SCN));
FFT_Reference_LPE_SCN = fftshift(fft(Reference_LPE_SCN));
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% TAKE FFT OF TARGET BANDPASS, ANALYTIC, AND LPE SCANS %%%%%%%%%%%%
FFT_Target_Bandpass_SCN = fftshift(fft(Target_Bandpass_SCN));

FFT_Target_Analytic_SCN = fftshift(fft(Target_Analytic_SCN));

FFT_Target_LPE_SCN = fftshift(fft(Target_LPE_SCN));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% PLOT SPECTRUM OF REFERENCE AND TARGET BANDPASS, %%%%%%%%%%%%%%%%%%
%%%% ANALYTIC, AND LPE SCANS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
F_axis = (-1*size(Reference_Bandpass_SCN,2)/2:(size(...
    Reference_Bandpass_SCN,2)/2)-1) / (((...
    Reference_Taxis(2)-Reference_Taxis(1)))*(size(...
    Reference_Bandpass_SCN,2)));

figure
plot(F_axis,abs(FFT_Reference_Bandpass_SCN));
xlabel('Frequency [GHz]');
ylabel('Magnitude (Linear Scale)');
%title('Spectrum of Reference Bandpass Signal');

figure
plot(F_axis,abs(FFT_Reference_Analytic_SCN));
xlabel('Frequency [GHz]');
ylabel('Magnitude (Linear Scale)');
%title('Spectrum of Reference Analytic Signal');

figure
plot(F_axis,abs(FFT_Reference_LPE_SCN));
xlabel('Frequency [GHz]');
ylabel('Magnitude (Linear Scale)');
%title('Spectrum of Reference LPE Signal');

figure
plot(F_axis,abs(FFT_Target_Bandpass_SCN));
xlabel('Frequency [GHz]');
ylabel('Magnitude (Linear Scale)');
%title('Spectrum of Target Bandpass Signal');

figure
plot(F_axis,abs(FFT_Target_Analytic_SCN));
xlabel('Frequency [GHz]');
ylabel('Magnitude (Linear Scale)');
%title('Spectrum of Target Analytic Signal');

figure
plot(F_axis,abs(FFT_Target_LPE_SCN));
```

```
xlabel('Frequency [GHz]');
ylabel('Magnitude (Linear Scale)');
%title('Spectrum of Target LPE Signal');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% PLOT REFERENCE AND TARGET BANDPASS SIGNALS VS. TIME AND RANGE %%%%
figure
plot(Reference_Taxis,Reference_Bandpass_SCN);
%title('Reference Bandpass Signal')
xlabel('Time [ns]');
ylabel('Magnitude (Linear Scale)');

figure
plot(Reference_Rbin,Reference_Bandpass_SCN);
%title('Reference Bandpass Signal vs Range')
xlabel('Range [meters]');
ylabel('Magnitude (Linear Scale)');

figure
plot(Target_Taxis,Target_Bandpass_SCN);
%title('Target Bandpass Signal')
xlabel('Time [ns]');
ylabel('Magnitude (Linear Scale)');

figure
plot(Target_Rbin,Target_Bandpass_SCN);
%title('Target Bandpass Signal vs Range')
xlabel('Range [meters]');
ylabel('Magnitude (Linear Scale)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% PLOT MAGNITUDE OF REFERENCE AND TARGET LPE SIGNALS %%%%%%%%%%%%%%%
%%%% VS. TIME AND RANGE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure
plot(Reference_Taxis,20*log10(abs(Reference_LPE_SCN)));
%title('Reference LPE Signal')
xlabel('Time [ns]');
ylabel('Magnitude (dB Scale)');

figure
plot(Reference_Rbin,20*log10(abs(Reference_LPE_SCN)));
%title('Reference LPE Signal vs Range')
xlabel('Range [meters]');
ylabel('Magnitude (dB Scale)');

figure
plot(Target_Taxis,20*log10(abs(Target_LPE_SCN)));
%title('Target LPE Signal')
```

```
xlabel('Time [ns]');
ylabel('Magnitude (dB Scale)');

figure
plot(Target_Rbin,20*log10(abs(Target_LPE_SCN)));
%title('Target LPE Signal vs Range')
xlabel('Range [meters]');
ylabel('Magnitude (dB Scale)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% MATCHED FILTER WITH CONVOLUTION USING THE MATLAB FUNCTION XCORR %%
Reference_Range = ...
    sqrt((Reference_Position_X_Value^2)+(...
    Reference_Position_Y_Value^2));

Find_Reference_Index = Reference_Rbin - Reference_Range;
[Var_Not_Used,Reference_Taxis_Rbin_Index] = ...
    min(abs(Find_Reference_Index));

Cross_Corr = xcorr(Target_LPE_SCN,Reference_LPE_SCN);

if Reference_Taxis_Rbin_Index > 1
    Cross_Corr(1:size(...
        Reference_LPE_SCN,2) - Reference_Taxis_Rbin_Index) = [];
end

if size(Cross_Corr,2) > size(Reference_LPE_SCN,2)
    Cross_Corr(size(Reference_LPE_SCN,2)+1:size(Cross_Corr,2)) = [];
end

Matched_Filtered_Target_LPE_SCN = Cross_Corr;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% PLOT MAGNITUDE OF MATCHED FILTERED TARGET LPE SIGNAL %%%%%%%%%%%%%
%%%% VS. TIME AND RANGE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure
plot(Target_Taxis,20*log10(abs(Matched_Filtered_Target_LPE_SCN)));
%title('Matched Filtered Target LPE Signal')
xlabel('Time [ns]');
ylabel('Magnitude (dB Scale)');

figure
plot(Target_Rbin,20*log10(abs(Matched_Filtered_Target_LPE_SCN)));
%title('Matched Filtered Target LPE Signal vs Range')
xlabel('Range [meters]');
ylabel('Magnitude (dB Scale)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX C

Cross_Range_Imaging.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%File Name: Cross_Range_Imaging.m
%Author: Brian Ernzen
%Current Revision: - (Initial Release)  Released 03/06/2016

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
clc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% ADJUSTABLE PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Reference_Data_Filename = 'SCN_01182016_20';
Reference_Position_X_Value = 6.63; %[meters]
Reference_Position_Y_Value = 0; %[meters]

Target_Data_Filename = 'SCN_01182016_22';
Target_Position_X_Value = 6.63; %[meters]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% DEFINED CONSTANTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fc = 4.3; %[GHz] Center frequency

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% LOAD THE REFERENCE DATA FROM FILE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Reference_Data = load(Reference_Data_Filename);

Reference_Rbin = Reference_Data.Rbin; %[meters]
Reference_Taxis = Reference_Data.Taxis; %[ns]
Reference_radar_position_column = ...
    Reference_Data.radar_position_column; %[meters]
Reference_SCN_MATRIX = Reference_Data.SCN_MATRIX;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% LOAD THE TARGET DATA FROM FILE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Target_Data = load(Target_Data_Filename);

Target_Rbin = Target_Data.Rbin; %[meters]
Target_Taxis = Target_Data.Taxis; %[ns]
Target_radar_position_column = ...
    Target_Data.radar_position_column; %[meters]
Target_SCN_MATRIX = Target_Data.SCN_MATRIX;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% CREATE REFERENCE BANDPASS, ANALYTIC, AND LPE SCAN MATRICES %%%%%%%
Reference_Bandpass_SCN_MATRIX = Reference_SCN_MATRIX;

for n = 1:size(Reference_Bandpass_SCN_MATRIX,1)
    Reference_Analytic_SCN_MATRIX(n,:) = ...
        hilbert(Reference_Bandpass_SCN_MATRIX(n,:));
end

for n = 1:size(Reference_Analytic_SCN_MATRIX,2)
    Reference_LPE_SCN_MATRIX(:,n) = ...
        Reference_Analytic_SCN_MATRIX(:,n) * ...
        exp(-1 * 1i * 2 * pi * fc * Reference_Taxis(n));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% CREATE TARGET BANDPASS, ANALYTIC, AND LPE SCAN MATRICES %%%%%%%%%%
Target_Bandpass_SCN_MATRIX = Target_SCN_MATRIX;

for n = 1:size(Target_Bandpass_SCN_MATRIX,1)
    Target_Analytic_SCN_MATRIX(n,:) = ...
        hilbert(Target_Bandpass_SCN_MATRIX(n,:));
end

for n = 1:size(Target_Analytic_SCN_MATRIX,2)
    Target_LPE_SCN_MATRIX(:,n) = ...
        Target_Analytic_SCN_MATRIX(:,n) * ...
        exp(-1 * 1i * 2 * pi * fc * Target_Taxis(n));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%% CREATE IMAGES OF REFERENCE AND TARGET LPE SCAN MATRICES %%%%%%%%%
figure
imagesc(Reference_Taxis,Reference_radar_position_column,20*log10(...
    abs(Reference_LPE_SCN_MATRIX)));
%title('Reference LPE s_0(t,u) (dB Scale)');
xlabel('t [ns]');
ylabel('u [meters]');

figure
imagesc(Target_Taxis,Target_radar_position_column,20*log10(...
    abs(Target_LPE_SCN_MATRIX)));
%title('Target LPE s(t,u) (dB Scale)');
xlabel('t [ns]');
ylabel('u [meters]');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% FIND THE INDEX OF THE X POSITION VALUE %%%%%%%%%%%%%%%%%%%%%%%%%%%
Find_X_Position_Index = Reference_Rbin - Reference_Position_X_Value;
[Var_Not_Used,X_Position_Index] = min(abs(Find_X_Position_Index));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% PLOT REAL PART OF REFERENCE_LPE_SCN_MATRIX AND %%%%%%%%%%%%%%%%%%%
%%%% TARGET_LPE_SCN_MATRIX FOR T=TIME CORRESPONDING TO %%%%%%%%%%%%%%%%
%%%% INDEX OF X POSITION VALUE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% THIS IS PLOTTING A COLUMN OF EACH MATRIX %%%%%%%%%%%%%%%%%%%%%%%%%%
figure
plot(Reference_radar_position_column,real(...
    Reference_LPE_SCN_MATRIX(:,X_Position_Index)))
%title({'Spherical PM Signal of Reference';['Re[s_0(t=' num2str(...
    %Reference_Taxis(X_Position_Index)) ',u)]']});
xlabel('u [meters]');
ylabel('Magnitude (Linear Scale)');

figure
plot(Target_radar_position_column,real(...
    Target_LPE_SCN_MATRIX(:,X_Position_Index)))
%title({'Spherical PM Signal of Target';['Re[s(t=' num2str(...
    %Target_Taxis(X_Position_Index)) ',u)]']});
xlabel('u [meters]');
ylabel('Magnitude (Linear Scale)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% MATCHED FILTER IN THE U DIRECTION WITH CONVOLUTION USING %%%%%%%%%
%%%% THE MATLAB FUNCTION XCORR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Reference_Position_Y_Value;
```

```matlab
 Find_Reference_Y_Index = ...
    Reference_radar_position_column - Reference_Position_Y_Value;
[Var_Not_Used_2,Reference_Y_Index] = min(abs(Find_Reference_Y_Index));



for n = 1:size(Reference_LPE_SCN_MATRIX,2)

    Cross_Corr = xcorr(...
        Target_LPE_SCN_MATRIX(:,n),Reference_LPE_SCN_MATRIX(:,n));

    if Reference_Y_Index > 1
        Cross_Corr(...
            1:size(...
            Reference_LPE_SCN_MATRIX,1) - Reference_Y_Index) = [];
    end

    if size(Cross_Corr,1) > size(Reference_LPE_SCN_MATRIX,1)
        Cross_Corr(size(Reference_LPE_SCN_MATRIX,1)+1:size(...
            Cross_Corr,1)) = [];
    end

    Target_SCN_MATRIX_match_filt_u(:,n) = Cross_Corr;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% PLOT CROSS RANGE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure
plot(Target_radar_position_column,20*log10(...
    abs((Target_SCN_MATRIX_match_filt_u(:,X_Position_Index)))));
%title(['Cross Range at x = ' num2str(...
    %Reference_Position_X_Value) ' meters']);
xlabel('y [meters]');
ylabel('Magnitude (dB Scale)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX D

SAR_Imaging.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%File Name: SAR_Imaging.m
%Author: Brian Ernzen
%Current Revision: - (Initial Release)  Released 03/06/2016

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
clc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% ADJUSTABLE PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Reference_Data_Filename = 'SCN_01182016_20';
Reference_Position_X_Value = 6.63; %[meters]
Reference_Position_Y_Value = 0; %[meters]

Target_Data_Filename = 'SCN_01182016_23';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% DEFINED CONSTANTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fc = 4.3; %[GHz] Center frequency

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% LOAD THE REFERENCE DATA FROM FILE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Reference_Data = load(Reference_Data_Filename);

Reference_Rbin = Reference_Data.Rbin; %[meters]
Reference_Taxis = Reference_Data.Taxis; %[ns]
Reference_radar_position_column = ...
    Reference_Data.radar_position_column; %[meters]
Reference_SCN_MATRIX = Reference_Data.SCN_MATRIX;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%% LOAD THE TARGET DATA FROM FILE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Target_Data = load(Target_Data_Filename);

Target_Rbin = Target_Data.Rbin; %[meters]
Target_Taxis = Target_Data.Taxis; %[ns]
Target_radar_position_column = ...
    Target_Data.radar_position_column; %[meters]
Target_SCN_MATRIX = Target_Data.SCN_MATRIX;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% CREATE REFERENCE BANDPASS, ANALYTIC, AND LPE SCAN MATRICES %%%%%%%
Reference_Bandpass_SCN_MATRIX = Reference_SCN_MATRIX;

for n = 1:size(Reference_Bandpass_SCN_MATRIX,1)
    Reference_Analytic_SCN_MATRIX(n,:) = ...
        hilbert(Reference_Bandpass_SCN_MATRIX(n,:));
end

for n = 1:size(Reference_Analytic_SCN_MATRIX,2)
    Reference_LPE_SCN_MATRIX(:,n) = ...
        Reference_Analytic_SCN_MATRIX(:,n) * ...
        exp(-1 * 1i * 2 * pi * fc * Reference_Taxis(n));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% CREATE TARGET BANDPASS, ANALYTIC, AND LPE SCAN MATRICES %%%%%%%%%%
Target_Bandpass_SCN_MATRIX = Target_SCN_MATRIX;

for n = 1:size(Target_Bandpass_SCN_MATRIX,1)
    Target_Analytic_SCN_MATRIX(n,:) = ...
        hilbert(Target_Bandpass_SCN_MATRIX(n,:));
end

for n = 1:size(Target_Analytic_SCN_MATRIX,2)
    Target_LPE_SCN_MATRIX(:,n) = ...
        Target_Analytic_SCN_MATRIX(:,n) * ...
        exp(-1 * 1i * 2 * pi * fc * Target_Taxis(n));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% CREATE IMAGES OF REFERENCE AND TARGET LPE SCAN MATRICES %%%%%%%%%%
figure
imagesc(Reference_Taxis,Reference_radar_position_column,20*log10(...
    abs(Reference_LPE_SCN_MATRIX)));
```

```
%title('Reference LPE s_0(t,u) (dB Scale)');
xlabel('t [ns]');
ylabel('u [meters]');

figure
imagesc(Target_Taxis,Target_radar_position_column,20*log10(...
    abs(Target_LPE_SCN_MATRIX)));
%title('Target LPE s(t,u) (dB Scale)');
xlabel('t [ns]');
ylabel('u [meters]');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% FIND THE INDEX OF THE REFERENCE X POSITION VALUE %%%%%%%%%%%%%%%%%
Find_Reference_X_Index = Reference_Rbin - Reference_Position_X_Value;
[Var_Not_Used,Reference_X_Index] = min(abs(Find_Reference_X_Index));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% FIND THE INDEX OF THE REFERENCE Y POSITION VALUE %%%%%%%%%%%%%%%%%
Find_Reference_Y_Index = ...
    Reference_radar_position_column - Reference_Position_Y_Value;
[Var_Not_Used_2,Reference_Y_Index] = min(abs(Find_Reference_Y_Index));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% 2D MATCHED FILTER WITH CONVOLUTION USING %%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% THE MATLAB FUNCTION XCORR2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Cross_Corr = xcorr2(Target_LPE_SCN_MATRIX,Reference_LPE_SCN_MATRIX);

if Reference_X_Index > 1
    Cross_Corr(:,1:size(...
        Reference_LPE_SCN_MATRIX,2) - Reference_X_Index) = [];
end

if size(Cross_Corr,2) > size(Reference_LPE_SCN_MATRIX,2)
    Cross_Corr(:,size(...
        Reference_LPE_SCN_MATRIX,2)+1:size(Cross_Corr,2)) = [];
end

if Reference_Y_Index > 1
    Cross_Corr(1:size(...
        Reference_LPE_SCN_MATRIX,1) - Reference_Y_Index,:) = [];
end

if size(Cross_Corr,1) > size(Reference_LPE_SCN_MATRIX,1)
    Cross_Corr(size(...
```

```
        Reference_LPE_SCN_MATRIX,1)+1:size(Cross_Corr,1),:) = [];
end

Target_SCN_MATRIX_2D_match_filt = Cross_Corr;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%% CREATE IMAGE OF TARGET_SCN_MATRIX_2D_MATCH_FILT %%%%%%%%%%%%%%%%%%%
figure
imagesc(Reference_Rbin,Reference_radar_position_column,20*log10(...
    abs(Target_SCN_MATRIX_2D_match_filt)));
%title('Target Scene (dB Scale)');
xlabel('x [meters]');
ylabel('y [meters]');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

BIBLIOGRAPHY

[1] L. Maslikowski, P. Samczynski, M. Baczyk, P. Krysik and K. Kulpa, "Passive bistatic SAR imaging — Challenges and limitations," in *IEEE Aerospace and Electronic Systems Magazine*, vol. 29, no. 7, pp. 23-29, July 2014.

[2] C. Ozdemir, *Inverse Synthetic Aperture Radar Imaging With MATLAB*. John Wiley & Sons, 2012.

[3] M. Soumekh, *Synthetic aperture radar signal processing with MATLAB algorithms*. New York: J. Wiley, 1999.

[4] I. Cumming and F. Wong, *Digital processing of synthetic aperture radar data*. Boston: Artech House, 2005.

[5] D. GABOR, "A New Microscopic Principle", *Nature*, vol. 161, no. 4098, pp. 777-778, 1948.

[6] 47 CFR Part 15 Subpart F, Revised as of October 1, 2015.

[7] C. Nguyen and J. Han, *Time-domain ultra-wideband radar, sensor and components*. New York: Springer, 2014.

[8] M. Mabrouk, S. Rajan, M. Bolic, I. Batkin, H. R. Dajani and V. Z. Groza, "Model of human breathing reflected signal received by PN-UWB radar," *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, Chicago, IL, 2014, pp. 4559-4562.

[9] T. Tian, "An Ultra-Wide Band Radar Based Noncontact Device for Real-time Apnea Detection." M.S. thesis, Dept. Elect. and Comp. Eng., ECE, Worcester Polytechnic Inst., Worcester, MA, 2015.

[10] W. Troy, M. Thompson and Yang Li, "ISAR imaging of rotating blades with an UWB radar," *Wireless and Microwave Circuits and Systems (WMCS), 2015 Texas Symposium on*, Waco, TX, 2015, pp. 1-4.

[11] Chenchen Li, Shang-Te Yang and Hao Ling, "ISAR imaging of a windmill – measurement and simulation," *Antennas and Propagation (EuCAP), 2014 8th European Conference on*, The Hague, 2014, pp. 1-5.

[12] C. J. Li and Hao Ling, "Synthetic aperture radar imaging using a small consumer drone," *Antennas and Propagation & USNC/URSI National Radio Science Meeting, 2015 IEEE International Symposium on*, Vancouver, BC, 2015, pp. 685-686.

[13] “Data Sheet PulsON 410,” [pdf]. March 2, 2016. Available: http://www.timedomain.com/datasheets/320-0289E%20P410%20Data%20Sheet.pdf

[14] Data from Matlab file P410_wo_Amp_Tmplt.m from within 150-0115B-CAT-Sample_MATLAB_CLEAN-Algorithm.zip Retrieved from http://www.timedomain.com/channel-analysis-bistatic-radar/

[15] “Monostatic Radar Module Reconfiguration and Evaluation Tool (MRM RET) User Guide PulsON 400 MRM 1.2,” [pdf]. March 2, 2016. Available: http://www.timedomain.com/datasheets/320-0300E%20MRM%20RET%20User%20Guide.pdf

[16] “Host Command Reference Q and SCL commands for servo and stepper drives,” [pdf]. March 2, 2016. Available: http://www.applied-motion.com/sites/default/files/hardware-manuals/Host%20Command%20Reference%20Rev%20I.pdf

[17] “Bandpass Signals Equivalent Lowpass Signals I&Q Signals,” [pdf]. Retrieved from M. Fowler website. March 2, 2016. Available: http://www.ws.binghamton.edu/fowler/fowler%20personal%20page/EE521_files/II-1%20BP,%20Eq%20LP,%20I&Q%20Signals_2007_rev.pdf