UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

TOOLBOX AND POST-PROCESSING APPLICATIONS FOR AUTOMTOIVE RADAR

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

EVAN TISDALE

Norman, Oklahoma

2021

Downloaded from SHAREOK. Persistent Handle URL: https://hdl.handle.net/11244/332335

TOOLBOX AND POST-PROCESSING APPLICATIONS FOR AUTOMTOIVE RADAR

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Justin Metcalf, Chair

Dr. Nathan Goodman

Dr. Jay McDaniel

© Copyright by EVAN TISDALE 2021

All Rights Reserved.

Acknowledgments

First, I would like to thank my Advisor, Dr. Justin Metcalf, for his support, patience, and time. I also want to thank my advisory committee, Dr. Nathan Goodman and Dr. Jay McDaniel, for making this thesis possible.

To all my friends and family who have supported me through the years of graduate school, I can never thank you enough.

And finally, a special thanks to my partner Ashley, who supported me through long hours of research, writing, and driving me around to collect data. Your love and support will always be appreciated.

Table of Contents

Acknowledgment **iv**

Table of Contents **v**

List of Tables **viii**

List of Figures **ix**

Abstract **xiv**

1 Introduction **1**

1.1 History of Automotive Radar 2

1.2 Automotive Sensing Today 5

1.3 Regulations with Automotive Radar 6

1.4 Motivation and Outline 8

2 Proposed Capture System **11**

2.1 AWR1642Boost Evaluation Module 11

2.2 DCA1000EVM Capture card 16

2.2.1 LVDS Data Transfer 18

2.3 Proposed System Implementation 20

2.3.1 Field Testing 21

2.3.2 3D Printed Mount System 25

2.3.3 Final Remarks 28

3 Software Tool Chain **30**

3.1 mmWaveStudio 30

3.1.1 Install and Operation 31

3.1.2 Connection 31

3.1.3 Static, Data, and Test Source 34

3.1.4 Sensor, Chirp, and Frame Configuration 35

3.1.5 mmWaveStudio Post-Processing 36

3.2 Video Capturing 37

3.3 Final Remarks 37

4 FMCW Data Processing **40**

4.1 Radar Date Cube 41

4.2 Storing Radar Data 42

4.3 FMCW Radar 46

4.3.1 Range Processing 47

4.3.2 Range Estimation 51

4.4 Range-Doppler Processing 54

4.4.1 Velocity Estimation 55

4.4.2 Velocity Aliasing 59

4.5 Angle Estimation using MIMO Radar 60

4.5.1 MIMO and FMCW Overview 60

4.6 Summary 63

5 Post-Processing GUI **64**

5.1 Radar Fusion Interface 64

5.1.1 System Time Syncing Background 65

5.1.2 Video Radar Frame Syncing Implementation 65

5.2 CFAR 67

6 Data Collection Results **69**

6.1 Data Capturing 69

6.1.1 Driving: Country Road and Fence Posts 70

6.1.2 Stationary: Person Walking Dog in Neighborhood 71

6.1.3 Stationary: Handicap Sign in Parking lot 72

6.1.4 Driving: Debris and Pothole in Parking Lot 73

6.1.5 Driving: Person Walking on Side Walk next to Street 74

6.1.6 Driving: Approaching Stop Sign in Neighborhood 75

6.1.7 Stationary: Waiting at Stoplight with Passing Vehicles . . . 76

6.1.8 Driving: Busy Parking Lot with Moving and Stationary cars 77

7 Conclusions and Future work **78**

7.1 Future Work 79

7.1.1 Proposed System Case Upgrades 79

7.1.2 Expanded Operations 79

7.1.3 mmWaveStudio Automation 79

References **81**

List of Tables

1.1 Table of the common uses of automotive radar[1] 6

2.1 SOP operating modes. The Debug Mode highlighted is the standard operating mode when using *mmWwaveStudio* 13

2.2 SOP operating modes. The Debug Mode highlighted is the standard operating mode when using *mmWwaveStudio* 13

2.3 Laptop specifications used in this thesis. 21

3.1 Modes for ADC data capture as documented by TI. 34

4.1 Computer specifications for all post processing 44

4.2 Summary of standard range-Doppler and angle estimation equations. 63

List of Figures

1.1 A copy of the United Kingdom patent to Christian Hülsmeyer for his device the Telemobiloskop. 2

1.2 A 10GHz Horn system was mounted on top of the car, an example of the first automotive radar in the 1970s. Picture from the private collection of Holger H. Meinel[2]. 3

1.3 Legacy and new Automotive radar bands[3]. 4

1.4 Examples of YOLO object detection routine being used across various images, videos, and paintings.[4] 8

1.5 A public domain SAR image from the Space Shuttle Endeavour showing the Teide volcano. 9

1.6 The proposed system ready to capture data as it is mounted on the windshield of a car. 10

2.1 A) Front end of components on AWR1642 board B) Close up of user configurable switch and jumper pins that control the operating modes of the board. This configuration is how the board was operated in the data capture processes. 12

2.2 Front end of components on AWR1642 board (a) and a closeup of its antenna (b) 14

2.3 Layout of system used to capture radar system. [5] 14

2.4 HFSS Simulation of antenna pattern provided by TI. Elevation plane is red. Horizontal plane is purple. 15

2.5 Front end of components on DCA1000EVM board 16
2.6 A closeup of the user configurable swtiches as used in this report. A) Switch used for mode selection and other functionality. B) LVDS Bit mode setting C) LEDs used to inform users on operations occurring on the board. The far right LED highlighted in red highlights transferring of data when flashing. 17
2.7 Jitter Performance for CAT5 Cables [6]. 19
2.8 AWR1642 setup as delivered. Lacking any case or protection. The setup is good for laboratory testing, but limits applications in the field. 20
2.9 The proposed system as it was used in this thesis. 22
2.10 Radar Mounted to tripod for stationary target acquisition. 23
2.11 Using a suction cup phone mount with 3D printed mount, the full setup can be mounted to capture data while driving. 24
2.12 Capturing data in car required running and powering laptop internally. 24
2.13 The STL file was transferred into *Fusion360* CAD software and modified so that the dovetail mount could be customized for any mount 26
2.14 Radar mount designed for this system after being printed on a privately owned Ender 3 with PLA. 27
2.15 A) DCA1000EVM board with broken FTDI Micro USB port. B) Micro USB Port with detached pads. C) Micro USB reinforced with Liquid Electrical Tape 29

3.1 MMWAVE studio connection showing connection to AWR1642 and DCA1000 is ready 32
3.2 Two manual configurations to run *mmWaveStudio*: A) Static Ethernet setting at 192.168.33.30 B) FTDI driver install seen in COM4 and COM5 33

3.3 Screen Verifying that the DCA1000 is ready to capture data. Note the static IP. This must be manually set on the Ethernet port 33
3.4 Example of the ADC data for each chirp as it is stored for a DCA1000 EVM with an xWR16xx complex, 4 channel recieve, and 2 LVDS lanes 35
3.5 Chirp Profile of TI FMCW radar courtesy of the MMWAVE studio SDK manual[7]. 35
3.6 Data seen after being captured in MMWAVE studio. 37
3.7 Data Capture Process. 39

4.1 A representation of received data on one channel in FMCW Radar. [8]. 41
4.2 Radar cube is used to describe captured data in multi channel pulsed radar [8]. A) Data slice used to get an Angle Estimation. Fourier transform taken across receivers produces angle information. B) Slice used to create Velocity Range plot. Fourier transform taken across pulses produces velocity information. 42
4.3 Representation of raw ADC data captured over time. 43
4.4 The three methods for capturing RAW ADC data using *mmWaveStudio* and DCA1000EVM. 45
4.5 Top Plot shows example of FMCW Signal simulated in MATLAB. Bottom plot shows an example of the frequency slope of signal. . . 47
4.6 Reflected signals from multiple detected objects and the IF tones created. 49
4.7 IF Tones showing how increased bandwidth can help resolve signals. ADC Sample 1: the bandwidth is too close and will show up as a single peak in the frequency spectrum. ADC Sample 2: A higher bandwidth allows the signals to be resolved in the Frequency spectrum 50

4.8 Example of slow-time chirps plotted as Distance in meters vs. normalized receive power (n dB). Peaks represent objects being detected at a certain distance. First two peaks are the direct path coupling and ground response (and therefore a measure of antenna height from the ground), respectively. 52
4.9 Process to calculate down-range plot. 53
4.10 Example of a typical velocity vs down-range plot. A) An object moving a different velocity B) Stationary objects end up in a single velocity bin corresponding to the speed of the observing radar. C) Clutter due to internal reflections from the car taking data. From this frame of reference it ends up at the zero Doppler position. . . . 54
4.11 1) Multiple Objects are moving with a certain velocity and are positioned in various down-range positions. 2) In every time sample the phase of each object across slow-time is stored 3) Applying the DFT resolved the phase difference that can be plotted 57
4.12 Process to calculate Velocity vs down-range plot 58
4.13 This figure shows aliasing in a range-Doppler plot. 59
4.14 A) Configuration of MIMO Antenna on the AWR1642. B) Virtual antenna positioning that can be interpreted when using two transmitters on AWR1642 C) Chirp profile showing how chirps are called to each transmitter when operating in multi transmit mode. . . 61
4.15 Concept of angle estimation demonstrated on AWR1642 antenna array[9]. θ defined in equation 4.16. 61
4.16 Implementation of angle estimation with passing cars at a stoplight. 63

5.1 The GUI allows a user to modify parameters(1a), view the calculated parameters (1b), modify the CFAR Window(1c), user inputted radar and video files (1d). Additionally plots can be generated and scaled independently for viewing (2) and (3). 66

5.2 Two-dimensional window used in CFAR processing on range-Doppler plots. 67
5.3 Top) Down-range velocity plot with no CFAR post-processing. Bottom) Down-range velocity plot with added CFAR algorithm. Noise is all but eliminated. 68

6.1 Early sample of radar taken inside vehicle on country road. Fence posts on both side of the road a "curved" sequence of returns in the range-Doppler. 70
6.2 A stationary shot of a person and dog walking on the sidewalk. . . . 71
6.3 A stationary shot of a Handicap sign in parking lot. Notice the high RCS return of the sign. 72
6.4 Moving radar capture of pothole and debris in a parking lot. Notice the gap after the debris. The next detection is the tree and curb across the street. 73
6.5 View of a neighborhood road. 74
6.6 View of car approaching stop sign. 75
6.7 Scene of radar capture at a traffic light with passing cars. 76
6.8 A full parking lot with a large pot hole. 77

Abstract

In recent years, companies have pushed the limits to offer partial and fully autonomous driving systems. All vehicles now have a wide array of sensors to aid in driving, ranging from lane assist to automated parallel parking, which culminates into a hybrid system on the verge of complete autonomy. However, Formal adoption by consumers and regulatory agencies still faces justified scrutiny. Self-driving cars will need to drive better and perceive beyond the capabilities of the average driver before being allowed on the road. The ability to capture data for research and algorithm development is necessary. A cheap and reliable automotive radar system can help expand the field and drive the future of automated systems.

This thesis describes the development of a system toolbox using a 77 GHz frequency-modulated continuous-wave (FMCW) radar and video camera for data collection and post-processing applications. System details for the hardware, software, and overview of a custom-made 3D printable mount for vehicles are covered. Hardware specifics dive into using a Texas Instrument's AWR1642Boost system and DCA1000EVM FPGA capture card with a collocated HD video camera. In addition, the thesis covers the software toolchain used to collect data and prepare it for storage and post-processing. Inside is a detailed explanation of radar signal processing theories and applications with Matlab, including range-Doppler and angle-of-arrival estimation. Additional post-processing topics include radar-video fusion frame synchronization and constant false alarm rate (CFAR) applications in range-

Doppler plots. Results generated in this thesis focus on the setup process of the proposed system and raw analog-to-digital converter (ADC) data with recorded HD video from stationery and vehicle-based measurements. The data includes scenes in neighborhoods, city streets, parking lots, and country roads. Objects included in the dataset are pedestrians, dogs, vehicles, road signs, and potholes. Results include a promising, cheap millimeter-wave automotive setup that has the potential to become a fully-developed radar video fusion system.

Chapter 1

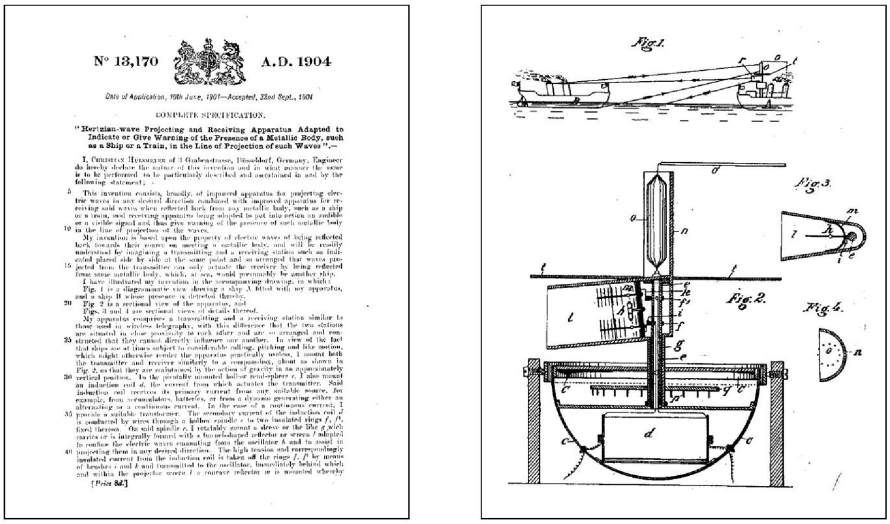
Introduction

Self-driving cars have always been a staple of science fiction. *Total Recall*, *Demolition Man*, and *Minority Report* offer distinct visions of autonomous vehicles in the not too distant future. In recent years, companies, including Uber and Tesla, have pushed those limits to offer partial and fully autonomous driving systems, though it will still be some time before the roads are full of robotic drivers. All vehicles now have a wide array of sensors to aid in driving, ranging from lane assist to automated parallel parking, which culminates into a hybrid system on the verge of complete autonomy. However, formal adoption by consumers and regulatory agencies still faces justified scrutiny. Self-driving cars will need to drive better and perceive beyond the capabilities of the average driver before being allowed on the road.

Vehicles have many sensors at their control; however, a combined system that hybridizes them is the next step in automotive sensing. A golden age of high computational embedded systems and next-generation millimeter-wave radar has opened up cheap sensing capabilities in vehicles. Using a fusion of sensors has become the path for more accurate classification systems needed in future AI. The ability to perceive information through conditions that an average human cannot perceive will revolutionize driving for all.

1.1 History of Automotive Radar

Although widely known for its military applications, radar's origins were actually in safety and collision avoidance. Christian Hülsmeyer, a German scientist, first demonstrated using electromagnetic energy to detect objects on May 17, 1904. Hülsmeyer made history using his device to detect a large barge hidden in fog passing on the Rhine River[10].



Nº 13,170 A.D. 1904

Date of Application, 10th June, 1904—Accepted, 23rd Sept., 1904

COMPLETE SPECIFICATION.

"Herzian-wave Projecting and Receiving Apparatus Adapted to Indicate or Give Warning of the Presence of a Metallic Body, such as a Ship or a Train, in the Line of Projection of such Waves".—

I, CHRISTIAN HÜLSMEYER of 3 Grabenstrasse, Düsseldorf, Germany, Engineer do hereby declare the nature of this invention and in what manner the same is to be performed to be particularly described and ascertained in and by the following statement:—

This invention consists, broadly, of improved apparatus for projecting electric waves in any desired direction combined with improved apparatus for receiving said waves when reflected back from any metallic body, such as a ship or a train, said receiving apparatus being adapted to put into action an audible or a visible signal and thus give warning of the presence of such metallic body in the line of projection of the waves.

My invention is based upon the property of electric waves of being reflected back towards their source on meeting a metallic body, and will be readily understood for imagining a transmitting and a receiving station such as hereinafter placed side by side at the same point and so arranged that waves projected from the transmitter can only actuate the receiver by being reflected from some metallic body, which, in any, would presumably be another ship.

I have illustrated my invention in the accompanying drawing, to which:—

Fig. 1 is a diagrammatic view showing a ship A fitted with my apparatus, and a ship B whose presence is detected thereby.

Fig. 2 is a sectional view of the apparatus, and

Figs. 3 and 4 are sectional views of details thereof.

My apparatus comprises a transmitting and a receiving station similar to those used in wireless telegraphy, with this difference that the two stations are situated in close proximity to each other and are so arranged and constructed that they cannot directly influence one another. In view of the fact that ships are at times subject to considerable rolling, pitching and like motion, which might otherwise render the apparatus practically useless, I mount both the transmitter and receiver similarly to a compass-bowl, about as shown in Fig. 2, so that they are maintained by the action of gravity in an approximately vertical position. To the pivotally mounted bell or recul-plate c, I also mount an induction coil d, the current from which actuates the transmitter. Said induction coil receives its primary current from any suitable source, for example, from accumulators, batteries, or from a dynamo generating either an alternating or a continuous current. In the case of a continuous current, I provide a suitable transformer. The secondary current of the induction coil d is conducted by wires through a leather spindle e to two insulated rings f, f1, fixed thereon. On said spindle e, I suitably mount a sleeve or the like g, which carries or is integrally formed with a frusto-shaped reflector or screen l adapted to confine the electric waves emanating from the oscillator k and to assist in projecting them in any desired direction. The high tension and correspondingly insulated current from the induction coil is taken off the rings f, f1 by means of brushes i and i1 and transmitted to the oscillator, immediately behind which and within the projector screen l a contrary reflector m is mounted whereby

[Price 8d.]

Figure 1.1: A copy of the United Kingdom patent to Christian Hülsmeyer for his device the Telemobiloskop.

Unfortunately for Hülsmeyer, his device did not hit mainstream success. Its true potential was realized decades later at the start of WWII. Ironically in the 1960s and 1970s, the idea of driver safety in automobiles became a new hot topic a couple of years after Hülsmeyer's death. Companies supported by the United States Department of Transportation and the Japanese companies Mitsubishi and Nissan began programs to research adding radar to automobiles. The German companies SEL, VDO, and AEG Telefunken began their research sponsored by the German Ministry of Science and Technology with a focus on pre-collision avoidance and

emergency braking [2][11]. Companies developing automotive radar utilized components from telecommunications and military devices already widely available and in frequencies: 35, 47, 60, and 94 GHz[2]. Germany would be the first to start using the 77GHz frequencies in the early 1980s for use in long-range automotive radar (LRR)[2].



Figure 1.2: A 10GHz Horn system was mounted on top of the car, an example of the first automotive radar in the 1970s. Picture from the private collection of Holger H. Meinel[2].

The United States mounted the first commercial 24 GHz automotive radar from a small start-up company Vorad Safety Systems Inc[12]. The radar was attached to buses and trucks and proved to reduce accidents considerably[2][11]. Pushback from drivers unions stopped the use of the Vorad radars due to drivers feeling the system had too much control, a sign that people were not yet ready to trust systems controlling their actions in cars[2][11]. Mercedes Benz utilized the first commercial 76GHz automotive radar for cars in 1998/99[11]. The German company Con-

tinental would build more advanced versions of this system using more advanced scanning systems. This system utilized a rotating drum with ridges to polarize energy, providing varying distances with periodic discontinuities giving radiation at varying angles in the horizontal direction[11].

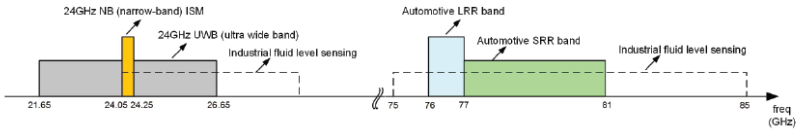


Figure 1.3: Legacy and new Automotive radar bands[3].

Today, radars utilize phased array technology with multi-input multi-output (MIMO) antenna arrays. Texas Instruments (TI) has developed a new series of highly integrated radar sensors for commercial use in the 77 GHz frequency band[3]. These sensors utilize the small form factor that comes with operating at the newly defined automotive band, offering many options for sensors, including various antenna configurations for testing.

The modern car has many radar sensors. Simple things such as Blind-spot detection to Adaptive Cruise Control are standard features in most vehicles now. Existing automotive applications use a mix of Short Range Radar (SRR) and Long Range Radar (LRR). These applications utilize their given frequency and bandwidth to balance the information they can capture to assist in driving. This adoption of sensors has opened the market for cheap automotive sensors. Researchers at all levels are now pushing the limits beyond basic safety features.

1.2 Automotive Sensing Today

Utilizing a wide range of sensors available in vehicles has opened up new possibilities for safety and control. These sensors do not only include radar-based sensors. HD cameras, LIDAR, thermal vision, and ultrasonics all offer distinct sensing capabilities for high-end detection and object identification on the road today[13]. All of these sensors offer varying information, with strengths and weaknesses.

Sensors operating in the optical light spectrum are becoming the most common. Previously, video processing speed was an issue, but recent advancements in graphics processor units (GPUS) and machine learning have made real-time object classification at high speeds possible. You Only Look Once (YOLO) is an open-source real-time object detection that has become a favorite in automotive detection research groups due to its speed and efficiency[4]. Even as machine learning advances, there are still shortcomings in distinguishing range and identifying untrained objects. Optical sensors that can operate beyond an average person include thermal, night vision, and LIDAR. These systems can be cost-prohibitive and prone to fail in low visibility conditions such as snow, fog, and dust.

Automotive radar sensors operate in various configurations for specified applications. Standard features using radar include lane-change assist, blind-spot detection, and adaptive cruise control. These features are cheap and effective for auto manufacturers to incorporate into existing systems. Table 1.1 offers a summary of common automotive radar applications.

Tesla Motors offered to incorporate more advanced radar-video fusion into their autonomous systems. However, earlier this year, Tesla announced it was removing radar from its suite of sensors. Saying it will rely entirely on its "Tesla Vision" for all object detection. Since this announcement, the partial driving assistance has

had reported issues with misclassifying the moon and advertisement signs as yellow traffic lights causing cars to warn and slow down the driver[14].

Common Automotive Radar Applications			
Name	**Type**	**Range (m)**	**FOV**
Blind-Spot Detection (BSD)	USRR SRR	40 80	150°
Lane-Change Assist	USRR SRR	40 80	150°
Front/Rear Cross Traffic Alert (F/RCTA)	MRR LRR	80-120 >200	60° 15°
Adaptive Cruise Control (ACC)	MRR LRR	80-120 >200	60° 15°
Autonomous Emergency Braking (AEB)	MRR LRR	80-120 >200	60° 15°

Table 1.1: Table of the common uses of automotive radar[1]

.

1.3 Regulations with Automotive Radar

Regulations involved in self-driving cars and sensors have become an operational challenge for governments across the world. The industrial, scientific, and medical (ISM) band from 24.0 to 24.25 GHz housed and unlicensed ultra-wideband (UWB), centered at 24 GHz with a 5 GHz wide bandwidth, was home for many automotive radar sensors. An agreement between the Federal Communications Commission (FCC) and the European Telecommunications Standards Institute (ETSI) will phase out the use of the 24 GHz UWB band after January 1, 2022[3]. This newly defined band allows vehicular radar operations mounted in terrestrial trans-

portation vehicles[15]. The FCC defines this as:

> Vehicles include automobiles, trucks, railroad train locomotives, train cars, monorails or trams, construction vehicles, farming vehicles such as tractors and harvesters, motorcycles, scooters and motorbikes, mobile scissor-lifts and mobile work platforms, and boats and ships operating within territorial waters of the United States[15].

SAE International, an organization focused on industry standardization, has already begun creating standards and rules for self-driving technology and definitions. These definitions are in six categories that break down regular human operation and automated driving features[16]. As the technology matures, there will be increased oversight, which may classify legal operations of self-driving cars using such definitions. A few years ago, a pedestrian was struck by an automated Uber vehicle in Arizona [17]. While rare, these accidents drive down consumer confidence and lower technology development out of fear of safety. In the future redundant sensing systems will very likely be mandatory.

1.4 Motivation and Outline

As mentioned previously, machine learning has opened the doors to real-time object identification. Increased processing power and large amounts of openly available image data have directly contributed to the successful training of systems like YOLO. Video naturally lends itself to future post-processing as verification of classification is simple for the end-user. Figure 1.4 shows examples of classification using the YOLO routine. The simple boxes with a classification caption make it easy for an end-user to verify results.

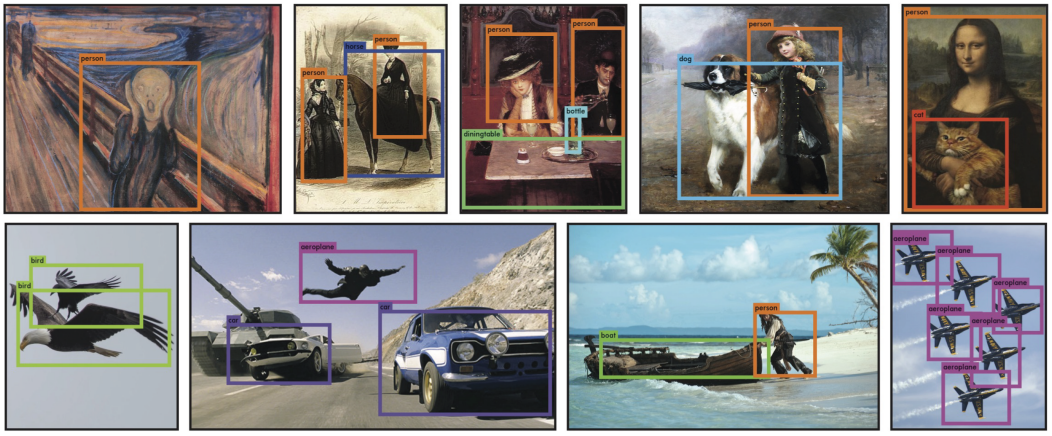


Figure 1.4: Examples of YOLO object detection routine being used across various images, videos, and paintings.[4]

Radar imaging does not produce easily recognizable data for object classification. Exceptional cases using synthetic-aperture radar (SAR) allow two or three-dimensional reconstructions of objects that are easily recognizable, as seen in Figure 1.5. However, processing and setup for such images are not available for automotive radar applications, which are limited to two-dimensional plots from a gods eye view.

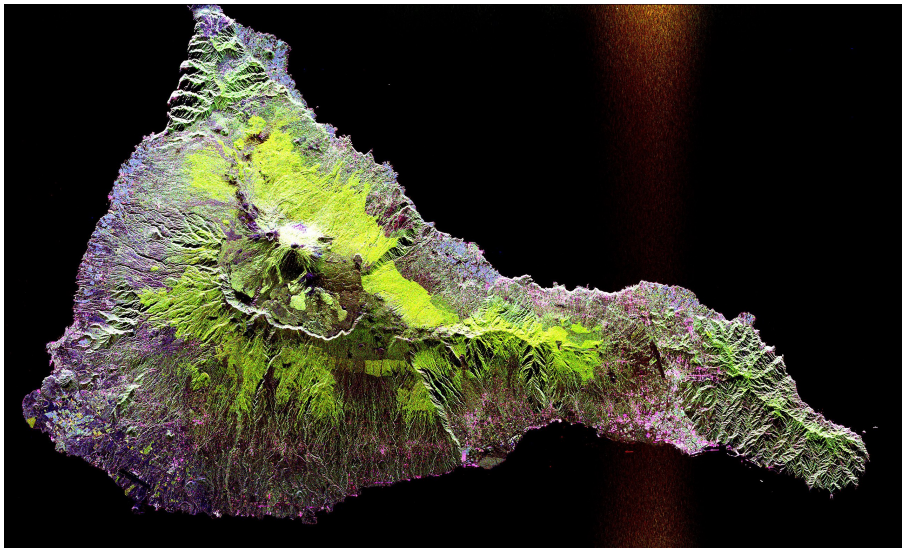


Figure 1.5: A public domain SAR image from the Space Shuttle Endeavour showing the Teide volcano.

A desire for a robust system that can actively identify targets in real-time with automotive radar has become a new field of interest. Currently, sensors are limited to specialized millimeter-wave radars utilizing limited object detection capabilities. New specialized techniques using micro-Doppler radar are promising but are still in experimental infancy [18]. New opportunities in millimeter-wave radar research are opening up but require the ability to measure actual data. The ability to capture data in moving cars paves the way for robust analysis.

This thesis proposes a hardware and software toolchain using an inexpensive commercial of the shelf evaluation radar board and high definition webcam to aid research, seen in Figure 1.6. The combined system allows for quick and easy data capture for post-processing at a later date. Applications directed at millimeter-wave radar algorithm development as well as applications into sensor fusion are the goal. In this thesis, the chapters walk through the setup of a Texas Instruments (TI) AWR1642Boost Revision B evaluation board [19], a DCA1000 EVM (evaluation module) Capture Board[20], and a High Definition web camera. Further, a detailed

explanation for gathering raw ADC data and video is given. Additional background information on typical automotive radar signal processing and the use of a custom-developed graphic user interface are also covered. Finally, results from this system are displayed, along with conclusions of this system and future work to expand the proposed toolchain.



Figure 1.6: The proposed system ready to capture data as it is mounted on the windshield of a car.

Chapter 2

Proposed Capture System

This chapter covers the hardware setup in the proposed system. Detailed descriptions of the TI AWR1642EVM radar module and DCA1000EVM capture are within. Further detail of a custom 3D printed mounting system for the radar and capture board is shown.

2.1 AWR1642Boost Evaluation Module

This section discusses the TI AWR1642BOOST sensor evaluation module (EVM). This microcontroller unit (MCU) belongs to the "mmWave" series of 77 GHz automotive-focused radars and support boards produced by TI. The evaluation boards are a demo piece for TI's automotive radar chips, widely available for custom radar board design. This demo board is a part of a myriad of support software to verify and test the various radar configurations.

The evaluation boards can perform embedded digital signal processing and then transmit results to a graphic user interface (GUI) on a PC, allowing customers to test code interfacing with the chipset and the radar array performance characteristics. Advanced signal processing is unavailable using just the AWR1642 board. Raw data from the board for demo applications is not available without additional capture

tools. For more advanced applications, TI sells an FPGA capture card covered later in this section.

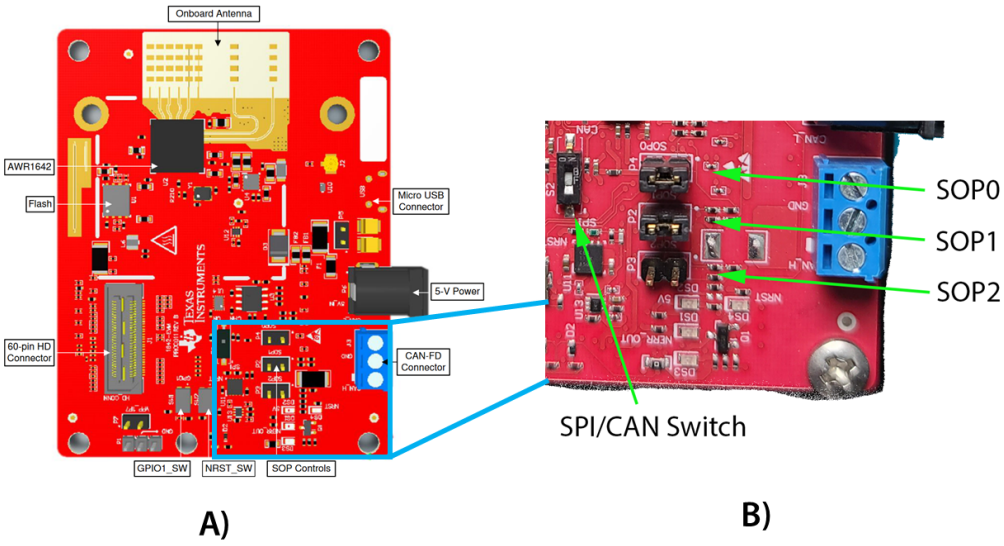


Figure 2.1: A) Front end of components on AWR1642 board B) Close up of user configurable switch and jumper pins that control the operating modes of the board. This configuration is how the board was operated in the data capture processes.

The AWR1642EVM has an onboard 76-GHz to 81-GHz frequency modulated continuous-wave (FMCW) automotive radar sensor with four receive (RX) and two transmit (TX) Antennas. The board is powered with a five-volt DC adapter and controlled via a micro USB Connector that allows it to interface with a PC. In Figure 2.1 The AWR1642EVM device has three operating modes controlled by sense-on-power (SOP) jumpers. Table 2.2 describes the operating modes of the SOP jumpers. As of Revision B, the AWR1642EVM has a switch to control the Serial Peripheral Interface (SPI) and Controller Area Network (CAN) bus. The multiplexed data channels belong on the same bus starting in revision B. This change requires the end-user to toggle the switch in the correct direction operating mode. In this thesis, the SPI bus was used exclusively, and the recreation of tests herein must follow

suit. Labels are included on the board to determine the mode, with more details explained in the TI datasheet [21].

Pin	SOP	Mode	Description
P4	0	011	Debug Mode
P2	1	001	Function Mode
P3	2	101	Flash Programming

Table 2.1: SOP operating modes. The Debug Mode highlighted is the standard operating mode when using *mmWwaveStudio*

Pin	SOP	Mode	Description
P4	0	011	Debug Mode
P2	1	001	Function Mode
P3	2	101	Flash Programming

Table 2.2: SOP operating modes. The Debug Mode highlighted is the standard operating mode when using *mmWwaveStudio*

The onboard antenna is laid out to create a virtual antenna array of eight receive elements utilizing multi-input-multi-output (MIMO) concepts. The receive antennas have two dummy elements that are grounded to reduce gain and phase mismatch between the four receive elements, as seen in Figure 2.2. An HFSS simulation of the antenna pattern provided by TI is shown in Figure 2.4. Utilization of the MIMO antenna and its effects on angle estimation are covered in detail in Chapter 5.

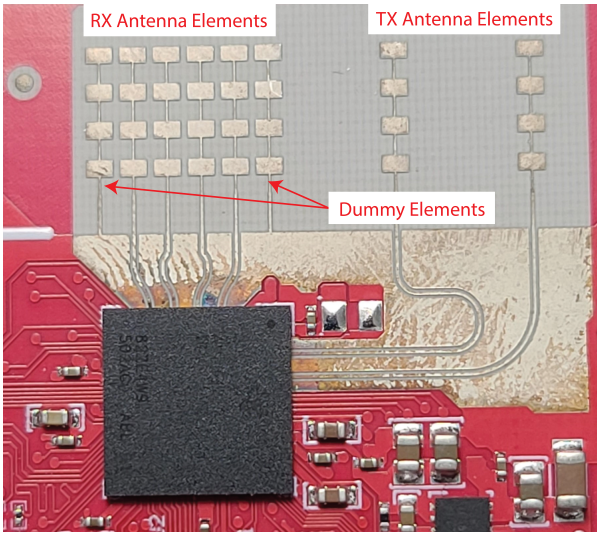


Figure 2.2: Front end of components on AWR1642 board (a) and a closeup of its antenna (b)

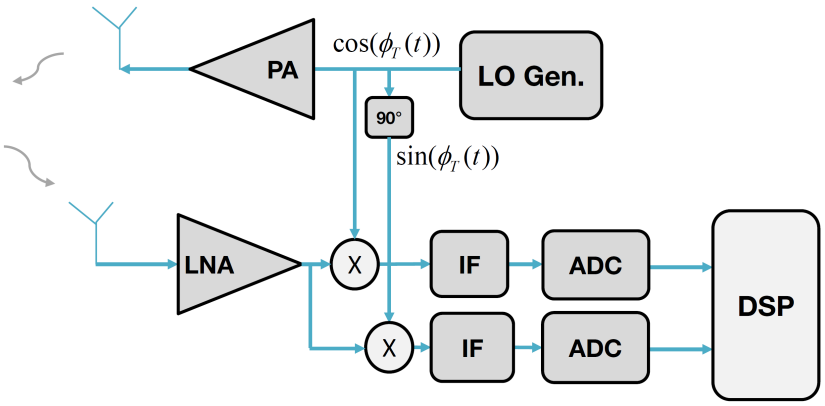


Figure 2.3: Layout of system used to capture radar system. [5]

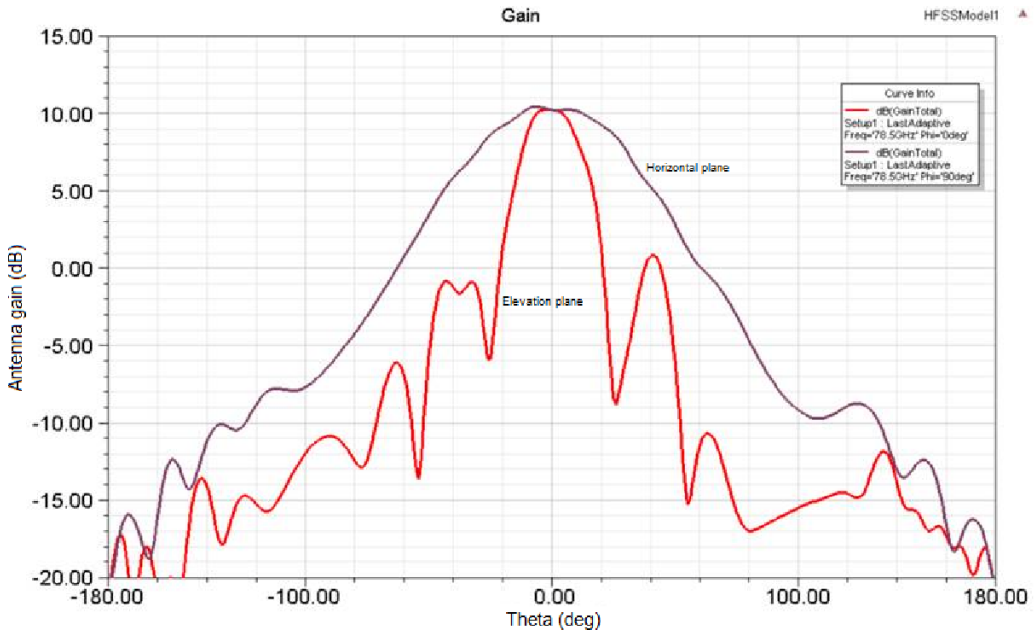


Figure 2.4: HFSS Simulation of antenna pattern provided by TI. Elevation plane is red. Horizontal plane is purple.

2.2 DCA1000EVM Capture card

The DCA1000EVM is a shield board capture card made to interface with Texas Instrument's line of 77GHz xWRxxxx EVM boards. This design utilizes a Lattice FPGA LFE5UM-85F-8BG381I with DDR3L memory to stream ADC data over Ethernet. Interfacing between both boards is handled by a 60-pin Samtec high-density connector. This connector handles the transfer of data as it is captured on the xWR board and provides the necessary five volts DC power to the DCA1000EVM.

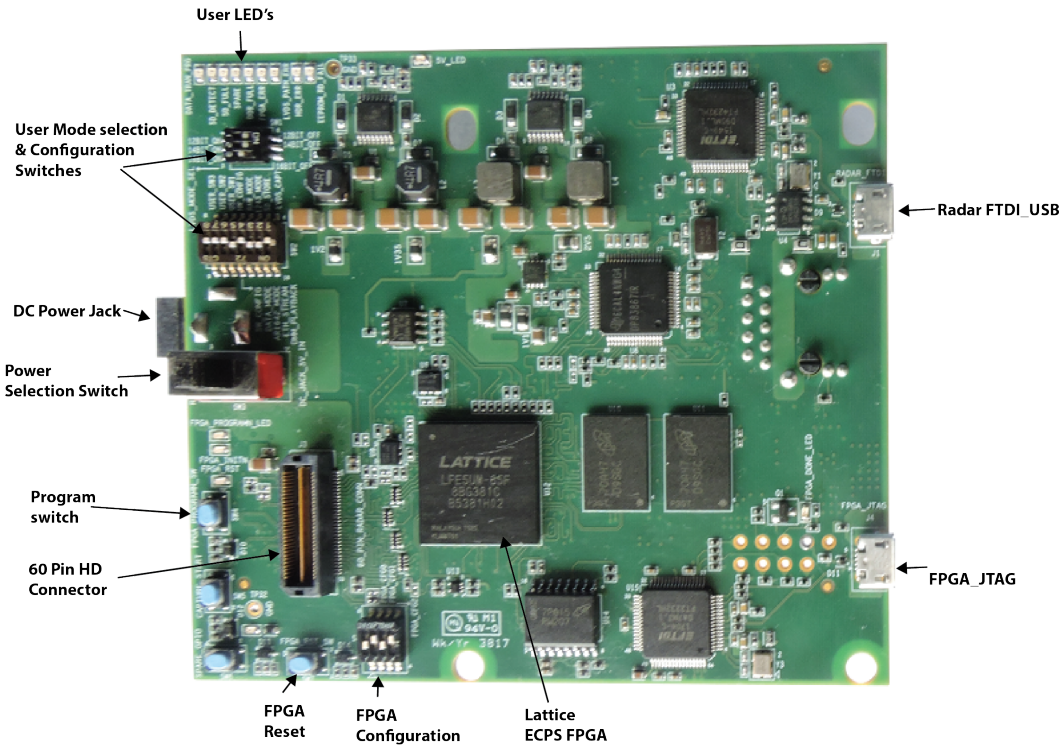


Figure 2.5: Front end of components on DCA1000EVM board

Operations of the DCA1000EVM are mostly handled using the *mmWwaveStudio* software but also require manual switch configurations in order to operate properly. More detail into the user-configurable switches and their operations are seen in Figure 2.6.

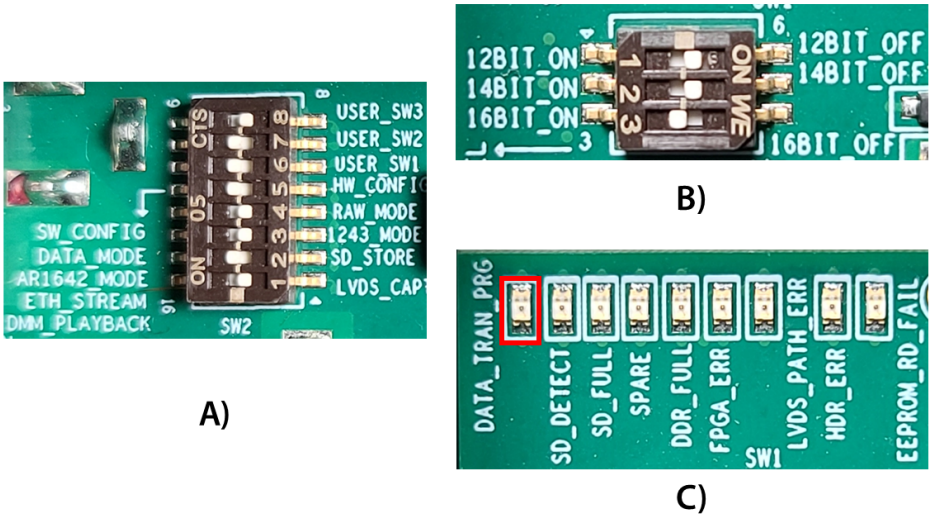


Figure 2.6: A closeup of the user configurable swtiches as used in this report. A) Switch used for mode selection and other functionality. B) LVDS Bit mode setting C) LEDs used to inform users on operations occurring on the board. The far right LED highlighted in red highlights transferring of data when flashing.

Configurations for the DCA1000EVM are concerned with data transfer from the ADC as it is streamed over Ethernet. This is broken up into two modes [20]:

1. "Raw Mode" allows all low-voltage differential signaling (LVDS) data captured sent by default headers specified in the user guide.

2. "Data Separated Mode" allows the user to add specific headers to different data types.

In this thesis, only "Raw Mode" was utilized. Recording of the data is handled by a command-line interface (CLI) executable program called by *mmWwaveStudio*.

Open-source efforts to use this executable without using *mmWwaveStudio* were found online but were not pursued in this thesis. Future efforts could be implemented if a complete departure from *mmWwaveStudio* is desired.

2.2.1 LVDS Data Transfer

LVDS, standardized TIA/EIA-644 [22], is a technical standard specifying characteristics of differential and serial signals standards. It is not a communication layer, acting only as a physical layer using other protocols to transfer data. LVDS offers low signal amplitudes to reduce power on the line circuits, and balanced signaling reduces noise coupling for higher signal rates [6]. The DCA1000EVM supports data rates at a maximum of 600 Mbps and two lanes when used with the xWR1642 board. In this study, CAT5 UTP cable was used in measurements. As with any transmission line, jitter is a factor that could easily affect results by end-users if they are not careful[1]. Future efforts that utilize this system will need to be aware of cable lengths. Mounting of the proposed systems in vehicles could easily use cable lengths greater than 10 meters as seen in Figure 2.7. An internal TI report classifying CAT cable performance characteristics was found and should be referenced [6].

[1] Jitter is the deviation from the true periodicity of a periodic signal

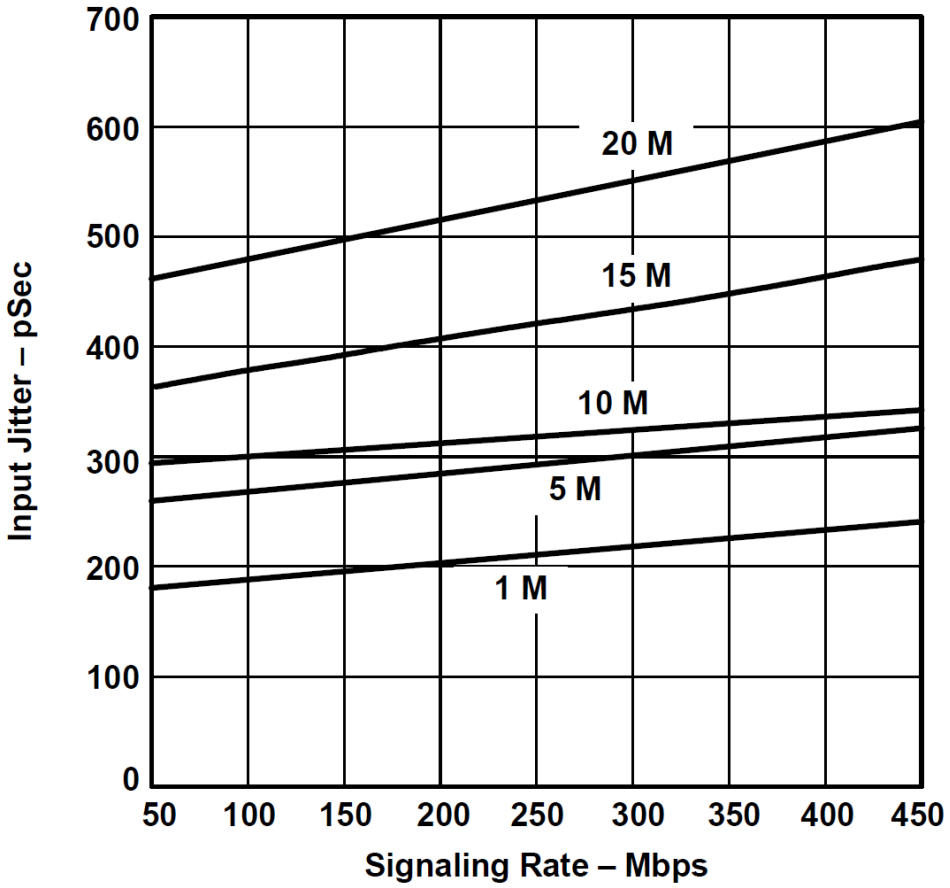


Figure 2.7: Jitter Performance for CAT5 Cables [6].

2.3 Proposed System Implementation

The xWR series of evaluation boards was never meant to be a fully realized data acquisition sensor. The low price of the system with its numerous features has made the board a fantastic budget board for millimeter-wave radar sensing with many supporting tools and a growing user community. From TI, the board comes with small L-brackets that allow desktop testing inside a lab. Online you can find many examples of people creating some cases to hold the system and allow it to be mounted for testing. In this section, the development of a hybrid case and mount is discussed, challenges faced during development, and the overall final setup.

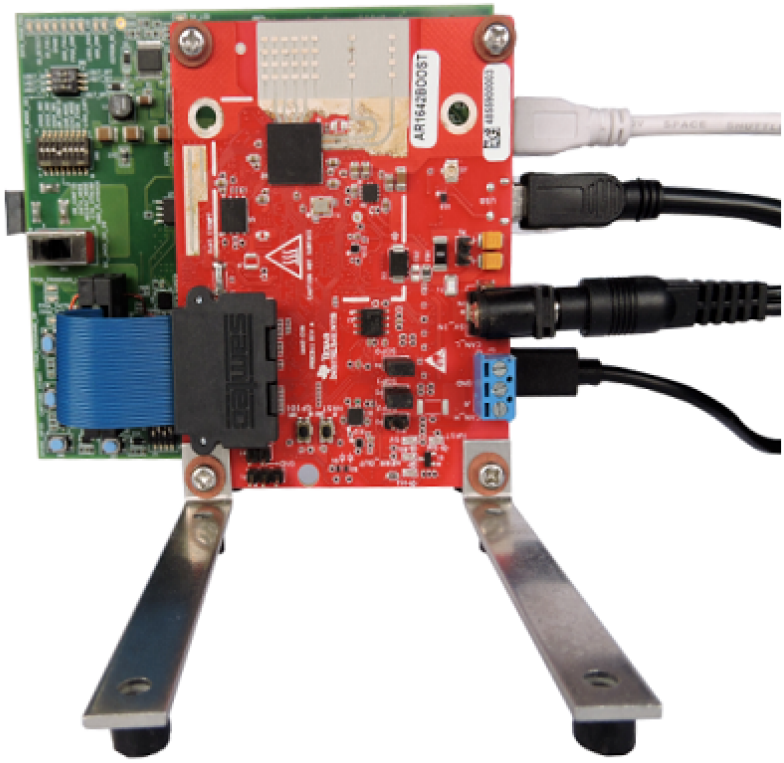


Figure 2.8: AWR1642 setup as delivered. Lacking any case or protection. The setup is good for laboratory testing, but limits applications in the field.

2.3.1 Field Testing

The system utilizes the AWR1642 EVM stacked on top of the DCA1000, and a USB web camera mounted on top. The two boards are connected via a ribbon cable that supplies power to the DCA1000EVM board. The system requires running three USB cables, an Ethernet cable, and a single 5 volt DC power jack. Data acquisition was made using a laptop with specifications in Table 2.3. To power the system in outdoor environments, an EGO brand electric battery with an inverter was used to power the system.

System	Specifications
CPU	Intel® Core™ I7-7000HQ CPU operating at 2.8 GHz
RAM	8 GB DDR4 RAM
GPU	NVIDIA GeForce GTX 1050
Storage	1 TB Physical Storage
Periphials	3 USB Ports 1 Ethernet Port

Table 2.3: Laptop specifications used in this thesis.

A 3D printed case fitted to any standard camera stand with a dovetail mount was custom-designed for this thesis. The entire case comes out to be around 100 mm wide and 110 mm tall. A small slit in the top of the mount allows a web camera mounted with a screw. The first fieldable mount used a standard camera tripod with a quick-release dovetail mount, allowing for stationary testing of moving objects. The second operation mounted the system to a vehicle that could be safely secured and used while driving a vehicle. A strong suction cup camera mount purposed to grab onto car bodies was also equipped with the system. Figure 1.6 shows an early prototype of the case with this mount.

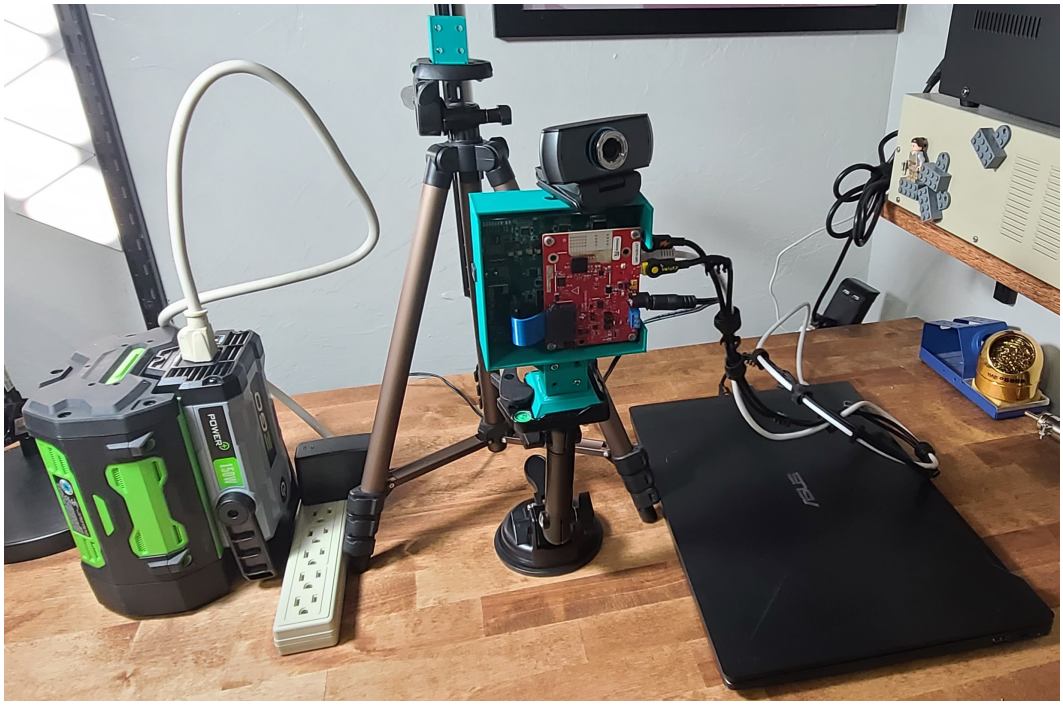


Figure 2.9: The proposed system as it was used in this thesis.

Being mounted internally caused interior features inside the cabin to be detected. Mounting the system on top and running the cabling through the sunroof proved effective. A laptop running *mmWwaveStudio* was connected to the system allowing an operator to record the video and radar data. See Chapter 3 for a detailed discussion of this synchronization procedure. Attempts at full automation of this process are feasible, but the laptop used in this study was not capable. *mmWwaveStudio* which is required to control the radar, is a very resource-intensive program. Attempts at managing a webcam tool with Matlab while simultaneously capturing radar data resulted in frequent errors as both programs competed for computational power on the system. For future real-time operations a high-end laptop with additional hard-disk storage and processing power will be needed.



Figure 2.10: Radar Mounted to tripod for stationary target acquisition.



Figure 2.11: Using a suction cup phone mount with 3D printed mount, the full setup can be mounted to capture data while driving.



Figure 2.12: Capturing data in car required running and powering laptop internally.

2.3.2 3D Printed Mount System

A case and mount system were made using an open-source Standard Triangle Language (STL) file found on the popular 3D printing website Thingiverse.com [23]. The case used a modified quick-release dovetail mount standard for quick-release systems found in commercial camera tripods. Figure 2.14 shows the use of the four small screws made for mounting the AWR1642EVM and DCA1000EVM board by using plastic standoffs to hold the board in place. The original model was ready to be printed for a set dovetail mount used by the original designer. Therefore, adjustments were made and printed for each unique stand used in this thesis.

Continuous printing of the cases and adapters became a nuisance. A removable dovetail mount, separate from the case, needed to be created to speed up testing and lower the waste of materials. This new capability developed in *Fusion360* features a removable dovetail mount that uses four M3 25mm machine screws to hold the case to the mount. Figure 2.13 shows a model of the radar case with its dovetail adapter. Now the case could be detached and swapped between mounts quickly. The board was printed on an Ender 3 V2 board with versions using PLA and PETG plastic. With the proposed adapter system, new modified versions can be generated in under an hour of printing. All CAD and STL files will be made available for public use.

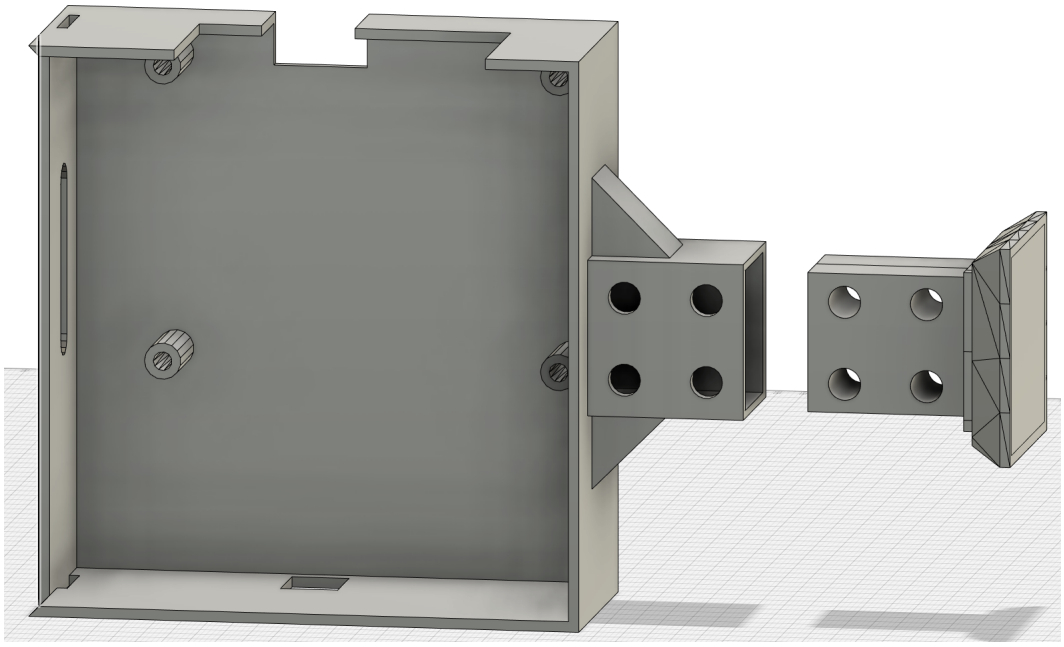


Figure 2.13: The STL file was transferred into *Fusion360* CAD software and modified so that the dovetail mount could be customized for any mount

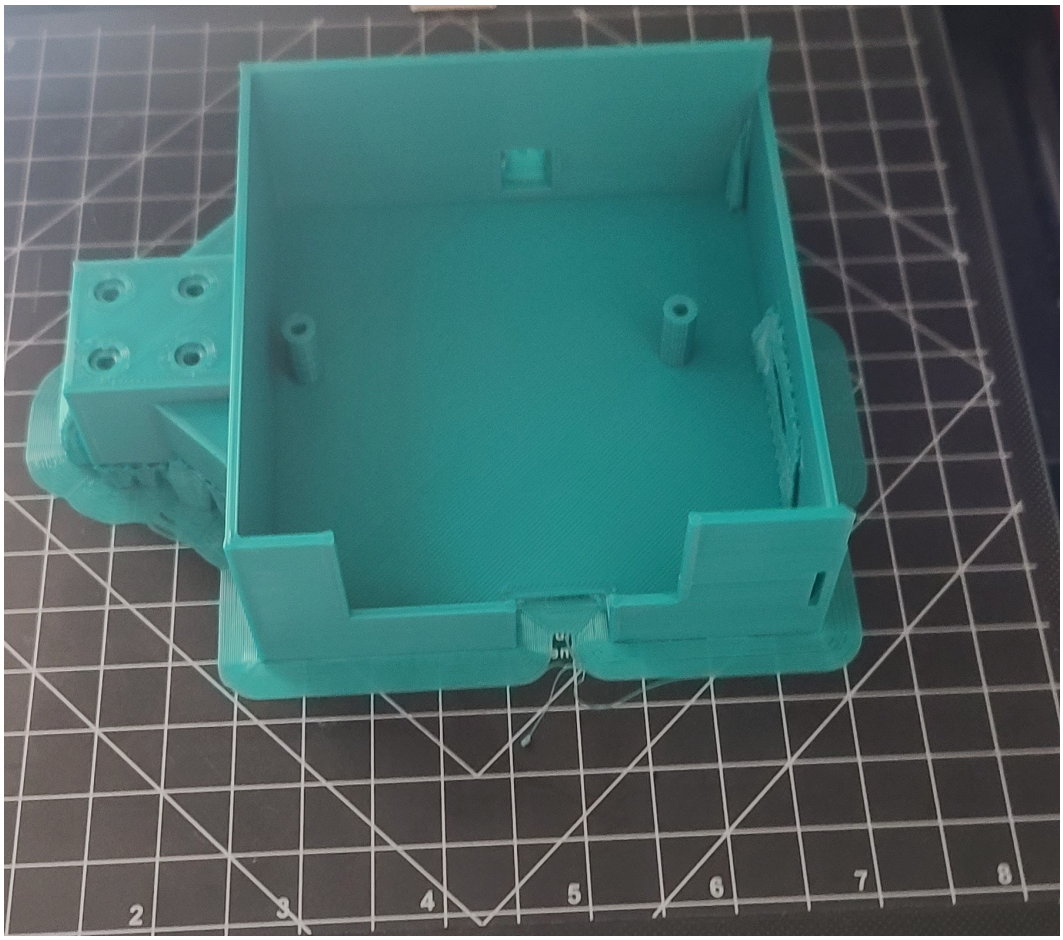


Figure 2.14: Radar mount designed for this system after being printed on a privately owned Ender 3 with PLA.

2.3.3 Final Remarks

The proposed system was successful in raw ADC data capturing in both indoor and outdoor environments. Further detail is provided in Chapter 6 which covers the data and environments captured. However, the TI AWR1642EVM and DCA1000 are not fieldable systems out of the box. The purpose of the TI radar device described herein was intended for evaluation purposes only. Users should exercise great caution to protect these devices when operating in an outdoor environment.

Sustained testing with this system eventually broke one of the DCA1000EVM micro USB ports off, as seen in 2.15. Although a minor component, this single USB port prevents controlling the device via *mmWaveStudio*. Consequently, the loss of this port effectively bricks the device. This damage occurred due to small amounts of solder holding the components to these evaluation boards. With enough torque from the attached cabling and vibrations from the operation, damage to these systems will be inevitable. Attempts at repair had limited success. A professional soldering lab under a microscope was able to re-solder a replacement micro-USB port. However, most structural support comes from small copper pads used to ground the component. These came off the board with the old micro-USB adapter and prevented a strong bond with the replacement component. The repaired device was reinforced with liquid electrical tape but it was not enough to maintain contact and the connector eventually broke again[2].

[2]Threads on TI's website show numerous cases with similar failure modes. At a price tag of over $1000, what starts as a budget setup could quickly become a costly investment. Future modifications to the case used in this report are highly suggested, as well as delicate operations by its users.

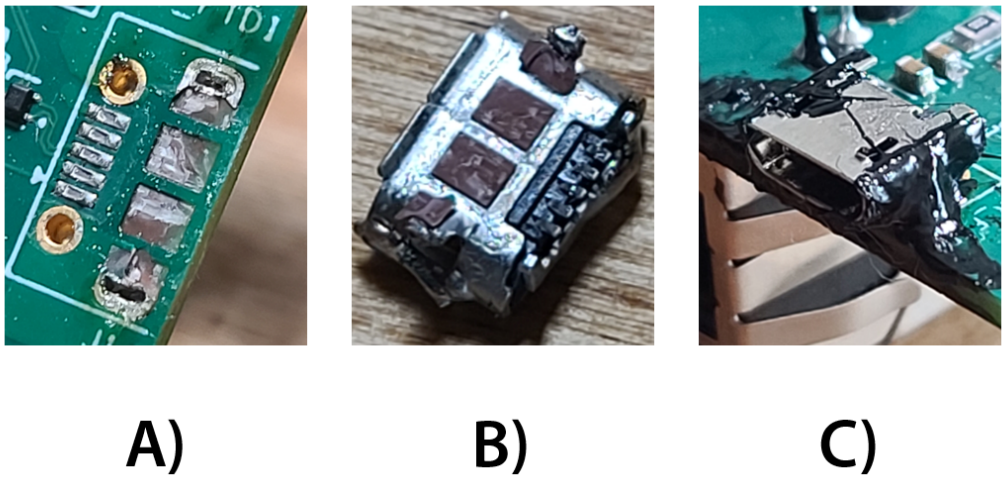


Figure 2.15: A) DCA1000EVM board with broken FTDI Micro USB port. B) Micro USB Port with detached pads. C) Micro USB reinforced with Liquid Electrical Tape

Chapter 3

Software Tool Chain

As previously mentioned, the radar evaluation boards made by Texas Instruments were made to be a demonstration piece for their automotive chipset rather than an off-the-shelf experimental data capture system. Utilizing a combination of provided software and tools, the AWR1642EVM board proved a reliable data acquisition tool. This chapter will detail the processes of using mmWaveStudio and scripts to use the proposed system as a data acquisition tool.

3.1 mmWaveStudio

The *mmWaveStudio* guide describes itself as a graphical user interface (GUI) designed to characterize and evaluate the TI Radar devices. The program only runs on Windows-based computers and uses Matlab compiler 8.5.1 to run its GUI. Installation of FTDI USB drivers and Microsoft Visual C++ 2013 is required for operation. *mmWaveStudio* allows full board control, board firmware uploading via RS232 connection, TI Radar board configuration, and data capturing via DCA1000EVM or TSW1400EVM capture board for raw ADC data capture[1].

[1]Version 2.1 of *mmWaveStudio* was used in this report.Version 3+ does not support the AWR1642EVM board series. Refer to proper version documentation if using any other TI devices.

3.1.1 Install and Operation

Program installation is trivial, with TI providing many guides for setup and operation. Local admin privileges are required for correct installation and operation. Two open USB ports with proper FTDI drivers installed, an Ethernet port with a configurable static IP address, seen in Figure 3.2, must be available to operate on a PC. A third USB must be available if this system will also be using a USB camera. Operation of *mmWaveStudio* will require the hosting computer firewall to be disabled or granted manual access to connect to the DCA1000EVM properly.

mmWaveStudio walks the user through the setup and data capture process. The following subsections highlight spots not covered clearly by TI's documentation and were learned through trial and error. The "MMWAVE Studio User Guide" should be used in conjunction with this section.

3.1.2 Connection

Initial connection and startup of *mmWaveStudio* is divided into six stages. First, the user must connect the RS232 operations that control the xWR radar boards. Selection of the correct COM port allows the board to receive commands from *mmWaveStudio* seen in Figure 3.2. After the connection is established, two separate firmware files must be loaded and verified[2]. After successful connection, the SPI is connected, and the power-up of the RF system is initialized. At this stage, it is recommended that the user sets up the DCA1000EVM connection seen in Figure 3.3[3]. *mmWaveStudio* will not remind you to do this step like the rest of its interface.

[2]Loading firmware would frequently freeze halfway through operations and requires the xWR board to be powered on and off and a reset of *mmWaveStudio*.

[3]This section has a pop-out window. The GUI should verify the connection. This window has a tendency to be hidden off the main screen. The scrollbar at the bottom must be all the way set to the left.

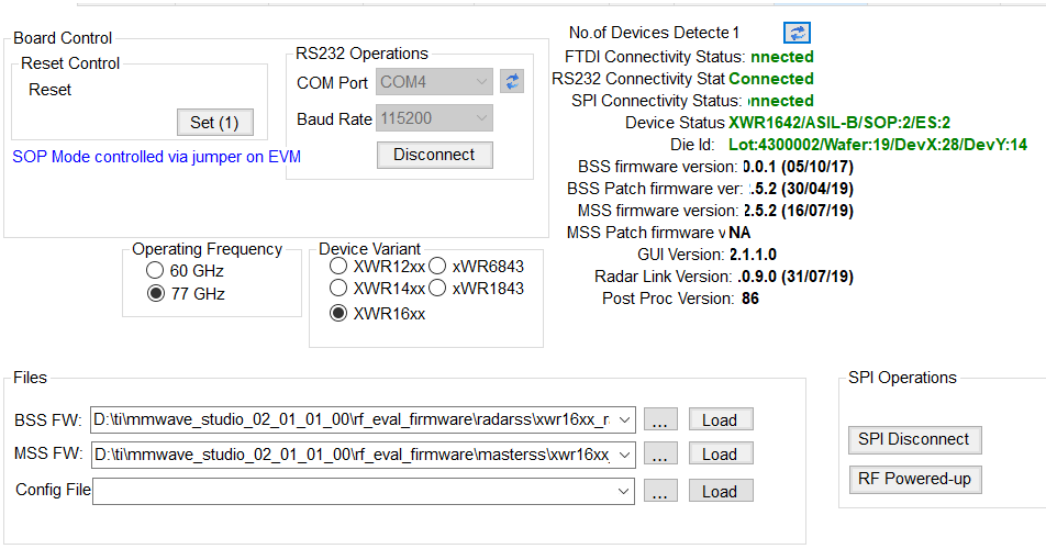


Figure 3.1: MMWAVE studio connection showing connection to AWR1642 and DCA1000 is ready

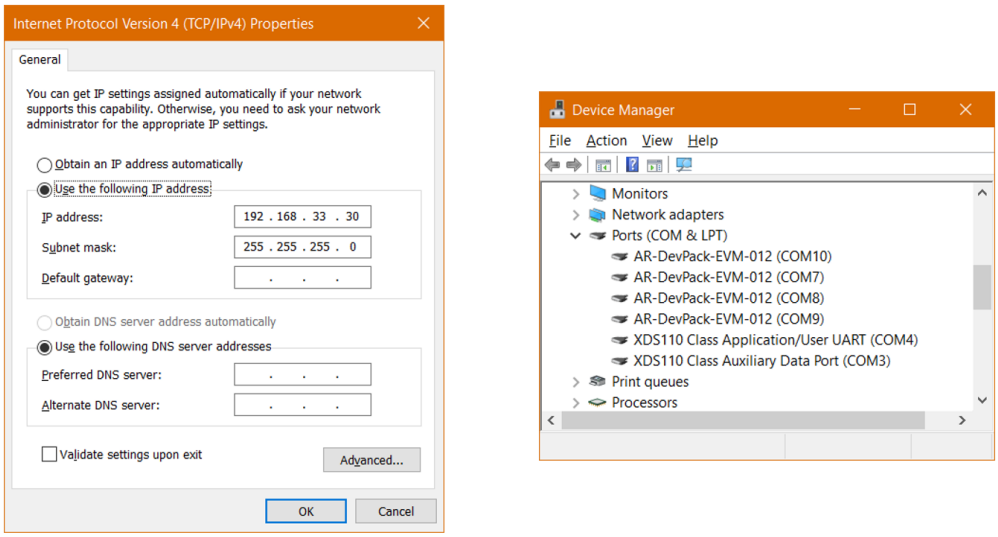


Figure 3.2: Two manual configurations to run *mmWaveStudio*: A) Static Ethernet setting at 192.168.33.30 B) FTDI driver install seen in COM4 and COM5

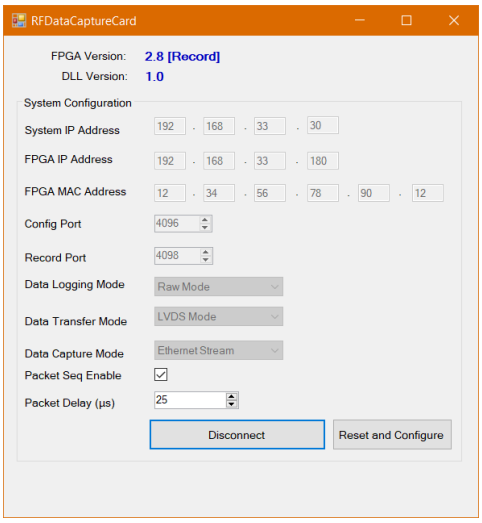


Figure 3.3: Screen Verifying that the DCA1000 is ready to capture data. Note the static IP. This must be manually set on the Ethernet port

3.1.3 Static, Data, and Test Source

The "Static Configuration" tab allows the user to configure the ADC. Documentation of selections is critical used further in the data processing covered in Chapter 5. This tab also controls the transmission and receiving channels used on the radar array, the ADC storage configuration, and the antenna operating modes. The format of the ADC data has four options:

Mode	Description
Real	No formal TI documentation on this mode.
Complex1x	Default mode that was used. This mode returns the raw ADC data in a traditional I and Q format.
Complex2x	This mode enables the image band to have "negative frequencies." Data can be seen as if it is behind the radar. Complex2x mode should only use this mode in interference detection and monitoring.
PseudoReal	No formal TI description of this mode.

Table 3.1: Modes for ADC data capture as documented by TI.

Raw ADC data is arranged based on the ADC Mode, capture board, and XDR board. [4]. Example code and explanations on how the data storage for each combination can be found in the "MMWAVE SDK User Guide"[7] and "Mmwave Radar Device ADC Raw Data Capture"[24]. The "Data Configuration" tab controls the LVDS lane configuration and clock. Default operations were never changed during data capturing. The "Test Source" tab allows the end-user to emulate received reflectors at varied positions and strengths.

[4]This is only needed if the user does not utilize the default "PostProc" button in the Sensor Configuration tab

Chirp 1	Rx0I0	Rx0I1	Rx0Q0	Rx0Q1	Rx0I2	Rx0I3	Rx0Q2	Rx0Q3
	Rx0I4	Rx0I5	Rx0Q4	Rx0Q5	Rx0I6	Rx0I7	Rx0Q6	Rx0Q7
	…		…		…		…	…
	Rx3IN-4	Rx3IN-3	Rx3QN-4	Rx3QN-3	Rx3IN-2	Rx3IN-1	Rx3QN-2	Rx3QN-1
Chirp 2	Rx0I0	Rx0I1	Rx0Q0	Rx0Q1	Rx0I2	Rx0I3	Rx0Q1	Rx0Q3
	Rx0I4	Rx0I5	Rx0Q4	Rx0Q5	Rx0I6	Rx0I7	Rx0Q6	Rx0Q7
	…		…		…		…	…
	Rx3IN-4	Rx3IN-3	Rx3QN-4	Rx3QN-3	Rx3IN-2	Rx3IN-1	Rx3QN-2	Rx3QN-1

Figure 3.4: Example of the ADC data for each chirp as it is stored for a DCA1000 EVM with an xWR16xx complex, 4 channel recieve, and 2 LVDS lanes

3.1.4 Sensor, Chirp, and Frame Configuration

The "Sensor Configuration" tab is what controls the FMCW radar characteristics. The tab includes complete FMCW control and chirp profiling. Once initialized, the radar runs and sends a command to the DCA1000 to capture all the raw ADC data and store it in a binary file. The user must select a post-processing button that converts that data to a format that can be read into MATLAB and used in the code in Appendix **??**.

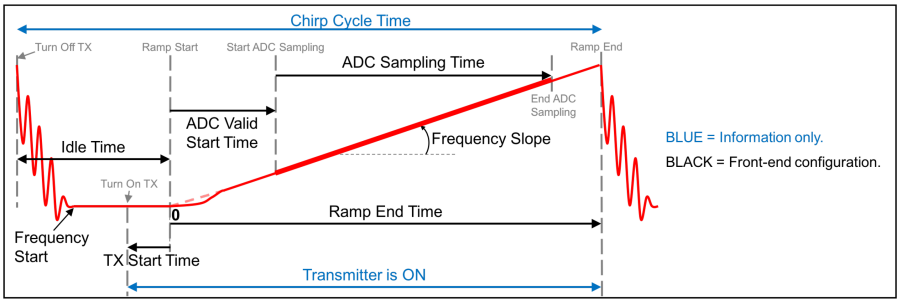


Figure 3.5: Chirp Profile of TI FMCW radar courtesy of the MMWAVE studio SDK manual[7].

3.1.5 mmWaveStudio Post-Processing

mmWaveStudio includes a Matlab post-processing tool that gives users the ability to view all the data with pre-configured radar imaging. The GUI uses two input files: One is the raw LVDS capture data stored as "ADC_DATA.bin," and the other is the LUA script application programming interface (API) calls used during the operation of *mmWaveStudio*. The "PostProc" program can interpret data to provide plots. Plots are adjustable by the user and have several configurable options. The operating software allows users to playback captured frames. The *mmWaveStudio* software development toolkit (SDK) manual contains details of these plots and applications [7].

The "PostProc" tool helps check the captured data and demonstrate radar functionality but does not allow modification to the processing of plots or provide ways to work with saved data. The GUI handles several types of figure generation including range vs. received amplitude (in decibel scale), range-Doppler, angle estimation, and a rudimentary position plot. New implementations of each of these plots was custom written in this thesis to be used with the proposed data capture system in Chapter 6. This custom implementation enables new flexibility for future improvements to utilize, and interpret the data captured by the radar. Further, this implementation is required for fusing the radar data with the images captured by the webcam.

It is important to note that verifying the logs for the capture time is necessary when using this tool. Files stored by default are not verified by *mmWaveStudio* to match logs. *mmWaveStudio* will read an older BIN file and try to fit the settings used for capture in that GUI.

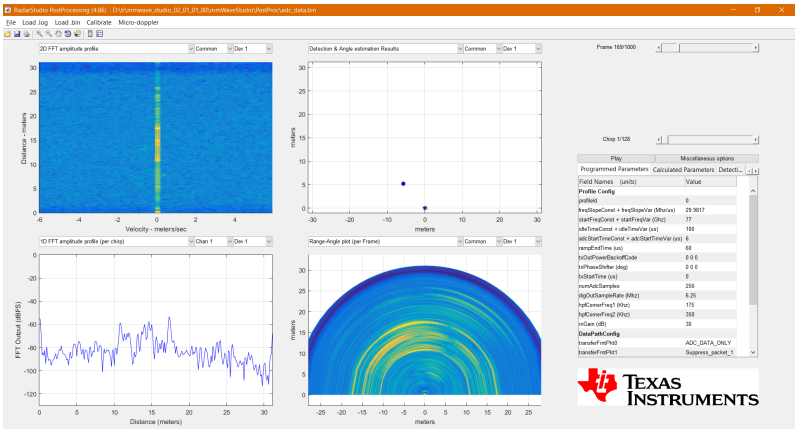


Figure 3.6: Data seen after being captured in MMWAVE studio.

3.2 Video Capturing

Capturing video data was initially started while trying to completely automate the data capture process in Matlab. The idea was to create a video feed at the time of triggering the data capture using *mmWaveStudio*. The ability to automate the process had to be dropped due to hardware constraints of the laptop operating this system. In order to increase efficiency, a Python script using the *OpenCV* library was used. This open-source library is cross-platform and free-for-use under the open-source Apache 2 License written in C/C++. A simple tutorial script was modified and can be seen in section **??**. Figure 3.7 shows the steps leading from configuring the hardware to operations of data and video capture proposed in this system.

3.3 Final Remarks

mmWaveStudio is an easy-to-use software toolset with numerous capabilities that make data collection and visualization simple and quick using the TI line of mmWave sensors. Despite the limits of the software for post-processing, it reliably

allows users the ability to capture and save data for later use. However, *mmWaveStudio* is not a resource-friendly application. Automation of data capture is recommended for any massive data collection capabilities, either through pursuing an external scripting system or utilizing the internal LUA scripts that run *mmWaveStudio*.

Combined video-radar data acquisition was successful in this thesis. However, operating both *mmWaveStudio* and the video capturing scripts becomes a resource-intensive process for the laptop used in this thesis. With a tremendous amount of data coming from both Ethernet and USB, the processing and memory needed to operate pushed the hardware limits of the laptop. Future real-time data acquisition and experimentation will require a higher-end laptop with increased RAM and processing power.

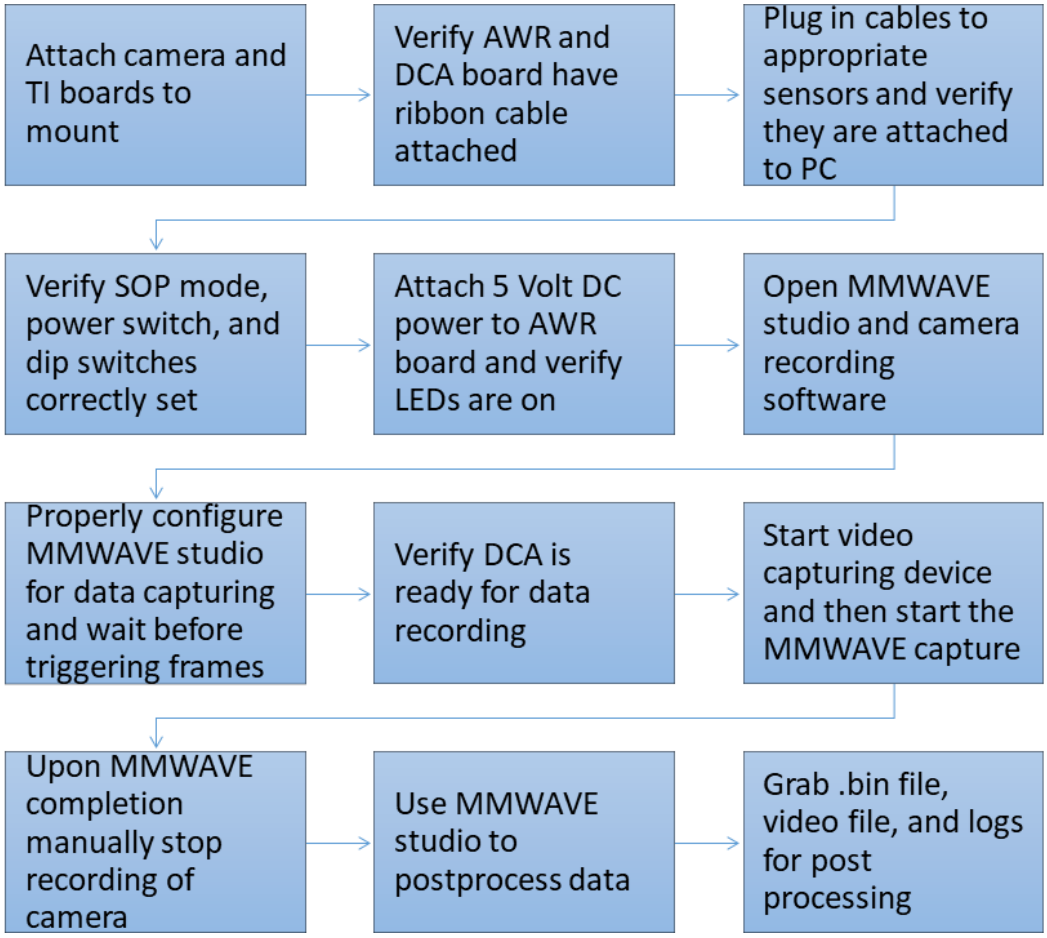


Figure 3.7: Data Capture Process.

Chapter 4

FMCW Data Processing

The radar data cube is a concept representing complex RF signals in space-time processing broken into three dimensions[8]. "Fast-time" represents signals in the down-range space where the resolution is controlled by the bandwidth of the radar. FMCW radar transmits a periodic series of pulses continuously. Processing data across these repeated chirps is the "slow-time" dimension. The frequency of chirp repetition is known as pulse repetition frequency (PRF). The inverse of the PRF is the pulse repetition interval (PRI), which is the time (in seconds) between each sweep and represented in Figure 4.5. The final dimension is accessible to radar systems with multiple simultaneous receive channels (i.e., outputs). Multiple input, multiple output (MIMO) systems also have the ability to emit independent waveforms from each transmit antenna. MIMO antenna systems are becoming common in automotive radar systems such as the AWR series of radars. An example of the system diagram for a two receiver channel system is in Figure 2.3. Every receive channel, either virtual or real, stacks its frequency sweep and PRF into a cube. Analyzing combinations of these dimensions produce various forms of radar imaging.

4.1 Radar Date Cube

The radar data cube represents complex RF signals in space-time processing. The dimensions are broken up across "fast-time," "slow-time," and receiver channels [8]. "Fast-time" represents the range from the radar, controlled by the sampling interval. In FMCW radar, "slow-time" represents the repeated signals by a set number of pulses. Figure 4.1 shows a diagram of "fast-time" and "slow-time." The final

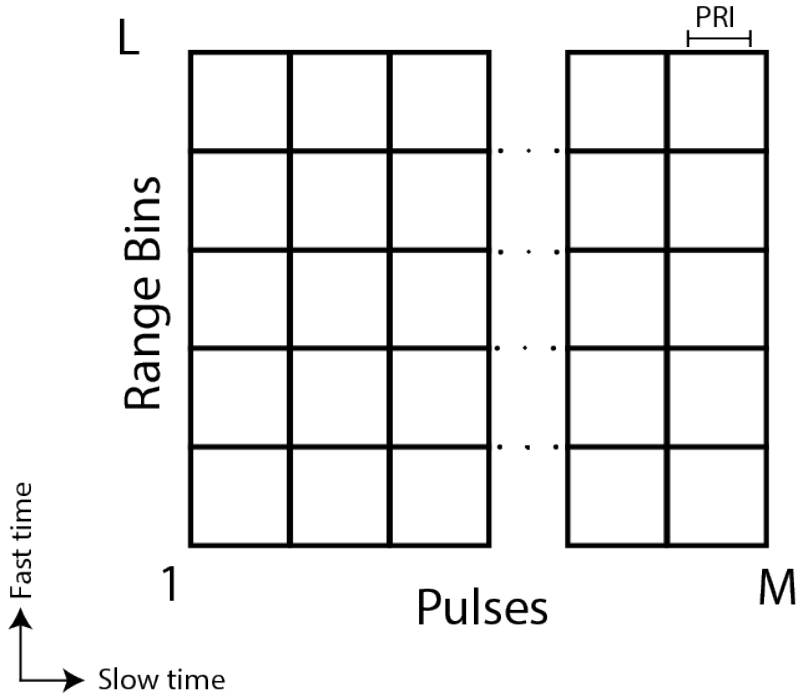


Figure 4.1: A representation of received data on one channel in FMCW Radar. [8].

dimension is accessible to radar systems with multiple simultaneous outputs. Multi receiver systems need the ability to store data for future processing. An example of the system diagram for a two receiver channel system is in Figure 2.3. *mmWaveStudio* captures data on a periodic scale set by the user. A single frame of data with four receive channels comes out to 0.5 MB of data.

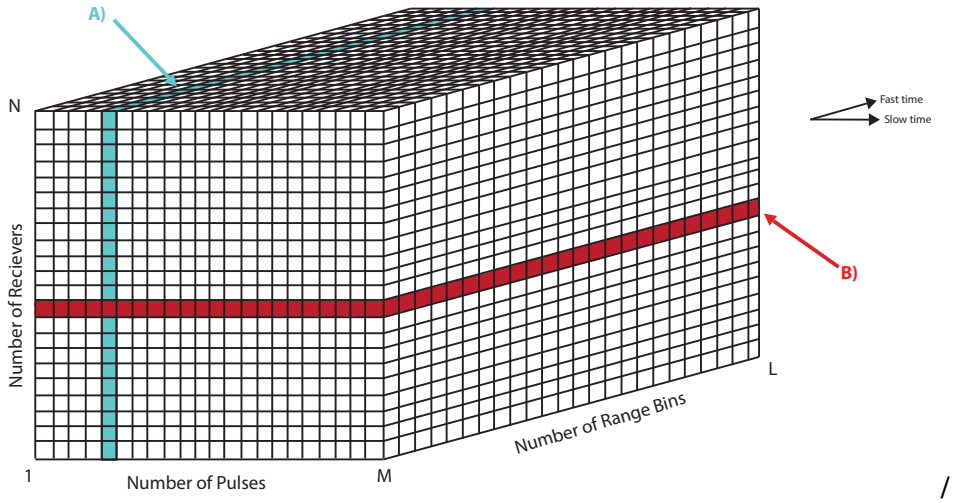


Figure 4.2: Radar cube is used to describe captured data in multi channel pulsed radar [8]. A) Data slice used to get an Angle Estimation. Fourier transform taken across receivers produces angle information. B) Slice used to create Velocity Range plot. Fourier transform taken across pulses produces velocity information.

4.2 Storing Radar Data

Storing a single radar cube of data is trivial and easy to manage. Experiments with long duration data captures can result in significant data storage and real-time processing requirements. This section breaks down the reasons planning and proper management of data are necessary, as well as how *mmmWaveStudio* handles raw ADC data.

mmmWaveStudio refers to all complete raw ADC data across all three dimensions as frames. Figure 4.3 displays a series of radar data cubes captured over time similar to frames in a video. The typical capture rate used in this thesis was set to 50ms per frame with a maximum of 1000 frames. The raw ADC BIN file size at 1000 frames came to about 500MB and around 50 seconds in length. Additional considerations for memory management come when recording video during a capture event. Video memory size depends on the video quality and frames per second

(FPS). Both data sets quickly fill hard-disk space and space requirements should be done before any field testing, using equation (4.2), where K is the maximum number of frames desired and M stands for the maximum memory in MB.

$$M = \frac{K}{2} \tag{4.1}$$

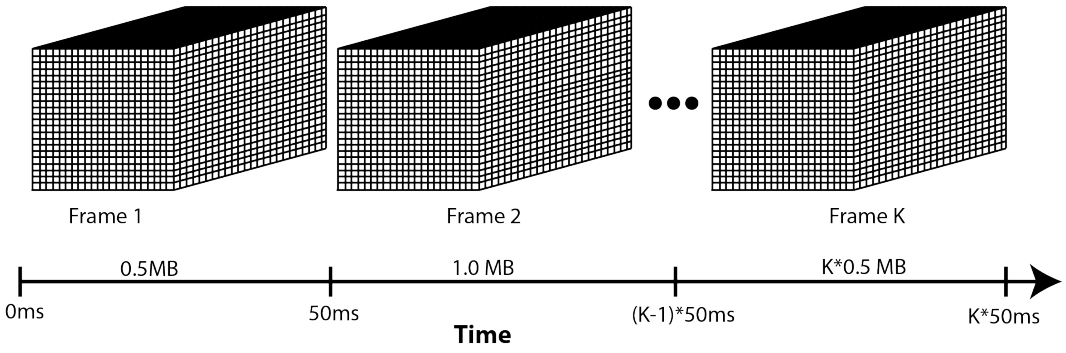


Figure 4.3: Representation of raw ADC data captured over time.

This thesis utilized a TI-provided Matlab script to read the raw ADC BIN file generated from the "PostProc" tool, found here **??**. This method requires only the radar capture parameters set by the user in *mmWaveStudio*. Data can also be read directly without using the "PostProc" tool. This method requires referring to the "mmwave Studio User Guide" to handle and read data properly. The final option uses a CLI executable function to capture the raw ADC data directly from the DCA1000EVM. This tool is found in *mmWaveStudio* code. Application of the CLI tool still requires some other means of running the xWR series board. Figure 4.4 shows a quick overview of these methods.

mmWaveStudio stores captured raw ADC data from the DCA1000 into a binary file. This thesis did not find hard limits for maximum data capturing time. Assumptions were made that data gathering happens indefinitely, as long as hard-disk

memory is available on the system. However, later processing of data may be affected. For example, a test captured data two minutes in length. The size of the raw ADC binary file was around 1GB in size. Reading of data sets in the gigabyte scale can push computers with limited RAM. [1] Recommendation for data capture is to limit data to around 1000 frames without modifications to the code supplementary to this thesis. Matlab frequently crashed when processing data sets over 1GB with the system specifications in Table 4.1.

System	Specifications
CPU	Intel® Core™ I7-8700K CPU operating at 3.7 GHz
RAM	32 GB DDR4 RAM
GPU	NVIDIA GeForce GTX 3080

Table 4.1: Computer specifications for all post processing

[1] Matlab has support functions for large files and big data but they were not used in this thesis.

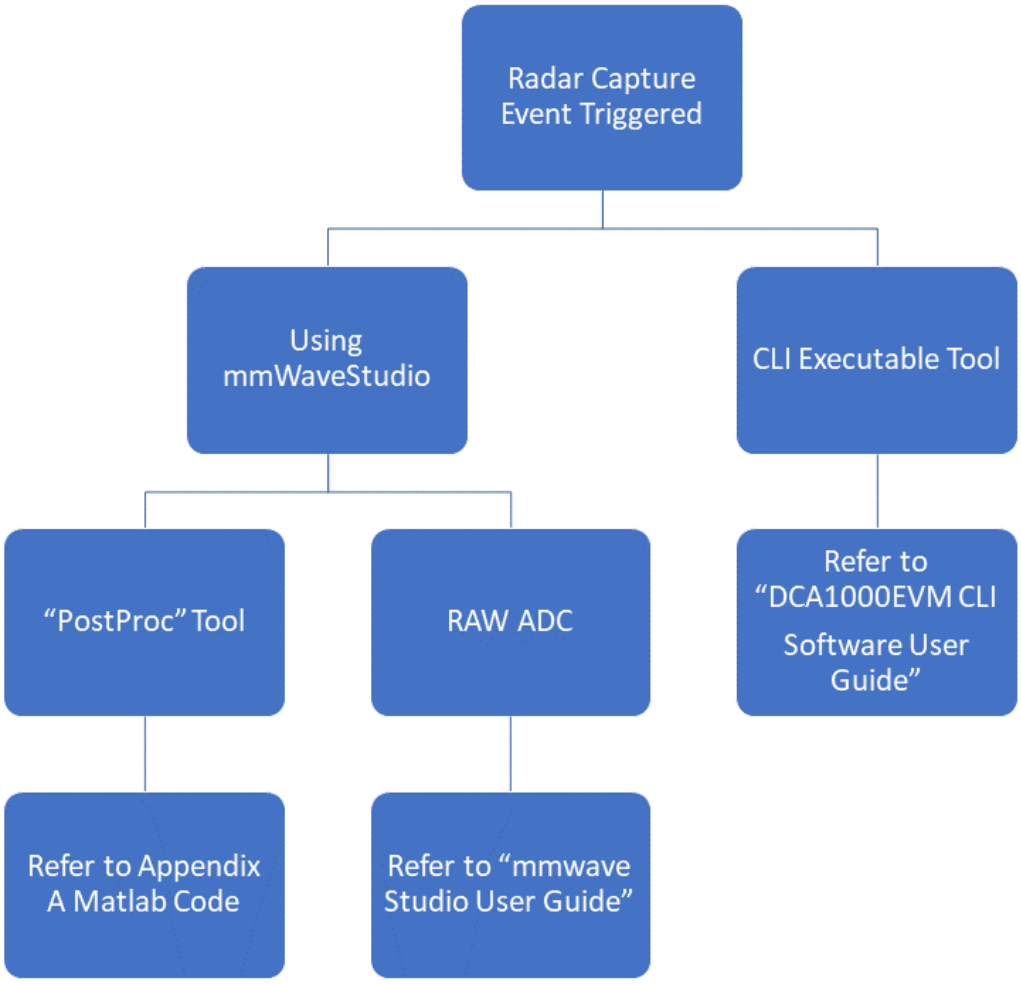


Figure 4.4: The three methods for capturing RAW ADC data using *mmWaveStudio* and DCA1000EVM.

4.3 FMCW Radar

This section covers the background information of FMCW radar and its applications in range processing. Continuous-wave (CW) radar transmits and receives continuously. This feature is why CW radar is frequently used in safety systems. CW systems do not have problems with gaps in range measurements[8]. While limited to short-range, CW radar is a perfect fit for, ground-based sensors for safety-critical systems. The most common CW waveform is a linear FMCW system. In this system, one can imagine a sinusoidal signal that compresses over time. This compression comes from the frequency increasing in a linear slope across a set bandwidth sweep. A sinusoidal signal from an FMCW chirp plotted as a function of time versus frequency generates a sawtooth waveform tooth waveform. Figure 4.5 shows both examples of these waveforms. The bandwidth set by the FMCW chirps plays a crucial role in setting down range resolution that will be explored further in the chapter. Characterizing the chirp can be broken into three parameters: start frequency f_s, bandwidth B, and chirp time T_c, seen in equation (4.2).

$$\alpha = \frac{B}{T_c} \tag{4.2}$$

Where the variable α represents chirp slope.

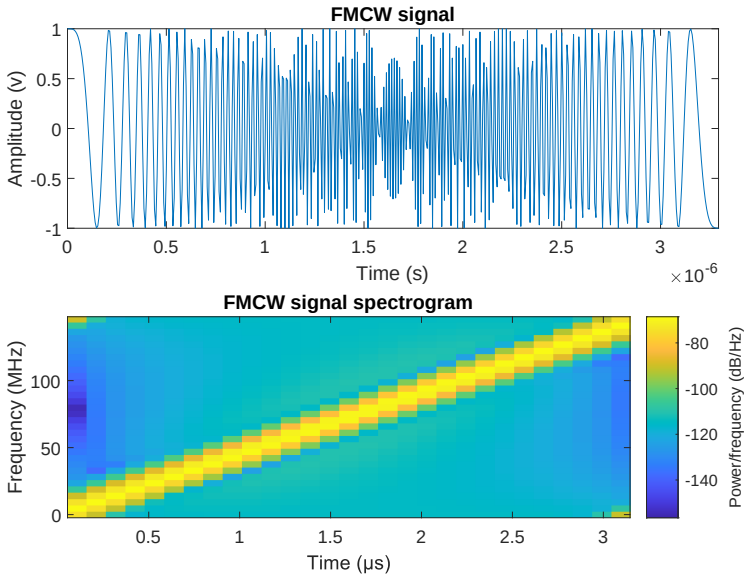


Figure 4.5: Top Plot shows example of FMCW Signal simulated in MATLAB. Bottom plot shows an example of the frequency slope of signal.

4.3.1 Range Processing

The most common application for an FMCW radar is range estimation. This allows the distance to be measured from the antenna array to objects in line of sight of the radar. Calculating distance requires finding the instantaneous frequency (IF) signal. Finding the IF tone starts at the mixer block output, seen in Figure 2.3, and described [9]:

$$x_{out} = sin[(w_1 - w_2)t + (\phi_1 - \phi_1)] \tag{4.3}$$

x_{out} is found by taking the difference in IF and phase between the transmitted and received signal. The difference is fed into a low pass filter and gets sampled by the AWR1642 ADC. Setting the sampling rate,F_s, of the ADC determines the maximum distance resolvable with the following equation,[9]:

$$R_{max} = \frac{F_s c}{2\alpha} \tag{4.4}$$

Any object in front of the radar will produce an IF signal that has a constant frequency tone. This tone can be used to calculate the distance from the radar to the object detected where τ is the round-trip propagation delay[25]. Further, two tones can be resolved in frequency as long as the delta frequency is greater than the inverse of chirp time[26]. Knowing that $\Delta f = \frac{2\alpha\Delta R}{c}$ we can derive the distance resolution[26]:

$$\Delta f > \frac{1}{T_c} \Rightarrow \frac{2\alpha\Delta R}{c} > \frac{1}{T_c} \Rightarrow \Delta R > \frac{c}{2\alpha T_c} \Rightarrow \frac{c}{2B} \tag{4.5}$$

Or more simply:

$$\Delta R = \frac{c}{2B} \tag{4.6}$$

This definition shows that bandwidth of the signal is the defining property for range resolution [25][26]. Further examples of how bandwidth can be used to resolve objects in distance can be referred to in Figures 4.6 and 4.7.

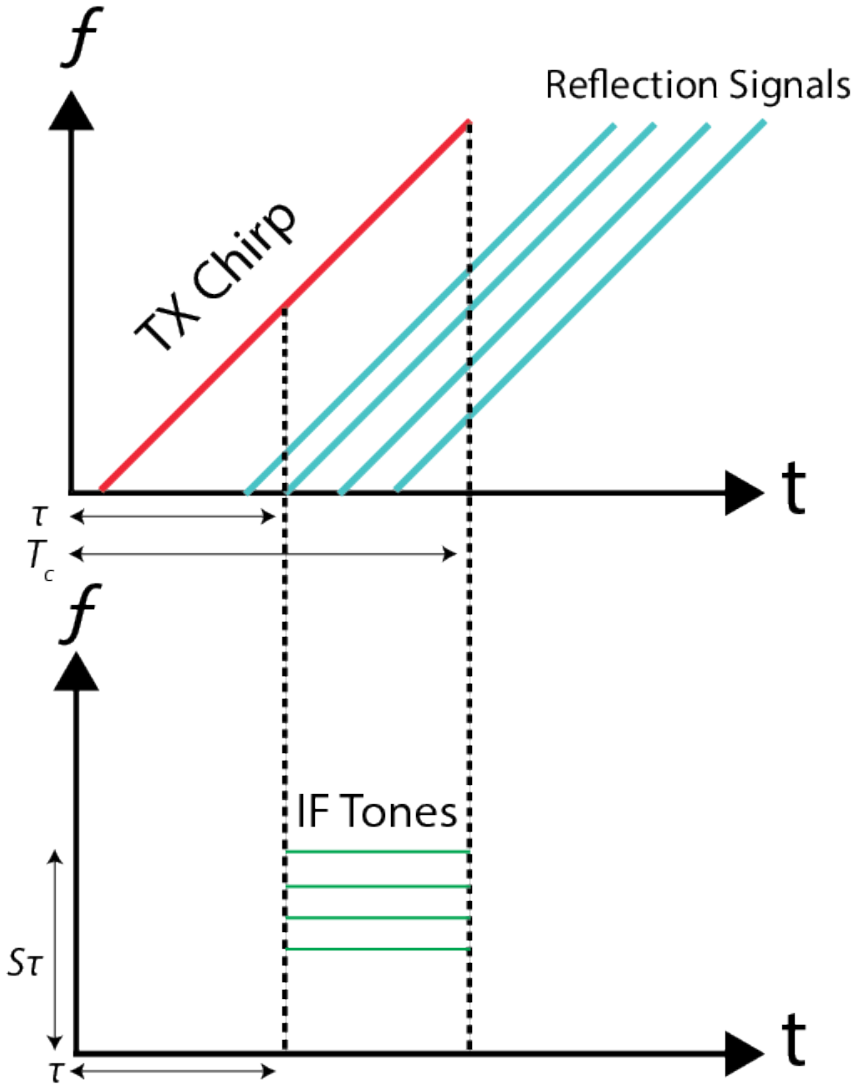


Figure 4.6: Reflected signals from multiple detected objects and the IF tones created.

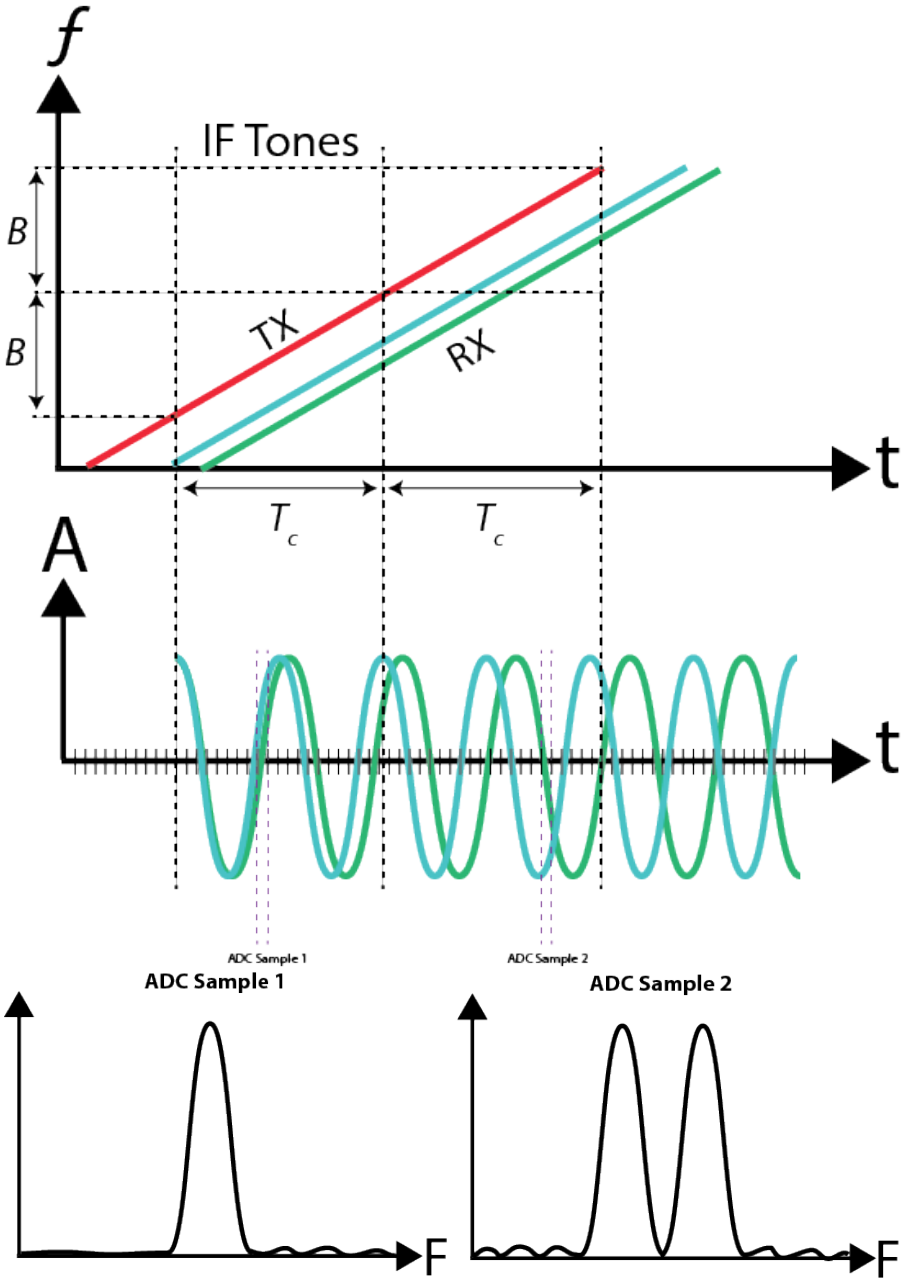


Figure 4.7: IF Tones showing how increased bandwidth can help resolve signals. ADC Sample 1: the bandwidth is too close and will show up as a single peak in the frequency spectrum. ADC Sample 2: A higher bandwidth allows the signals to be resolved in the Frequency spectrum

4.3.2 Range Estimation

Now that the theory of range processing has been defined, this section will describe using the Fourier transform on the radar data cube to generate down range plots. The process requires taking a sample of the raw ADC data for each slow-time chirp. The data sample is run through a window function and processed using the discrete Fourier Transform (DFT). The power of the system is calculated and scaled using the following definition provided by TI:

$$P_{Rx,norm}(dB) = P_{Rx} - 20log_{10}(2^{n-1}) + 20log_{10}(w_{sum}) - 20log_{10}(\sqrt{2}) \quad (4.7)$$

Where variable n represents bits of the radar and w_{sum} represents the sum of the window [27].

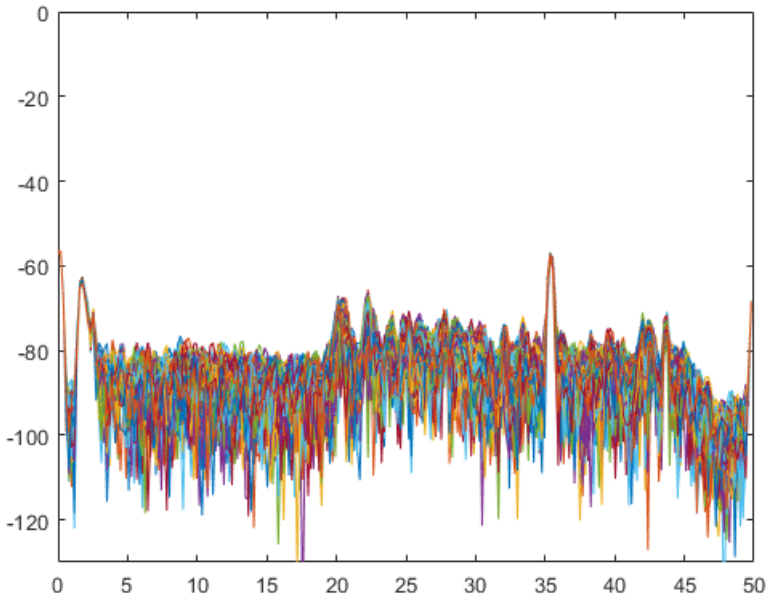


Figure 4.8: Example of slow-time chirps plotted as Distance in meters vs. normalized receive power (n dB). Peaks represent objects being detected at a certain distance. First two peaks are the direct path coupling and ground response (and therefore a measure of antenna height from the ground), respectively.

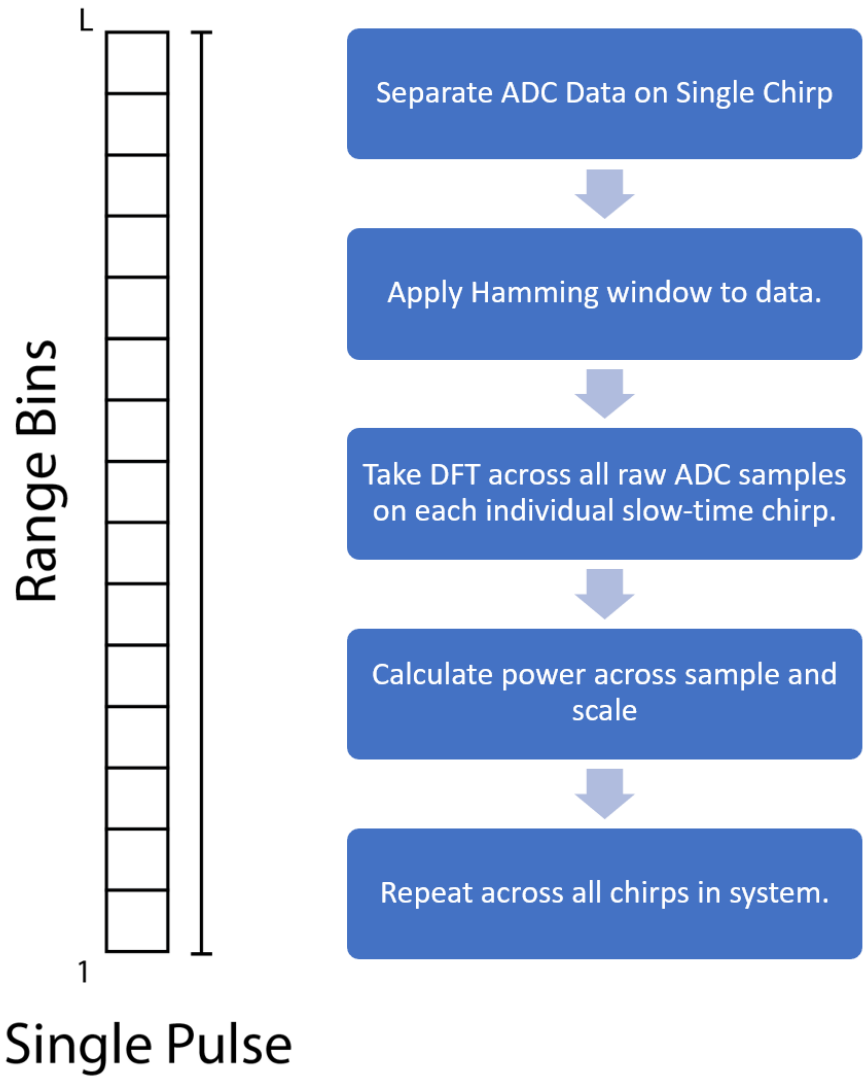


Figure 4.9: Process to calculate down-range plot.

4.4 Range-Doppler Processing

Estimating velocity is a standard application in automotive radar for adaptive cruise control. Newer systems utilize velocity for collision avoidance as well as distinguishing objects at the same down-range point. In this sense, a two-dimensional figure is generated, which allows for object discrimination on the road. This section will break down handling velocity estimation with FMCW radar and implementing a routine using raw ADC data captured.

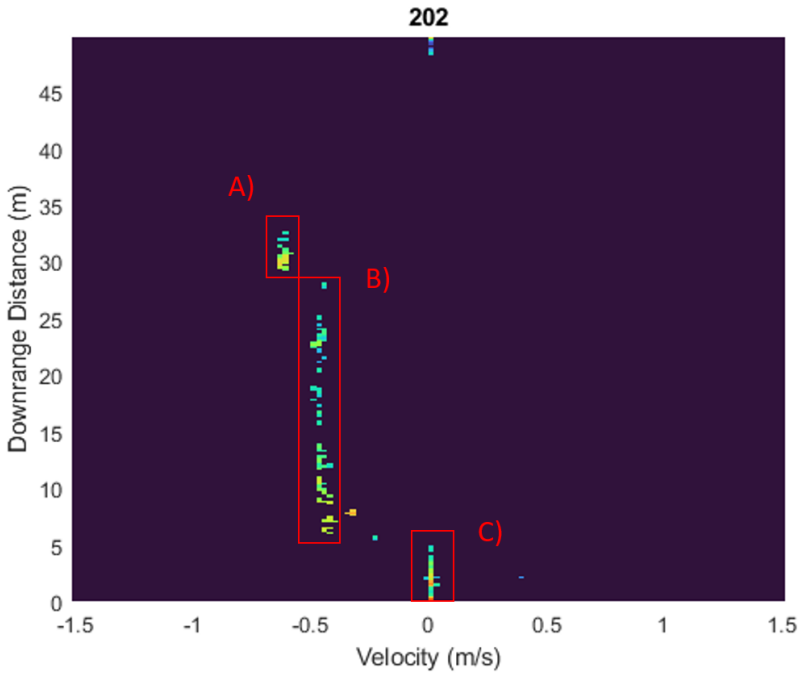


Figure 4.10: Example of a typical velocity vs down-range plot. A) An object moving a different velocity B) Stationary objects end up in a single velocity bin corresponding to the speed of the observing radar. C) Clutter due to internal reflections from the car taking data. From this frame of reference it ends up at the zero Doppler position.

4.4.1 Velocity Estimation

Measuring velocity is done by comparing the phase of return signals across the slow-time dimension. Each phase is compared across each chirp time, T_c. While the amplitude for an object at the same distance will appear at the same position during each chirp, the phase will differ due to the Doppler effect. This difference in phase can be taken advantage of to calculate phase difference [26]. This starts with defining the Doppler frequency, F_d, as:

$$F_d = \frac{2v}{\lambda} \tag{4.8}$$

Now representing the change between chirps as $\Delta\phi$:

$$\Delta\phi = F_d T_c \tag{4.9}$$

Therefore, this change is the change in frequency (i.e., the Doppler frequency) F_d multiplied by the chirp time (or chirp duration), T_c. Substituting (4.9) into (4.8) provides the progressive change (in cycles) across chirps, $\Delta\phi$, in terms of the velocity and time between chirps. This change is found to be:

$$\Delta\phi = \frac{2vT_c}{\lambda} \tag{4.10}$$

Then converting this change to radians through a factor of 2π radians/cycle yields is added to derive the difference in phase between chirps:

$$\Delta\omega = 2\pi\frac{2\Delta vT_c}{\lambda} = \frac{4\pi\Delta vT_c}{\lambda} \tag{4.11}$$

The sign of the phase change determines direction. A positive phase change means movement away from radar, $|\omega| > 0$. A negative phase is movement towards the radar $|\omega| < 0$. Phase measurement is unambiguous only if: $|\omega| < \pi$ rads. With this definition, Maximum velocity can be derived with the following [26]:

$$\frac{4\pi v T_c}{\lambda} < \pi \Rightarrow v_{max} = \frac{\lambda}{4T_c} \tag{4.12}$$

Using equation (4.11), we can see maximum velocity, v_{max}, can be increased by decreasing chirp time, T_C (i.e., increasing the chirp repetition frequency).

Resolving multiple objects in a frame can be done using a similar technique to finding objects in down-range. Taking the DFT on the phasors from the down-range DFT, resolves objects [26]. This creates a range-Doppler plot scene in Figure 4.10. Graphical representation of this process can be seen in Figures 4.11 and 4.12. Further expansion can be used to calculate velocity resolution, v_{res}. The DFT on a sequence of length M, separates frequencies of phasors as long as:

$$\Delta\omega = \mid \omega_1 - \omega_2 \mid > \frac{2\pi}{M} \tag{4.13}$$

Establishing the inverse of the coherent processing interval (i.e., the frames per second) as:

$$\delta F = \frac{1}{T_{tot}} = \frac{1}{N_p * T_c} \tag{4.14}$$

where N_p is total PRI chirps. When applied to equation (4.11), one can derive the velocity resolution, v_{res} [26]:

$$\Delta\omega = \frac{4\pi \Delta v T_c}{\lambda} \Rightarrow \Delta\omega > \frac{2\pi}{N_p} \Rightarrow V_{res} = \delta F \frac{\lambda}{2} \tag{4.15}$$

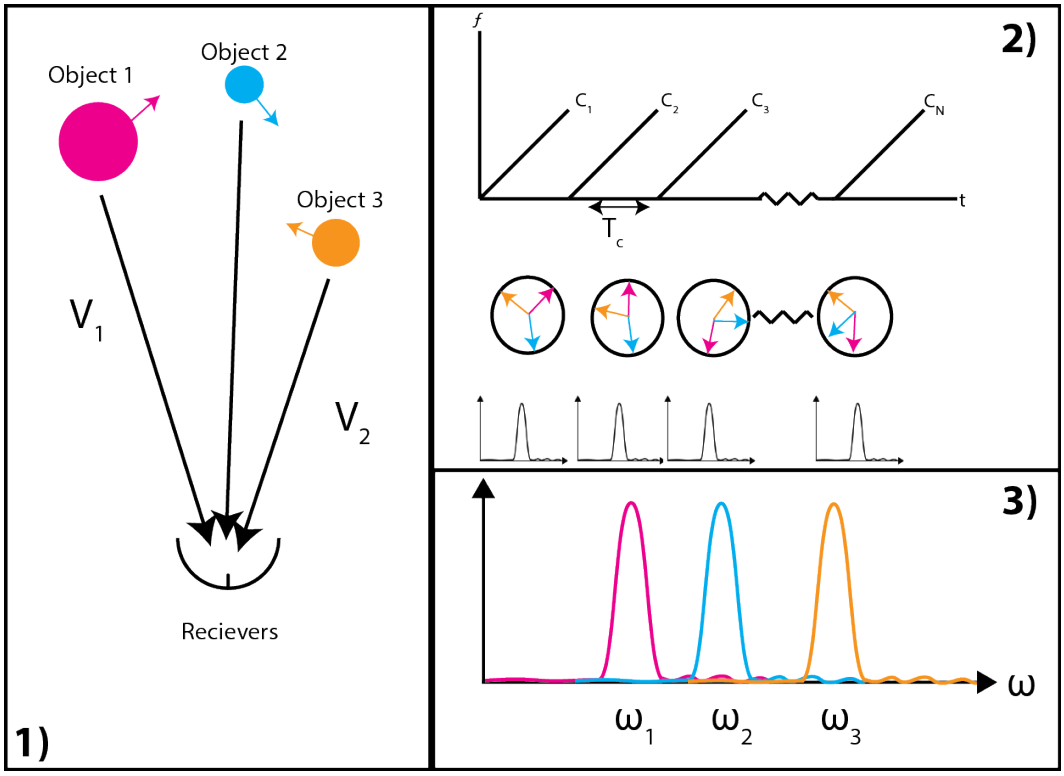


Figure 4.11: 1) Multiple Objects are moving with a certain velocity and are positioned in various down-range positions. 2) In every time sample the phase of each object across slow-time is stored 3) Applying the DFT resolved the phase difference that can be plotted

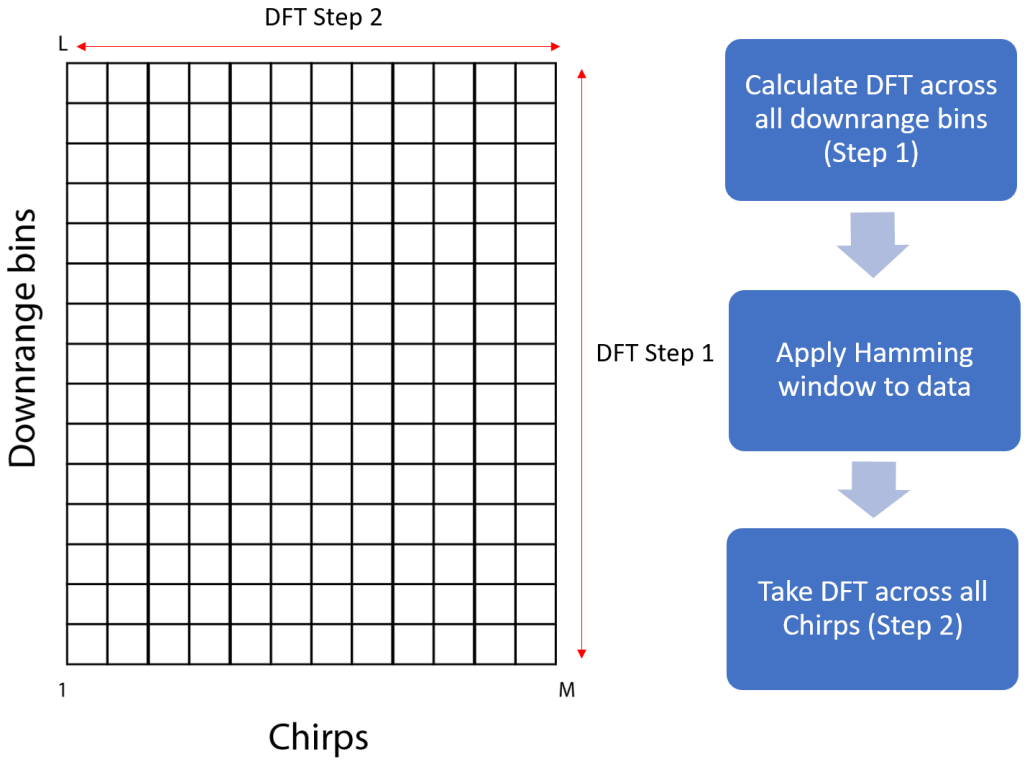


Figure 4.12: Process to calculate Velocity vs down-range plot

4.4.2 Velocity Aliasing

This section highlights the importance of the proper setup of the sample rate when capturing data. During the capture process, issues with velocity aliasing came up when driving beyond the v_{max} defined in equation (4.12). The maximum velocity while driving must be considered prior to data capture. Any speed greater than the maximum Doppler velocity results in the Doppler response wrapping around the x-axis of the plot. In Figure 4.13, fence posts and trees taken while driving beyond the pre-calculated v_{max} resulted in data wrapping.

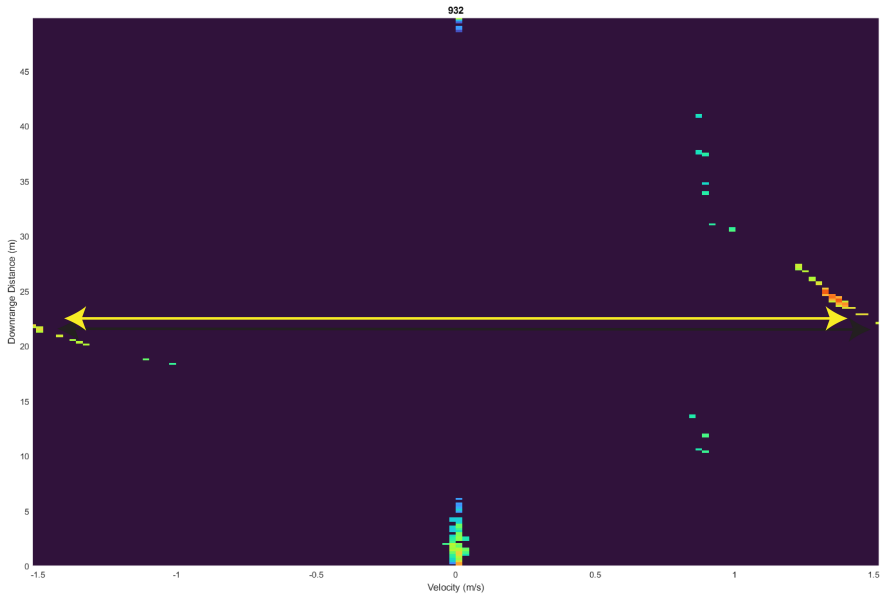


Figure 4.13: This figure shows aliasing in a range-Doppler plot.

4.5 Angle Estimation using MIMO Radar

4.5.1 MIMO and FMCW Overview

Multi-input-multi-output (MIMO) radar with multiple transmit and receive antennas can be used to achieve improved angular resolution with respect to traditional phased arrays. Angle estimation exploits the change in phase across the FMCW chirps across each antenna element. Angle estimation resolution is linearly related to the electrical distance spanned by the RX and TX antennas configured on a MIMO array. For example, an array of one TX and four RX antennas can double its angular resolution by adding a second TX antenna. This form of MIMO radar creates what is known as virtual antennas. These represent the position an actual array would have by taking advantage of the ability to isolate the responses from each of the TX antennas.

In Figure 4.14, subfigure A the antenna positioning as it appears on the AWR1642 EVM is shown. Each receive element is spaced by $\frac{\lambda}{2}$, where λ is the wavelength of the center frequency of the radar. Because the two transmitters are spaced 2λ it is possible to effectively double the "virtual" receive antennas as seen in Figure 4.14, subfigure B. This setup enables a virtual antenna configuration of eight antennas spaced $\frac{\lambda}{2}$. In order to take advantage of this spacing requires a split configuration of chirps of the FMCW radar. Traditionally, the system would continuously chirp on a single TX antenna. For MIMO operations the radar uses time division multiplexing (TDM) where the two transmitters alternate transmissions. The consequence of this technique is that number of effective chirps per frame is halved to maintain the same capture time for a single frame. This configuration requires specialized handling of data when captured on the DCA1000.

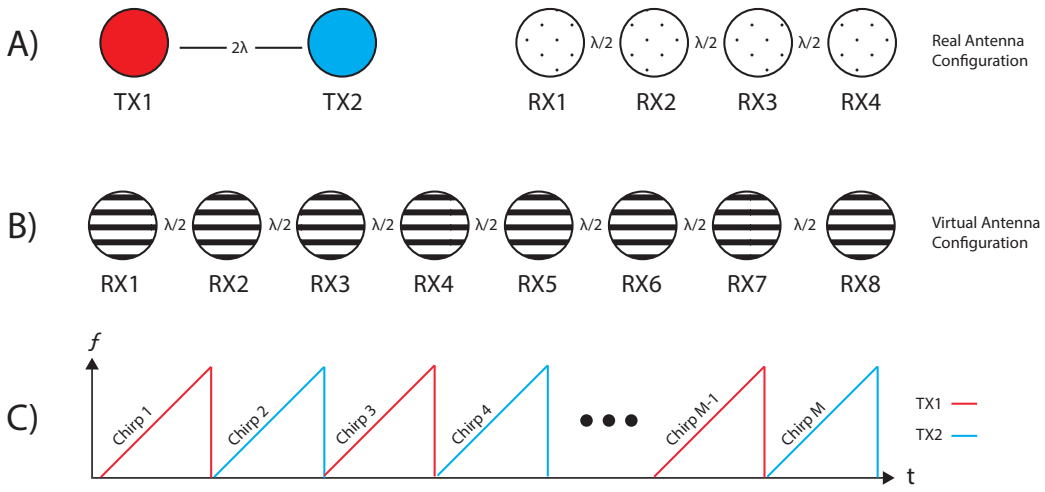


Figure 4.14: A) Configuration of MIMO Antenna on the AWR1642. B) Virtual antenna positioning that can be interpreted when using two transmitters on AWR1642 C) Chirp profile showing how chirps are called to each transmitter when operating in multi transmit mode.

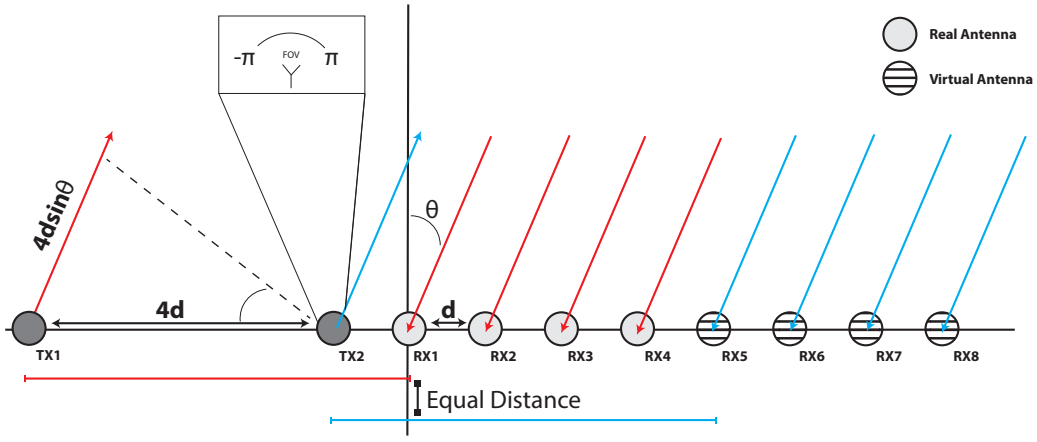


Figure 4.15: Concept of angle estimation demonstrated on AWR1642 antenna array[9]. θ defined in equation 4.16.

Once the virtual antenna array is setup and data is collected processing can continue. Angle estimation measures the phase difference $\omega = \frac{2\pi}{\lambda} dsin(\theta)$ between signals on all receive antennas. Using this newly calculated phase difference one can find the angle of arrival[9]:

$$\theta = sin^{-1} \frac{\omega\lambda}{2\pi d} \tag{4.16}$$

Using equation 4.16 with our spacing of receive antennas $\frac{\lambda}{2}$ and setting ω between $(-\pi, \pi)$, we find a max field of view of $\pm$ 90 °. Further, assuming the same receive antenna distance and a $\theta = 0$ we can arrive at the following equation where N equals the number of virtual antennas:

$$\theta_{res} = \frac{2}{N} \tag{4.17}$$

A simple implementation of an angular plot can be seen in Figure 4.16. The data was zero padded to increase the angles resulting in a large return of objects unresolved in the x-axis.

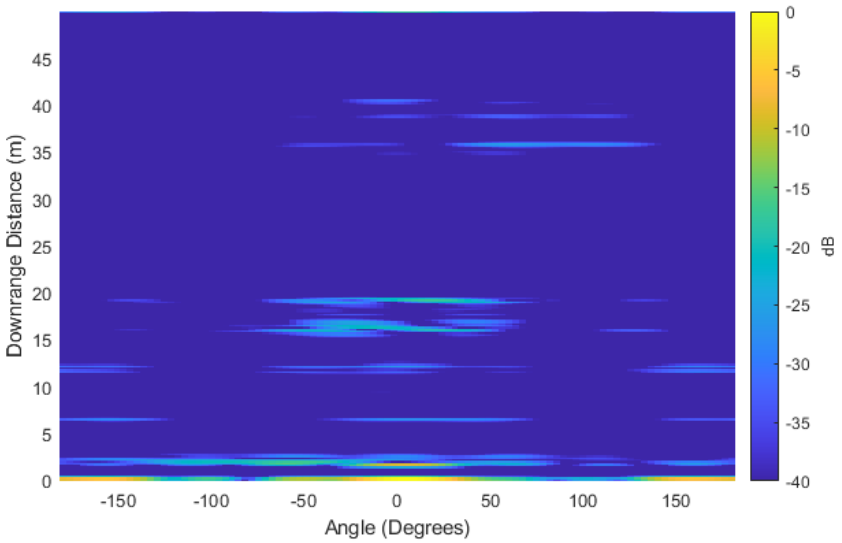


Figure 4.16: Implementation of angle estimation with passing cars at a stoplight.

4.6 Summary

For reference, a summary of the primary equations explained in this chapter are in Table:4.2.

Type	Resolution	Max
Range	$R_{res} = \frac{c}{2B}$	$R_{max} = \frac{f_c}{2\alpha}$
Velocity	$V_{res} = \delta F \frac{\lambda}{2}$	$V_{max} = \frac{\lambda}{4T_c}$
Angle	$\theta_{res} = \frac{2}{N}$	$\theta_{max} = -\frac{\pi}{2}, \frac{\pi}{2}$

Table 4.2: Summary of standard range-Doppler and angle estimation equations.

Chapter 5

Post-Processing GUI

5.1 Radar Fusion Interface

A GUI program was created for this thesis to help radar-video synchronization and develop post-processing algorithms. Matlab's "App Designer" program was used to create this GUI. The program aided in the development of all radar image processing techniques mentioned in this chapter. Features include:

- GUI interface support
- File loading for raw ADC and Video file
- Time corrections to sync loaded files
- Individual frame selection for syncing video and radar files
- Display of processing parameters used
- Active updating of post-processing parameters

The code utilizes proper class definition to allow further expansion into all other models. Processing the raw ADC data gives the user the ability to test everything from windowing effects to zero padding. The program will also read in captured

raw ADC data, video data, and log files to sync the frames captured between the radar and video systems. Ti's *mmWaveStudio* has many of these features by default but does not allow the user to modify the actual code for processing, limiting the application to verify data and results. The desire was to create an open, expandable framework for future radar processing algorithms and hybrid fusion applications testing.

5.1.1 System Time Syncing Background

A "hard real-time" task is defined by results produced after a deadline causes catastrophic results [28]. "Hard real-time" systems require an internal clock on the data capture system. Unfortunately, using *mmWaveStudio* without a devoted clock makes it impossible to sync timing between the radar and camera systems with exquisite fidelity. "Soft-time" logs compensate and synchronize frames between the radar and video collected after the capture event has occurred. This data syncing is done programmatically in post-processing. Future safety systems will need a "hard real-time" clock system to ensure the correct detection and data fusion between both systems.

5.1.2 Video Radar Frame Syncing Implementation

Typical detection algorithms used in traditional image and video processing can be limited due to their computational requirements. Graphics cards have advanced processing of videos to allow speeds well above 30 frames per second (FPS) and much greater resolutions while maintaining high accuracy. This system was not designed for live data acquisition and was chosen to use the limits of the AWR1642 capture speed. As previously mentioned in Chapter 5, radar frame data was set to

run with an exact periodicity of 50ms. At this rate, the video captured during the test had its FPS set downsampled to 19.2[1]. In the GUI application created for this thesis, the radar frames set the limit. Future work will need to verify the lowest FPS between the the radar data and video feed to correctly match each frame.

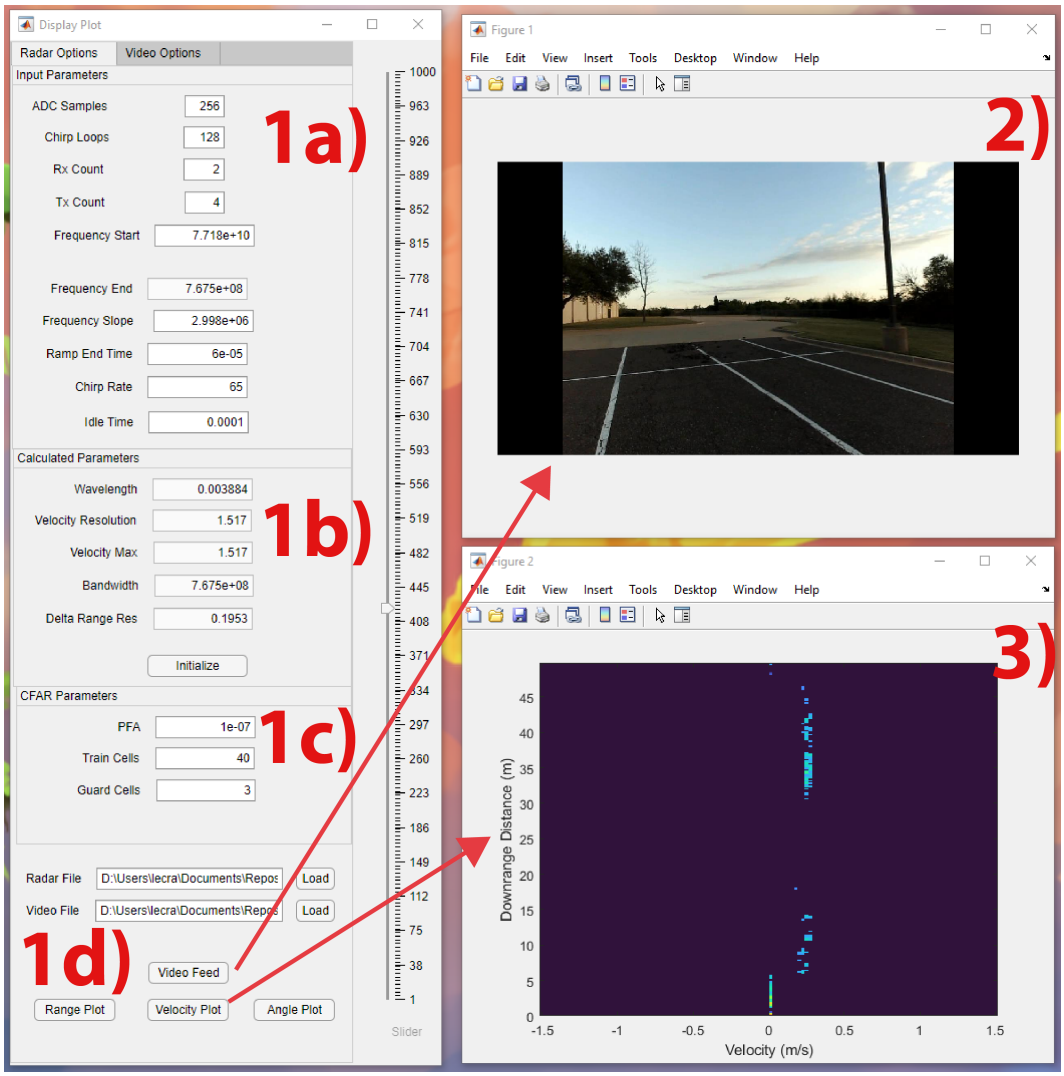


Figure 5.1: The GUI allows a user to modify parameters(1a), view the calculated parameters (1b), modify the CFAR Window(1c), user inputted radar and video files (1d). Additionally plots can be generated and scaled independently for viewing (2) and (3).

[1]The original video capture rate set to 30FPS and 720p

5.2 CFAR

Traditional standard radar threshold detection schemes assume that a known noise level is constant. Utilizing the constant false alarm rate (CFAR) detection algorithm allows an adaptive threshold where interference levels are variable. Further details on the implementation of this algorithm can be found in reference [8]. Unless otherwise stated, figures showing range-Doppler utilized CFAR to clean up noise. A Matlab implementation of the CFAR algorithm can be found in Appendix **??**.

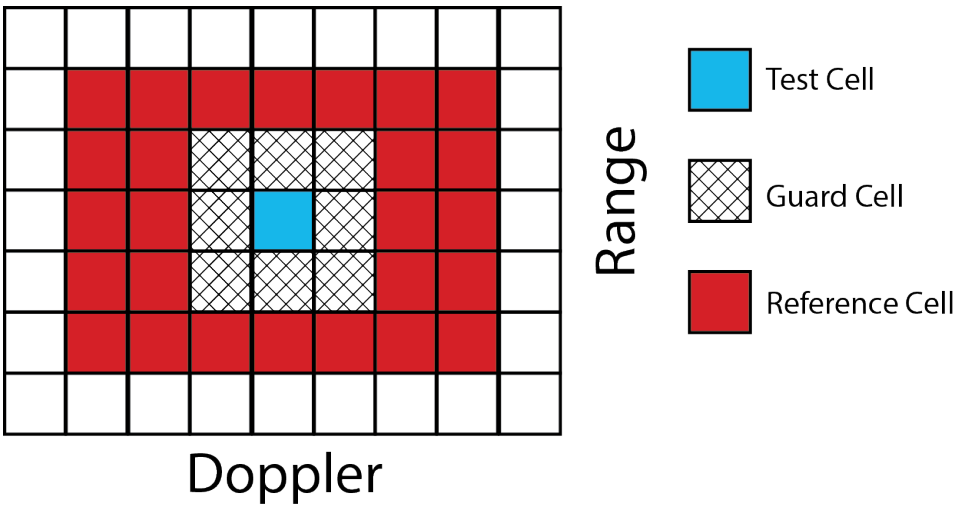


Figure 5.2: Two-dimensional window used in CFAR processing on range-Doppler plots.

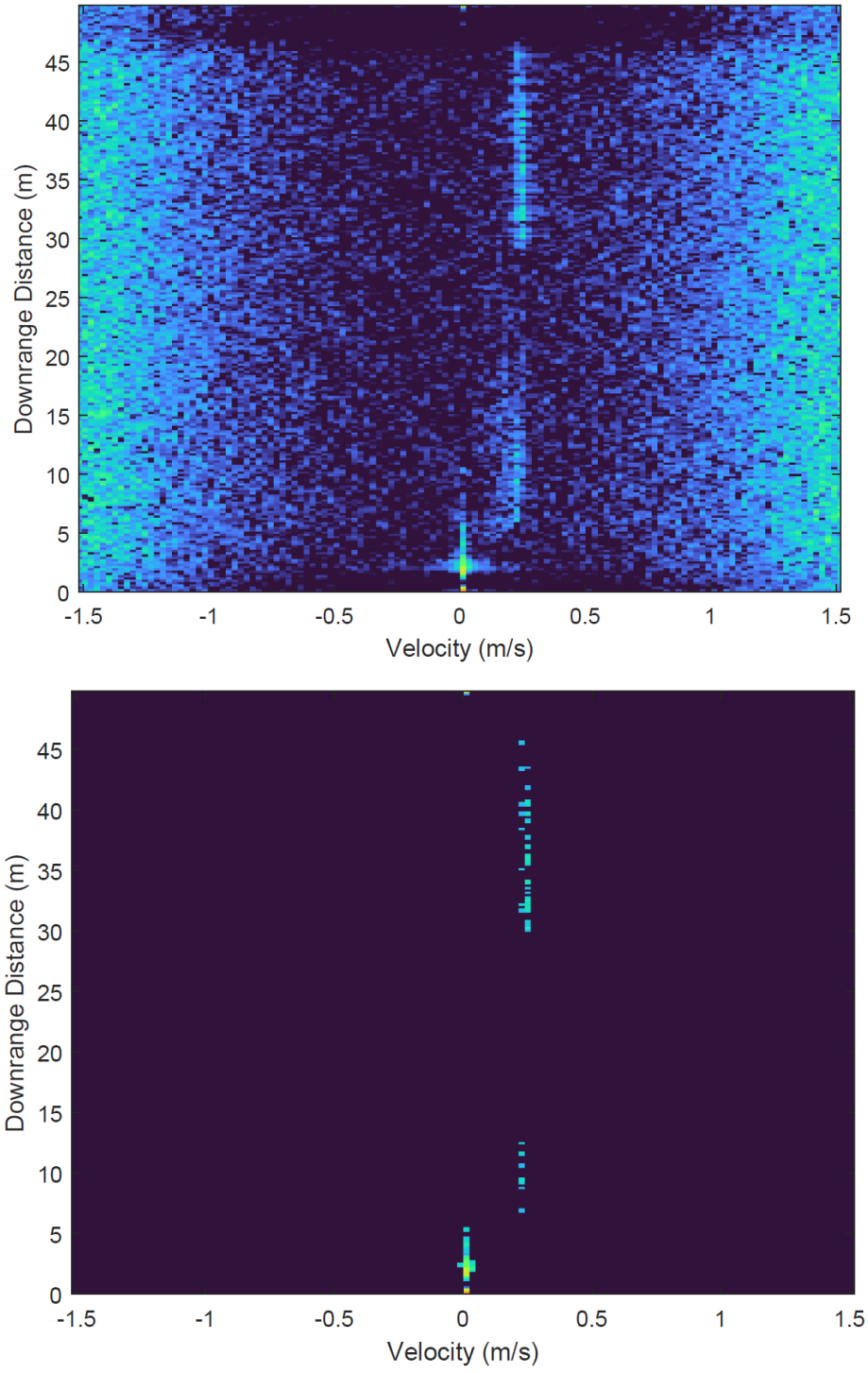


Figure 5.3: Top) Down-range velocity plot with no CFAR post-processing. Bottom) Down-range velocity plot with added CFAR algorithm. Noise is all but eliminated.

Chapter 6

Data Collection Results

6.1 Data Capturing

This section highlights canonical scenarios from the data captured during testing of the system. Testing of the capture device varied across stationary and moving tests. Early tests were performed using the tripod only with the proposed system in Chapter 2. Later tests moved the system to a car and took measurements while driving. The aim was to capture a wide variety of scenes and objects commonly found in an automotive environment. Locations included neighborhoods, local city streets, country roads, and parking lots. Potential classification objects collected in this data set include moving and stationary cars, pedestrians, animals, signs, and potholes. The raw ADC data taken using the AWR1642EVM and video data will be available upon request for future post-processing efforts. This experimental campaign provided a diverse data set for future work in fusion and artificial intelligence/machine learning (AI/ML) research and development.

6.1.1 Driving: Country Road and Fence Posts

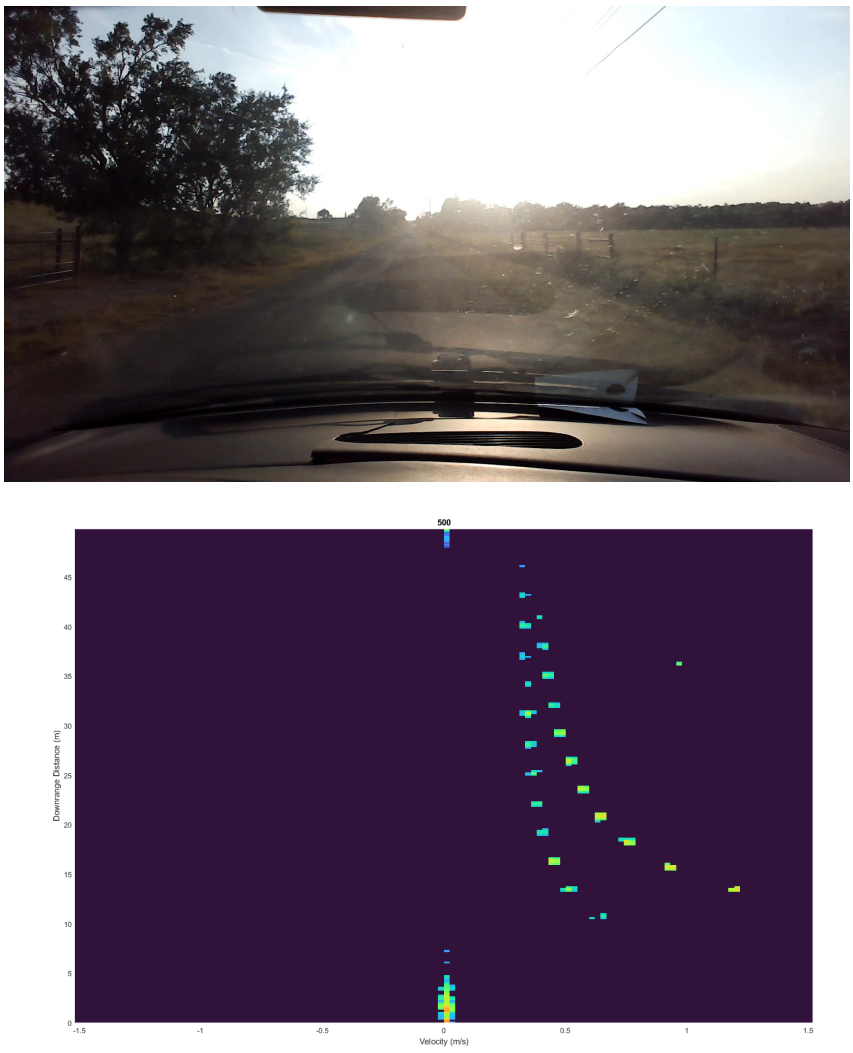


Figure 6.1: Early sample of radar taken inside vehicle on country road. Fence posts on both side of the road a "curved" sequence of returns in the range-Doppler.

6.1.2 Stationary: Person Walking Dog in Neighborhood

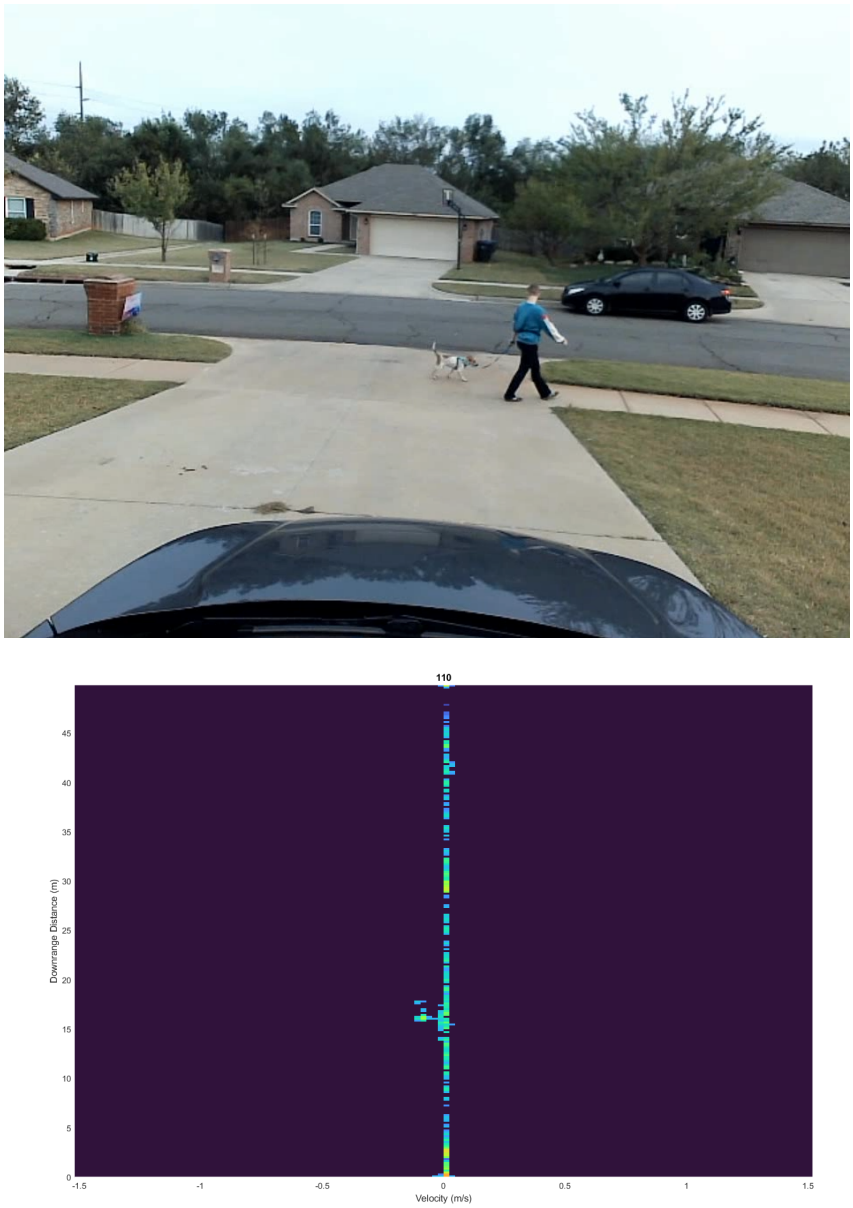


Figure 6.2: A stationary shot of a person and dog walking on the sidewalk.

6.1.3 Stationary: Handicap Sign in Parking lot

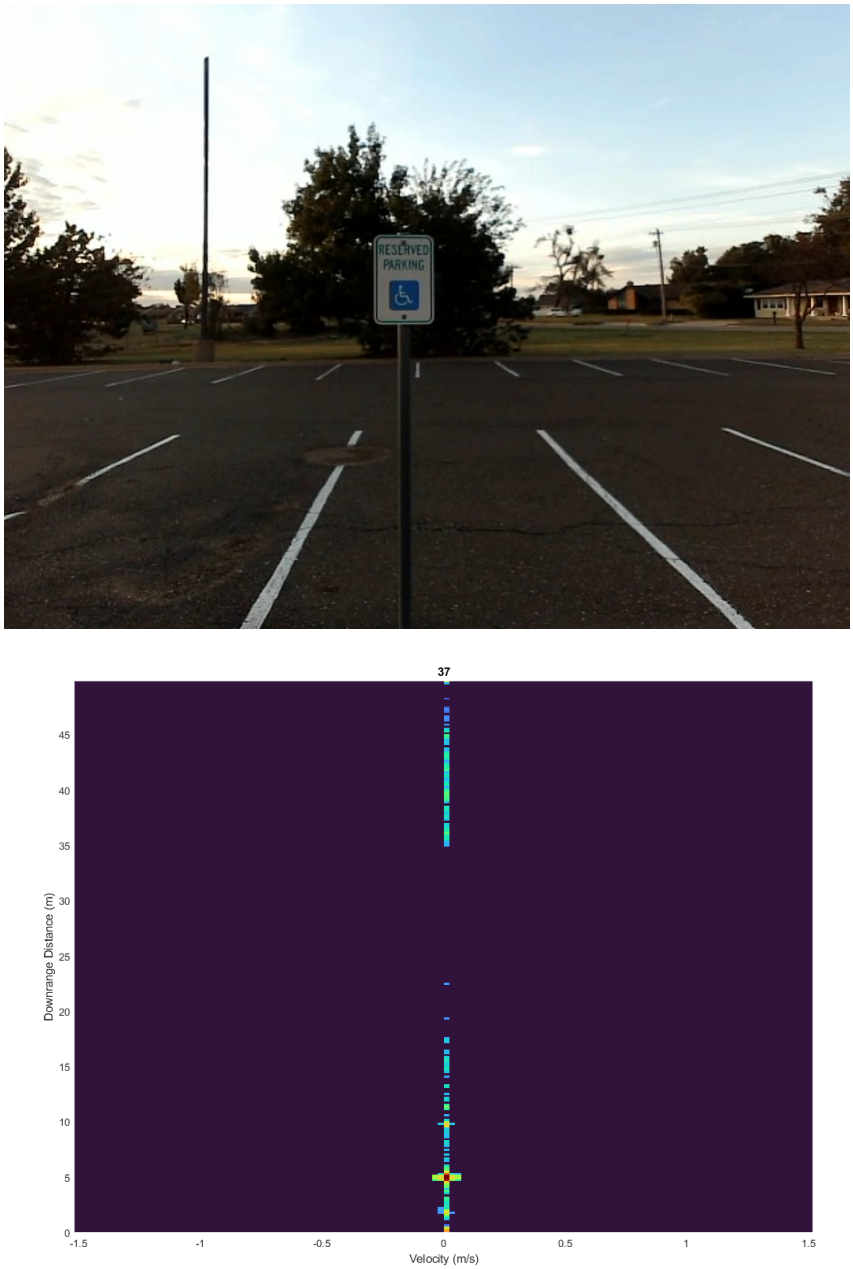


Figure 6.3: A stationary shot of a Handicap sign in parking lot. Notice the high RCS return of the sign.

6.1.4 Driving: Debris and Pothole in Parking Lot

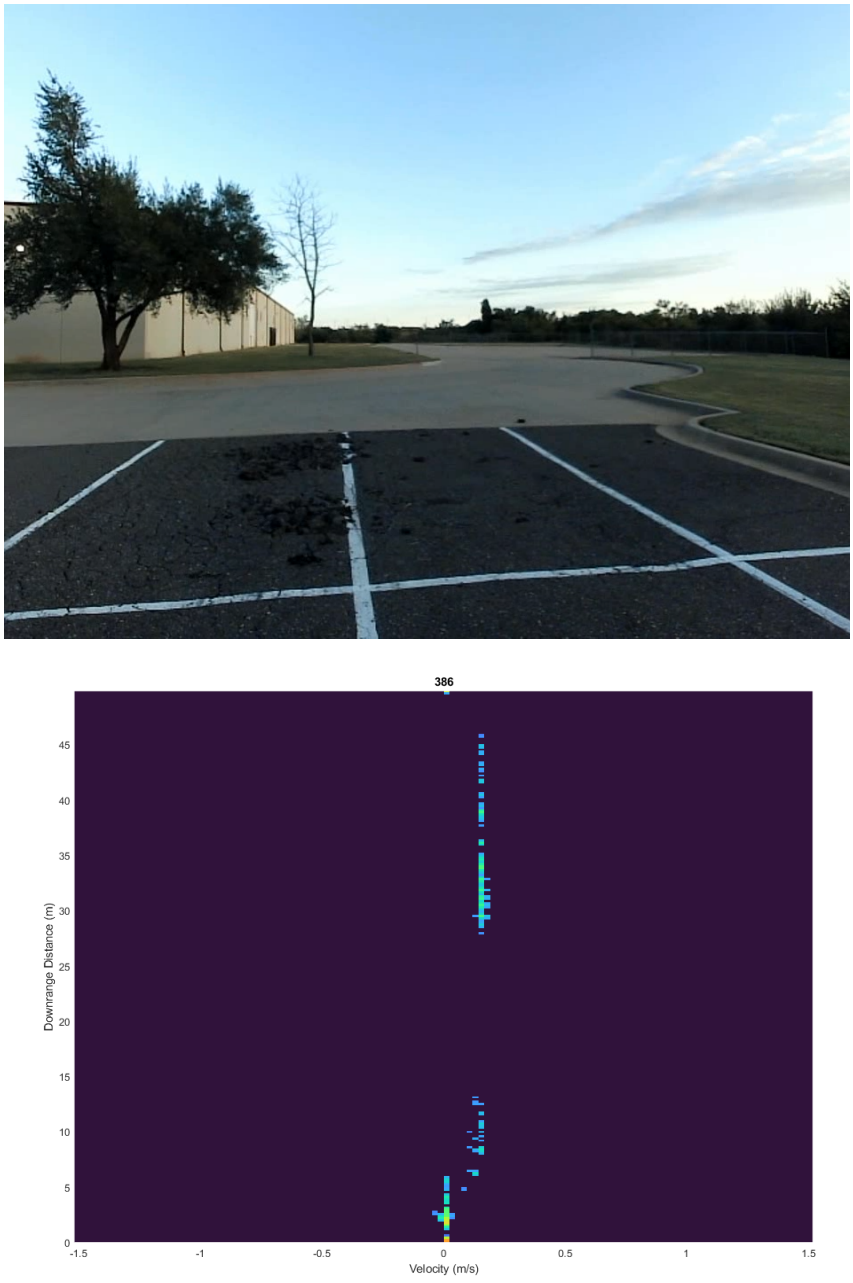


Figure 6.4: Moving radar capture of pothole and debris in a parking lot. Notice the gap after the debris. The next detection is the tree and curb across the street.

6.1.5 Driving: Person Walking on Side Walk next to Street

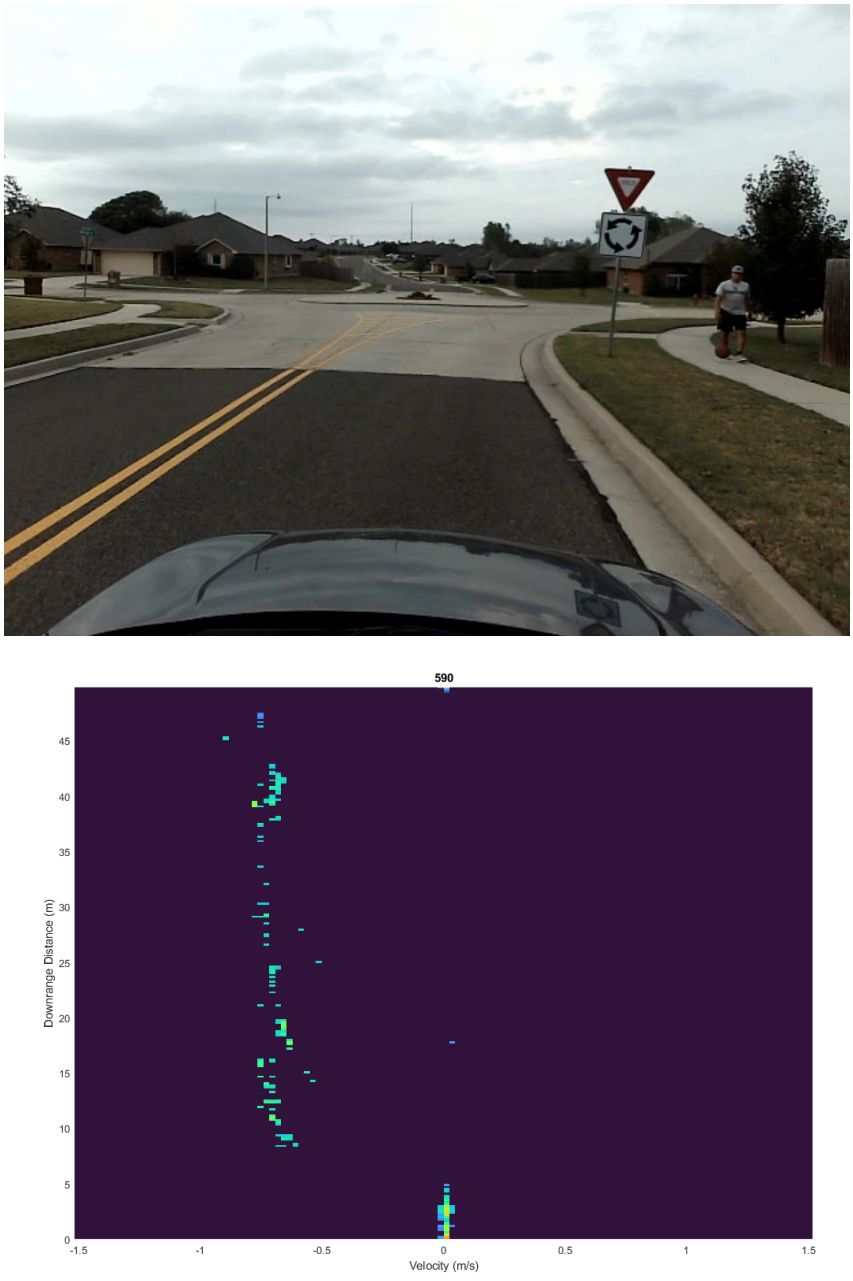


Figure 6.5: View of a neighborhood road.

6.1.6 Driving: Approaching Stop Sign in Neighborhood

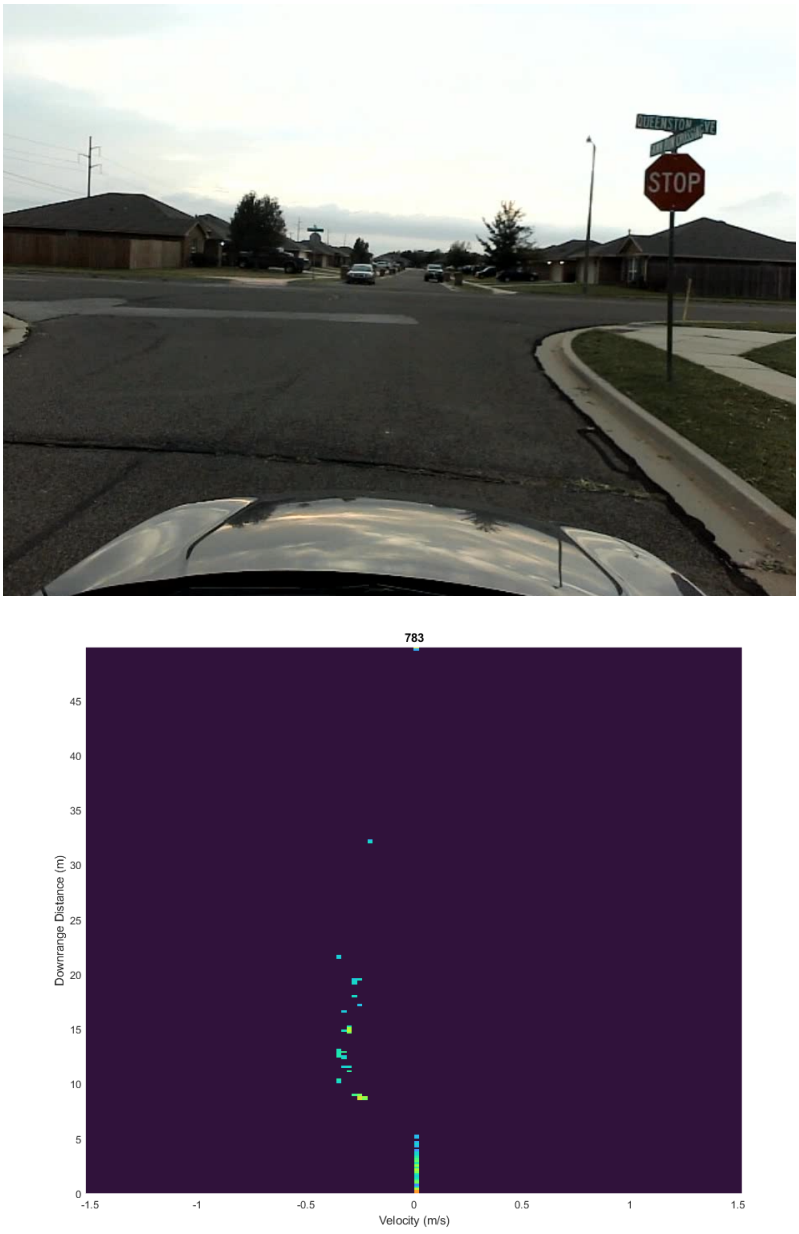


Figure 6.6: View of car approaching stop sign.

6.1.7 Stationary: Waiting at Stoplight with Passing Vehicles

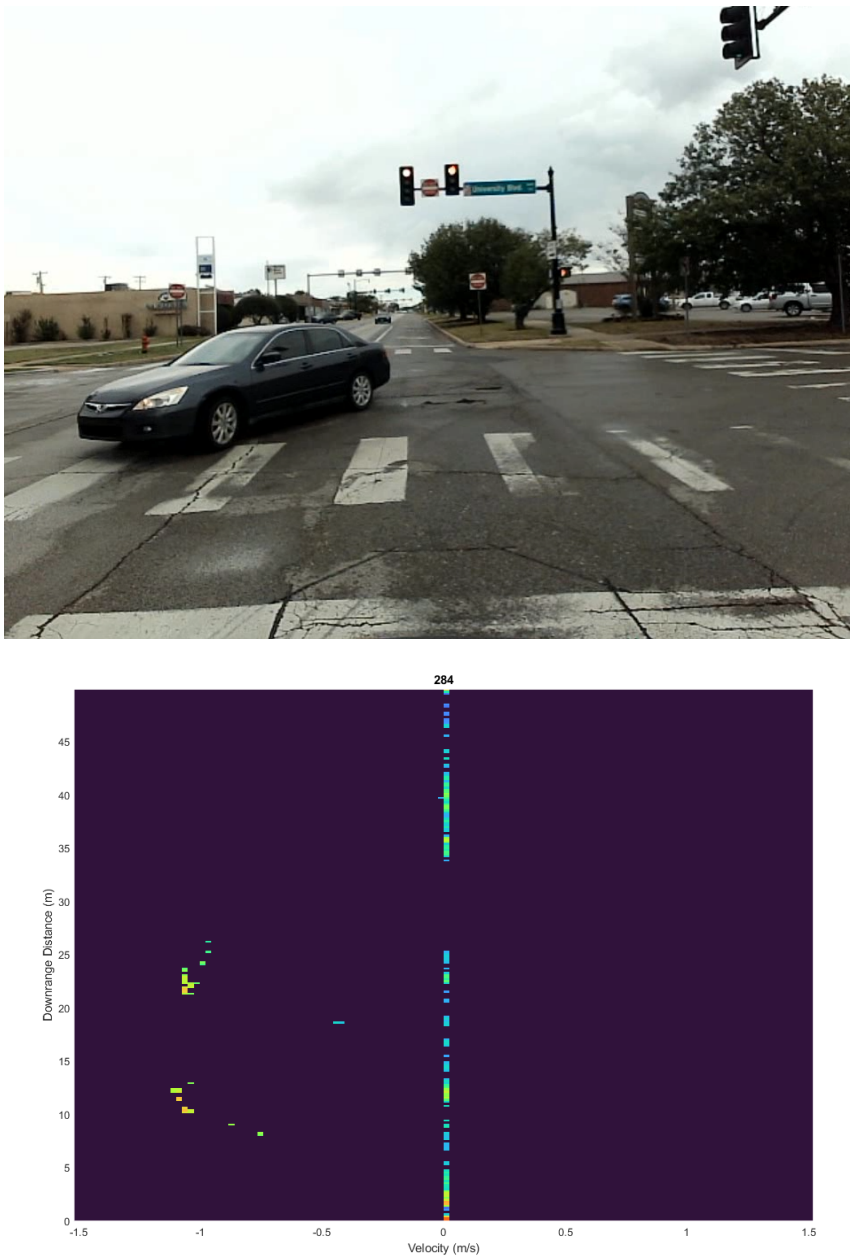


Figure 6.7: Scene of radar capture at a traffic light with passing cars.

6.1.8 Driving: Busy Parking Lot with Moving and Stationary cars



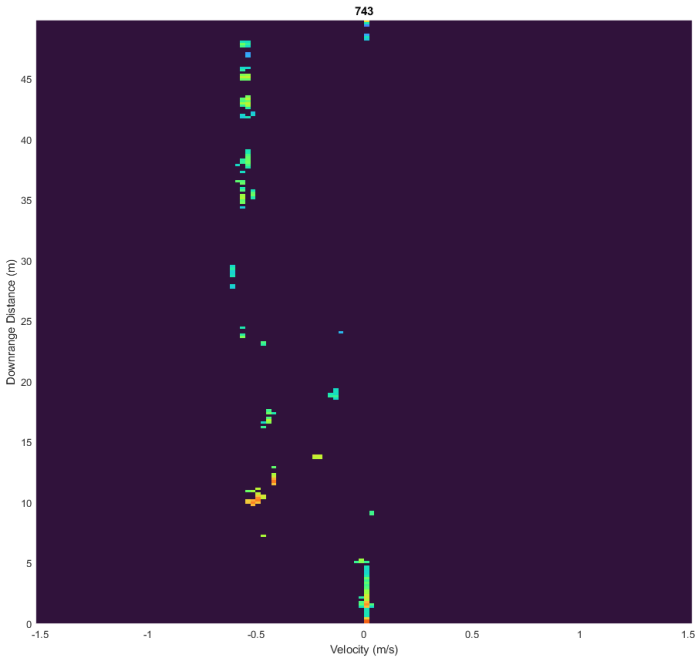


Figure 6.8: A full parking lot with a large pot hole.

Chapter 7

Conclusions and Future work

The future of automotive radar engineering is becoming a growing field in research. Enhanced safety systems and self-driving cars are actively being integrated into vehicles today. Future capabilities, including radar-based object detection and fusion with video, require sophisticated algorithms and system development. With the push to move into 77GHz radar by regulatory bodies, the TI AWR1642EVM proved a great and inexpensive data capture system.

This thesis covered the development of a deployable automotive radar toolbox. The choice to use TI's AWR1642Boost series proved to be a low-cost solution with significant expansion for collaboration in the larger automotive radar community. Creating a modified case that allowed the colocation of a video camera allowed for dynamic testing in moving vehicles. Further expansion of post-processing and data synchronization using the Matlab-developed GUI will support future automotive radar fusion activities.

7.1 Future Work

7.1.1 Proposed System Case Upgrades

Chapter 2 described a custom 3D printed case used to mount and protect the xWR board series with a DCA1000 . The CAD files for the case and dovetail mount will be made available for 3D printing. Future work to modify the case should focus on the protection of the board. Securely holding the USB ports like the power and Ethernet port could prevent the wires from entering the system. While the onboard radar array is convenient for testing, it is not easy to protect the radar without affecting radar performance. A fully encapsulated design will require a radome of some kind to be used by the device. TI has complete documentation on this topic for future revisions of a carrying case[29]. Any changes to the system may also require future software updates to compensate for changes to transmission and receive capabilities of the radar.

7.1.2 Expanded Operations

The "Cascading Mode" option uses a TI TDA2XX series operates as a bus for master or slave modes for the xWR series. This feature could be exploited to provide extended capabilities for interference testing, advanced beam-forming techniques, and MIMO radar applications. More information can be found on TI's website [30].

7.1.3 mmWaveStudio Automation

mmWaveStudio uses the high-level programming language Lua internally for scripting and automation. Considerable effort to automate a capturing system for a

unified video and radar system never made it to a fieldable system while working on this thesis. Scripts include all commands manually entered into the *mmWaveStudio* and range from connecting to an xWR board to radar configuration.

TI released a Matlab library allowing for Lua command injections into *mmWaveStudio*. In order to synchronize all control, test scripts were made in Matlab to run *mmWaveStudio*. Additional scripts written allowed video capturing using Matlab. With both combined features, a single script could simultaneously capture video and radar data. This feature had success on the desktop described in Table 4.1 but was unsuccessful on the laptop described in Table 2.3 due to limited RAM. Reimplementation of this feature seems feasible but requires additional testing and optimization.

References

[1] I. Bilik, "Introduction to automotive radars," in *2021 IEEE Radar Conference*, 2021.

[2] H. H. Meinel, "Evolving automotive radar — from the very beginnings into the future," in *The 8th European Conference on Antennas and Propagation (EuCAP 2014)*, 2014, pp. 3107–3114.

[3] K. Ramasubramanian, K. Ramaiah, and A. Aginskiy, "Moving from legacy 24ghz to state-of-the-art 77 ghz radar," Texas Instruments, Tech. Rep., 2017.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once:unified, real-time object detection," in *CVPR*, 2016. [Online]. Available: https://arxiv.org/pdf/1506.02640v5.pdf

[5] K. Ramasubramanian, "Using a complex-baseband architecture in fmcw radar systems," 2017.

[6] E. Cole, *Performance of LVDS With Different Cables*.

[7] *MMWAVE SDK User Guide*, Texas Instruments, Oct. 2018.

[8] M. A. Richards, *Fundamentals of Radar Signal Processing*, M. McCabe, Ed. McGraw-Hill Education, 2005, vol. 2.

[9] S. Rao, "Mimo radar," Texas Instruments, Tech. Rep., 2018.

[10] H. Griffiths, P. Knott, and W. Koch, "Christian hülsmeyer: Invention and first demonstration of radar, 1904," *IEEE*, 2018.

[11] C. Waldschmidt, J. Hasch, and W. Menzel, "Automotive radar-from first efforts to future systems," *IEEE Journal of Microwaves*, 2021.

[12] Reuters, "Greyhound's radar move," Newspaper/Digitized, Apr. 1992.

[13] I. Bilik, "Introduction to autmotive radars," *IEEE Radar Conference*, 2021.

[14] T. Levin, "Tesla's full self-driving tech keeps getting fooled by the moon, billboards, and burger king signs," Digital, July 2021, https://www.businessinsider.com/tesla-fsd-full-self-driving-traffic-light-fooled-moon-video-2021-7. [Online]. Available: https://www.businessinsider.com/tesla-fsd-full-self-driving-traffic-light-fooled-moon-video-2021-7

[15] *EQUIPMENT AUTHORIZATION GUIDANCE FOR 76-81 GHz RADAR DEVICES*, FCC.

[16] SAE, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," SAE, Tech. Rep., 2021.

[17] D. Wakabayashi, "Self-driving uber car kills pedestrian in arizona, where robots roam," Digital, March 2018, https://www.nytimes.com/2018/03/19/technology/uber-driverless-fatality.html. [Online]. Available: https://www.nytimes.com/2018/03/19/technology/uber-driverless-fatality.html

[18] Y. Wei, Y. Zhang, Z. Xu, and X. Li, "Classification of pedestrian motion based on micro-doppler feature with lfmcw radar," in *2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*, Dec 2019, pp. 1–5.

[19] *AWR1642 Evaluation Module Single-Chip mmWave Sensing Solution User's Guide*, Texas Instruments, 2020. [Online]. Available: https://www.ti.com/tool/AWR1642BOOST#tech-docs

[20] *DCA1000EVM Data Capture Card User's Guide*, Texas Instrucments, May 2019.

[21] *AWR1642 Single-Chip 77- and 79-GHz FMCW Radar sensor*, TI, 2017.

[22] *Interface Circuits for TIA/EIA-644 (LVDS)*, Texas Instruments, Sep. 2002.

[23] kalvik, "mmwave radar iwr/awr boost, dca1000 and camera," tI mmWave Radar IWR/AWR BOOST, DCA1000 and Camera/Kinect Case/Mount by kalvikis licensed under the Creative Commons. [Online]. Available: https://www.thingiverse.com/thing:3780598

[24] *Mmwave Radar Device ADC Raw Data Capture*, TI, Oct. 2018.

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

[26] S. Rao, *Introduction to mmwave Sensing: FMCW Radars*. [Online]. Available: https://training.ti.com/intro-mmwave-sensing-fmcw-radars-module-1-range-estimation

[27] S. Machad and S. Mancheno, “Automotive fmcw radar development and verification methods,” Master’s thesis, Chalmers University of Technology and University of Gothenburg, 2018.

[28] G. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*, ser. Real-Time Systems Series. Springer US, 2011. [Online]. Available: https://books.google.com/books?id=h6q-e4Q_rzgC

[29] C. Kumar, H. U. R. Mohammed, and G. Peake, “mmwave radar radome design guide,” 2021.

[30] *mmWave Studio Cascade*, TI. [Online]. Available: https://software-dl.ti.com/ra-processors/esd/MMWAVE-STUDIO/latest/exports/mmwave_studio_cascade_user_guide.pdf