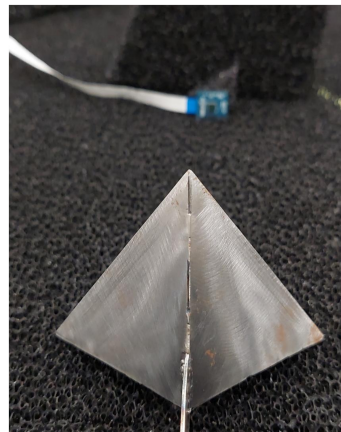
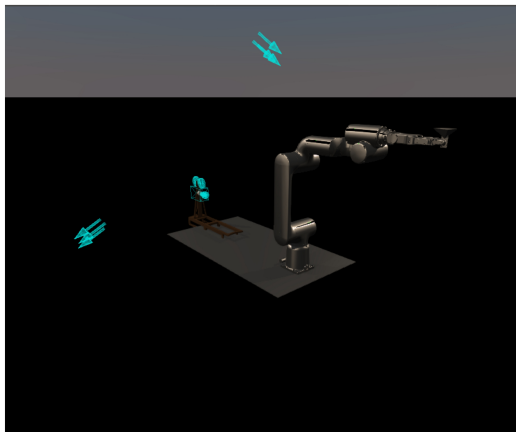
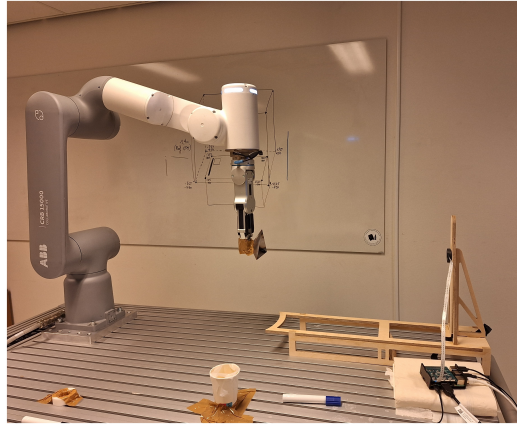
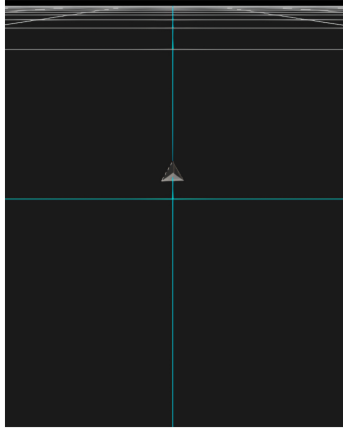




CHALMERS
UNIVERSITY OF TECHNOLOGY



Virtual Radar Simulations for Interior Sensing

Master's thesis in Information and Communication Technology

SURYA NAGA MARUTHI RAMYA MALLIPUDI VENKATA SATYA

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

MASTER'S THESIS 2024

Virtual Radar Simulations for Interior Sensing

A Master of Science thesis in the Information and Communication
Technology Master's Programme

SURYA NAGA MARUTHI RAMYA MALLIPUDI VENKATA
SATYA



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024

Virtual Radar Simulations for Interior Sensing
A Master of Science thesis in the Information and Communication Technology Master's Programme (MPICT)
SURYA NAGA MARUTHI RAMYA MALLIPUDI VENKATA SATYA

© SURYA NAGA MARUTHI RAMYA MALLIPUDI VENKATA SATYA, 2024.

Supervisor: Filip Johansson, Volvo Cars
Examiner: Prof Vessen Vassilev, Department of Microtechnology and Nanoscience

Master's Thesis 2024
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg

Virtual Radar Simulationd for Interior Sensing
SURYA NAGA MARUTHI RAMYA MALLIPUDI VENKATA SATYA
Department of Electrical Engineering
Chalmers University of Technology

Abstract

This master's thesis investigates the validation of radar simulation data for in-cabin sensing applications, utilizing AVxcelerate by Ansys to replicate and test real radar scenarios virtually. The research primarily focuses on comparing virtual radar readings from the simulation environment with real radar measurements gathered in a lab setting, aiming to ensure accuracy and consistency in radar data for future in-cabin safety applications. In this study, a controlled environment was established to minimize discrepancies between simulated and real-world conditions, with experiments conducted on a single target in two scenarios: stationary and controlled motion. Data from both settings were analyzed to assess the reliability of the simulation for representing real-world radar behavior. The findings contribute foundational insights for advancing radar-based occupant monitoring systems, supporting features like driver drowsiness detection and child presence alerts in automotive interiors.

Keywords: Radar Simulation Software, Real Radar Sensor, Real Radar Data, Lab Environment, Simulation Environment, Virtual Radar Data, Data Comparison.

Acknowledgements

First and foremost, I would like to express my heartfelt gratitude to my manager, Erik Liedholm, for providing me with the invaluable opportunity to work in a field I am deeply passionate about and for enabling me to collaborate with an amazing team. I am especially thankful to my supervisor, Filip Johansson at Volvo Cars, whose expert advice, unwavering support, and guidance were instrumental in the successful completion of this thesis. I am also deeply grateful to my examiner, Prof. Vessen Vassilev, for his timely support, constructive feedback, and encouragement, which were crucial to my progress. Their guidance and insights have been essential to my progress.

I would also like to extend special thanks to Stefan Axelsson from Volvo's Manufacturing Technology Development team for granting access to the industrial robot, a key component of this thesis, and for his guidance in operating it. My sincere appreciation and gratitude go to Sara Tell from Volvo's Radar team, whose expertise and unwavering support, combined with Prof. Vessen's expert advice on radar systems, greatly aided me in navigating the radar aspects of this project.

I am deeply thankful to Kaustubh Deshmukh from Enabling and Exploration Labs at Volvo and Ranjan Manohar from Chalmers for their assistance with CAD work. Special thanks also go to my brother, Pramod Mallipudi from RWTH Aachen; Arun Maslekar from ABB, Västerås; Abishek Swaminathan from Chalmers; Anand Jagadeesh from Volvo for their valuable insights and advice.

My heartfelt appreciation extends to the entire Internal Perception team at Volvo Cars for their encouragement, support, and camaraderie, which created an inspiring and enjoyable work environment. Every team activity—from lunches to fika to after-work gatherings—made this journey truly memorable. I would also like to express my gratitude to Vinayak Killedar and Mohammed Saif from Ansys for their invaluable support in using Ansys AVx. Additionally, I am grateful to the team at Enabling and Exploration labs at Volvo Cars, Prof. Per Nyqvist, Frederick Rexton, Ravindu Piyapema, Rob Vissers from the Microtechnology and Nanoscience group, as well as Zhaorui Zang from the Antenna Research group at Chalmers for their timely guidance and assistance.

Finally, I owe my deepest thanks to my parents and friends, whose unwavering support and encouragement have been the foundation of my success.

Surya Naga Maruthi Ramya Mallipudi Venkata Satya,
Gothenburg, November 2024

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ACC	Adaptive Cruise Control
ADC	Analog to Digital Converter
ADAS	Advanced Driver Assistance Systems
AEB	Automatic Emergency Braking
AVx	AVxcelerate by Ansys
BSD	Blind Spot Detection
CPI	Coherent Processing Interval
CW	Continuous Wave
FCW	Forward Collision Warning
FMCW	Frequency Modulated Continuous Wave Radar
IFFT	Inverse Fast Fourier Transform
LCA	Lane Change Assist
LiDAR	Light Detection and Ranging
LRR	Long Range Radar
MRR	Medium Range Radar
PDC	Park Distance Control
PRF	Pulse Repetition Frequency
PRI	Pulse Repetition Interval
RADAR	Radio Detection and Ranging
RCS	Radar Cross Section
RF	Radio Frequency
SNR	Signal to Noise Ratio
SRR	Short Range Radar
TCP	Tool Center Point
UHF	Ultra High Frequency
ULRR	Ultra Long Range Radar

Nomenclature

Below is the nomenclature of parameters and variables that have been used in this thesis.

Parameters

γ	Reflectivity
σ	Radar Cross Section
c	Speed of Light
ϕ	Phase
$\Delta\phi$	Phase shift
τ	Pulse Width
P_r	Received power
P_t	Transmit power
f_b	Beat frequency
f_r	Range dependent frequency shift
f_d	Doppler frequency shift
η	Efficiency
D	Directivity
N	Noise Power

Variables

ϵ	Relative permittivity
f	Frequency
B	Bandwidth
v	Radial Velocity
λ	Wavelength

G_t	Gain at transmitter
G_r	Gain at receiver
A	Amplitude
t	Time

Contents

List of Acronyms	ix
Nomenclature	xi
List of Figures	xv
List of Tables	xvii
1 Introduction	1
1.1 Automotive Radar	1
1.2 Aim	2
1.3 Scope	2
2 Theory	3
2.1 RADAR Basics	3
2.2 RADAR Classification by detection range	5
2.3 RADAR Classification by Waveform	8
2.4 Key Radar Principles	11
2.5 Acconeer A111 Radar Sensor	19
2.6 ABB CRB 15000	21
2.6.1 Key Features	21
2.7 Wizard Easy Programming	22
2.8 ABB Robotstudio	24
3 Method	27
3.1 Real Radar Data	27
3.1.1 Case 1- Stationary Target	27
3.1.2 Case 2- Target in Motion	30
3.2 Simulated Radar Data	36
3.2.1 Case 1- Stationary Target	36
3.2.2 Case 2- Target in Motion	40
3.2.2.1 Getting ABB CRB 15000 ready for simulation	41
4 Results	49
4.1 Results from Case 1- Stationary Target	49
4.2 Results from Case 2- Target in Motion	51
4.2.1 Results using Sparse format	51

4.2.2	Results using Envelope format	54
4.2.2.1	Method 1	54
4.2.2.2	Method 2	58
5	Conclusion and Future Work	65
5.1	Conclusion	65
5.2	Evaluation and Findings	65
5.3	Future Work	66
	Bibliography	69

List of Figures

2.1	Elements of a Basic Radar System [5]	5
2.2	Classification of Radars	11
2.3	A111 Radar chipset [4]	20
2.4	A111 functional block diagram [3]	20
2.5	ABB CRB 15000 Cobot [8]	22
2.6	Wizard Easy Programming on a controller	24
2.7	Wizard Easy Program Snippet- Multilingual	24
3.1	Static Corner Reflector in a Make-shift Anechoic Chamber	29
3.2	Corner Reflector of dimensions 10 cm × 10 cm × 10 cm	29
3.3	Corner Reflector placed at approximately 23.5 cms from the Radar	29
3.4	The Corner Reflector and the A111 Radar Sensor	30
3.5	Setup for physical measurements for target in motion	31
3.6	Signal Analyzer window from ABB Robotstudio	34
3.7	Corner reflector's motion in Envelope format	35
3.8	Sparse service format [1]	36
3.9	Corner's reflector's motion in Sparse format	36
3.10	A111 Sensor parameters in AVx Sensor Labs	38
3.11	A111 Sensor parameters in AVx Sensor Labs	38
3.12	Sensor Layout for static corner reflector in AVx Sensor Labs	39
3.13	Stationary corner reflector in AVx Asset Prep Environment	39
3.14	Static corner reflector as seen by the A111 radar in the simulation	40
3.15	Simulation output for stationary reflector	40
3.16	Joint states of CRB15000 captured by Signal Analyzer	41
3.17	ABB CRB 15000 in Blender	42
3.18	Local joint angles in a robot arm [11]	43
3.19	A111 sensor's view in a simulation frame	45
3.20	Simulation scene from AVx	46
3.21	Blender flowchart	47
4.1	Radar Range plot from Real Radar Data	49
4.2	Radar Range plot from Simulated Data	50
4.3	Radar Range plots from Simulated Data and Measurements for Stationary corner reflector	51
4.4	Sparse result from physical measurements	54
4.5	Sparse result from AVx simulation	54
4.6	AVx Simulation Raw output for target in motion	55

4.7	First frame from the real motion	56
4.8	First frame from the simulated motion	56
4.9	Inverse Fourier Transform plot for Pulse 1 in Frame 1 of Simulation .	57
4.10	Frame versus Range plot- Simulation	58
4.11	Frame versus Range plot- Real motion	59
4.12	Sensor Metadata Information from real motion recording	60
4.13	Sensor layout for target in motion with adjusted radar position	62
4.14	Frame versus Range plot- Simulation	63
4.15	Range versus Frame plots- Simulation v/s Real motion	64

List of Tables

2.1	System parameters of the radar sensor [4].	19
3.1	Axes mapping from Blender to AVx	42

1

Introduction

With the growth of advanced driver-assistance systems (ADAS) and autonomous vehicles, radar technology has become essential in the automotive sector. Traditionally, radar and LiDAR sensors have focused on imaging, sensing, tracking, and identifying obstacles around the vehicle during navigation. However, a promising new application is using radar sensors inside the vehicle to enhance passenger safety and comfort. Interior radar sensing could detect the presence of passengers, alert to the presence of a child left in the vehicle, or monitor the driver's alertness to prevent drowsiness. These innovative uses expand radar's role from external safety monitoring to ensuring a safer, more responsive in-cabin environment.

1.1 Automotive Radar

Radar, short for Radio Detection and Ranging, was originally developed for military applications, where it remains extensively used. Today, radar technology has expanded into various civilian applications, including air traffic control, weather monitoring, environmental sensing, ship and aircraft navigation, industrial speed measurement, space surveillance, and planetary observation. In the automotive industry, radar plays a crucial role in both exterior and interior sensing, supporting advanced driver-assistance systems (ADAS), autonomous navigation, and in-cabin monitoring.

Initially, automotive radar operated in the 24 GHz range, but it has since shifted to the 77 GHz band, enabling high-precision functions like Blind Spot Detection (BSD), Lane Change Assist (LCA), and Adaptive Cruise Control (ACC). In addition to these external sensing capabilities, radar is now being applied to monitor the vehicle's interior, detecting occupants and alerting for potential hazards, such as a child left unattended or a drowsy driver. This novel application of radar for in-cabin sensing enhances passenger safety and driver awareness.

Together with LiDAR and cameras, radar forms an essential part of a vehicle's sensor suite, offering robust performance in poor weather and low-light conditions, where other sensors may be limited. Furthermore, radar is a cost-effective solution compared to LiDAR and high-resolution cameras, making it a versatile and valuable technology for comprehensive vehicle safety, both inside and out.

1.2 Aim

The objective of this Master's thesis is to validate simulated radar data for in-cabin sensing applications, forming a foundational basis for future occupant monitoring developments, such as driver drowsiness detection, seat belt status monitoring, and passenger presence detection. Using AVxcelerate by Ansys, this research will simulate a virtual lab environment to capture radar readings, which will be systematically compared to real radar data collected under equivalent lab conditions. By aligning simulated and real-world data, this study aims to enhance the accuracy and reliability of virtual testing environments, facilitating further advancements in occupant monitoring systems for improved vehicle safety.

1.3 Scope

The scope of this thesis focuses on evaluating the accuracy of AVxcelerate by Ansys (AVx) as a radar simulation tool, using real radar measurements as a benchmark. To minimize discrepancies, both simulation and real-world lab measurements were conducted in closely matched environments. This study is limited to the use of a single target with restricted motion, selected to establish a controlled baseline. Two scenarios were examined: one with a stationary target and a second, more complex scenario where the target moves in a controlled manner. The findings from both scenarios are analyzed to assess how closely the simulation results match real-world data under these initial conditions.

2

Theory

This chapter presents the theoretical foundations of the thesis, beginning with an overview of radar fundamentals, the core element of this project. It discusses radar components and their functions, essential concepts related to radar measurements, various types of radar sensors, and the specific radar type employed in this research. A brief section provides available details on the radar chipset used in the project. The chapter also introduces ABB RobotStudio and the robot, which serve as additional key components in the overall project setup.

2.1 RADAR Basics

Radar is an acronym for radio detection and ranging. The first successful attempt at producing radio waves was done at a frequency of 200MHz which as per today's classification of frequencies, falls in the Ultra High Frequency(UHF) band which is very close to the lower edge of the Microwave band(300MHz - 300GHz). Radar which dates back to the 1900s was first used to detect the presence of ships in 1903 by Hullsmeyer who obtained a patent for his idea the following year. Fast forward by a couple of decades, Pulsed Radar systems were used during World War II by all parties to detect enemy ships and aircrafts.

Radar is an active sensor that has both a transmitter and a Receiver. It's powerful penetration capabilities make it very suitable for long distance ranging and detection. In recent times, however, Radar has also found its use in short-range detection in applications like autonomous driving, robotics, industrial automation, and even smart devices. Fundamentally Radars can measure the following:

- **Direction of an object:**
This can be done by gauging the direction in which the antenna is pointed.
- **Distance or Range of an object:**
This can be estimated by measuring the time of flight of the of the signal to and from the target.
- **Speed of an object:**
This can be estimated by measuring the Doppler shift of the echo from the target.

- **Dimension of the object:**

This can be estimated by calculating the Radar Cross section (RCS) of the target by measuring the received power.

A Typical RADAR System would have the following components as also seen in Figure 2.1:

- **Transmitter:**

This unit produces signals/pulses that are coupled by the transmitting antenna into space. The waves are then reflected or scattered back to the Radar system by objects in free space. The receiving antenna receives this reflected energy and send it to the receiver circuit of the system.

- **Antenna:**

Antenna is required for the transmission of Electromagnetic waves from the Transmitter and to receive electromagnetic energy from the target.

- **Receiver:**

This unit amplifies the received electromagnetic energy from the target. Also, some form of correlation needs to be done between the transmitted and received signals in order to make sure that they can be compared properly.

- **Duplexer/ Transmit-Receive switch(TR Switch):**

In most Radar Systems, a single antenna is used for both Transmission and reception. The antenna switches between transmitting and receiving modes by using a duplexer or a TR Switch that makes sure that there is no mutual interference when the radar transmitter and receiver share the same antenna.

- **Signal Processor:**

To obtain details about targets the radar has spotted, the signal processor examines the signals it has received. It carries out operations including tracking, target identification, Doppler processing and pulse compression.

- **Display Unit:**

The display unit shows the Radar output in a human readable format.It includes Radar screens, maps and graphical user interface to visualize the detected targets.

With the basics of radar covered, the focus now shifts to exploring the types of radar sensors available and their classifications.

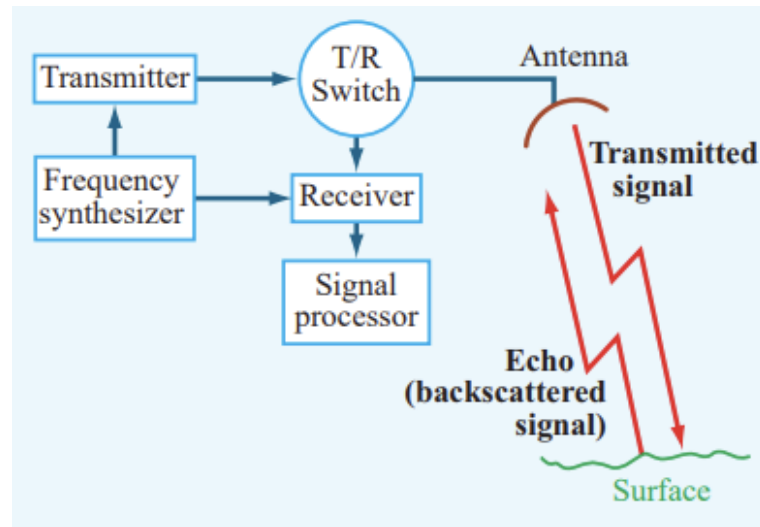


Figure 2.1: Elements of a Basic Radar System [5]

2.2 RADAR Classification by detection range

- **Short Range Radar:**

- **Detection Range:** It is very small, ranging from a few centimeters upto 30-50 metres.
- **Operating Frequency:** The usual frequency range is from 24GHz upto 77GHz with higher frequency ranges offering better precision.
- **Key Features:**
 - * Short Range Radar has a wide field of view (FoV) for detecting nearby objects from different angles.
 - * It provides high spatial resolution for detecting small or closely spaced objects.
 - * It covers short distances and is more focused on proximity detection.
- **Applications:**
 - * SRR is used in close object detection applications like industrial automation and Advanced Driver Assistance Systems (ADAS).
 - * It is also used Automatic parking and Park distance control(PDC) when parking or maneuvering in tight spaces.
 - * SRR can be used for Blind Spot Detection which is especially im-

portant during lane change where the Radar continuously monitors the vehicle's blindspot and alerts the driver if there is any vehicle or object in an adjacent lane.

- * Some other applications include collision avoidance in Autonomous robots operating in tight spaces, safe Human Robot interaction which includes stopping or slowing down the robot when in close proximity with humans to prevent injury.

- **Medium Range Radar:**

- **Detection Range:** It is slightly more than that of SRR, ranging between 50 meters to 150 meters.

- **Operating Frequency:** It is the same as that of SRR (24GHz to 77GHz). MRR sensors have good spatial resolution and a moderate field of view.

- **Key Features:**

- * MRR covers moderate distances and is optimized for tracking objects that are not too close but also not too far.
- * It has a narrower field of view compared to SRR, as it focuses on mid-range targets.
- * Resolution is moderate compared to both SRR and long-range systems, providing a balance between precision and distance.

- **Applications:**

- * MRR sensors are very important in Advanced Driver Assistance Systems and autonomous driving. They are used in conjunction with short range and long range radar sensors.
- * MRR is very useful in Adaptive Cruise Control(ACC). This sensor detects vehicles and objects at moderate distances and helps maintain a safe distance.
- * These sensors are also quite useful in Forward Collision Warning (FCW) and Automatic Emergency Braking (AEB) for mid range distances.
- * Apart from automotive applications, MRR is also used in robot automation for collision avoidance in heavy machinery by detecting objects at medium ranges.

- * MRR is also quite useful in enhancing a robot's perception of its environment, allowing it to make decisions based on the location of objects at medium distances.

- **Long Range and Ultra Long Range Radar:**

- **Detection Range:** The sensors can detect ranges from 150 metres to several hundreds of kilometers.
- **Operating frequency:** LRR sensors operate at 77GHz for automotive applications and at lower frequency bands like S band (1-2 GHz), C band (2-4 GHz) and L band (4-8 GHz) for aerospace applications.
- **Key Features:**
 - * LRR systems focus on distant objects and have a narrow field of view for long-range detection.
 - * These radars generally provide lower resolution compared to SRR and MRR because they prioritize long-distance detection.
 - * They are more expensive and complex, often requiring more power and sophisticated hardware.
- **Applications:**
 - * LRR is used for adaptive cruise control to maintain a safe following distance between two vehicles by detecting their speed and position at long distances.
 - * It is also used in collision avoidance and autonomous emergency braking by detecting potential obstacles ahead at long ranges, enabling timely responses.
 - * Apart from automotive applications, LRR and ULRR are used in defense applications like target tracking, surveillance and reconnaissance missions. They are also used in ship navigation.

2.3 RADAR Classification by Waveform

Figure 2.2 shows a flowchart showing the classification of Radar Systems.

Continuous Wave Radar:

Radar systems that continually and uninterruptedly emit Radio frequency (RF) waves are known as continuous wave (CW) radar systems. CW radar generates a continuous signal, in contrast to pulsed radar systems, which send brief bursts of RF energy followed by listening intervals. This type of Radar systems can be further classified as follows:

- **Frequency Modulated Continuous Wave Radar (FMCW):**

This kind of Radar continuously generates a signal whose frequency changes linearly with time. It measures both range and velocity by frequency modulation. FMCW Radar calculates the distance to a target by analyzing the beat frequency. Velocity is measured using Doppler shift, similar to CW Doppler radar. It is extensively utilised in navigation, altimetry, and distance measuring applications.

In FMCW Radar, velocity is measured by analyzing the **Doppler shift** superimposed on the range-dependent frequency shift. The transmitted signal is modulated (e.g., linear chirp), enabling simultaneous range and velocity measurements.

The total beat frequency f_b in FMCW radar is given by:

$$f_b = f_r + f_d, \quad (2.1)$$

where:

- f_b : Total beat frequency,
- f_r : Range-dependent frequency shift, proportional to the time delay,
- f_d : Doppler frequency shift, proportional to the velocity.

The Doppler shift f_d is extracted from the total beat frequency by separating the contributions of range (f_r) and velocity (f_d) using signal processing techniques.

Velocity calculation by an FMCW Radar:

1. The radar transmits a frequency-modulated signal, typically a linear chirp.
2. The received signal contains a time delay (related to range) and a Doppler shift (related to velocity).
3. The beat frequency f_b is analyzed to extract f_r and f_d .

4. The Doppler shift f_d is used to calculate velocity:

$$v = \frac{f_d \lambda}{2}. \quad (2.2)$$

Key Features:

- Measures both range and velocity simultaneously.
- Requires more complex signal processing compared to CW Doppler radar.
- Suitable for dynamic environments.

• **Continuous Wave Doppler Radar:**

A continuous radio frequency signal is sent and received constantly in a CW Doppler radar. The Doppler shift in frequency caused by the motion of targets within the radar beam is detected and analyzed for velocity estimation. No modulation is involved as it transmits a continuous wave of fixed frequency.

Velocity measurement using Doppler shift:

The velocity of a target can be measured using the **Doppler effect**, which causes a frequency shift in the radar signal when there is relative motion between the radar and the target. The frequency of the received signal increases if the target moves toward the radar and decreases if it moves away.

The Doppler shift f_d is directly proportional to the relative velocity v of the target along the radar's line of sight. It is given by:

$$f_d = \frac{2v}{\lambda} \quad (2.3)$$

where:

- f_d : Doppler frequency shift (in Hz),
- v : Radial velocity of the target (in m/s),
- λ : Wavelength of the transmitted signal (in meters).

Velocity calculation by a CW Doppler Radar:

1. The radar transmits a continuous wave of constant frequency.
2. The frequency of the reflected wave is compared to the transmitted frequency.
3. The Doppler shift f_d is calculated as the difference between the transmitted and received frequencies.
4. Using the formula for f_d in Equation 2.3, the velocity v can be determined.

Key Features:

- Directly measures the radial velocity.
- No range information is obtained.
- Simple and computationally efficient.

Pulsed Radar: A pulsed radar works by sending brief radio frequency (RF) energy bursts, or pulses, and then listens for reflections in the surroundings that the radar receiver picks up. Because pulsed radar systems can offer precise range measurements, high resolution, and interference mitigation, they are widely employed in many different applications. This is how a typical Pulsed Radar works,

- **Transmitting Pulses:** The radar system emits short bursts of electromagnetic energy, known as pulses, into the surrounding environment. The length of these pulses is usually quite brief, ranging from microseconds to milliseconds.
- **Listening Period:**
Following a pulse transmission, the radar system enters a listening mode and listens for echoes returned from nearby objects. The radar system's maximum range determines how long the listening period is; this gives the broadcast pulse enough time to reach its maximum range and back.
- **Receiving Echoes:**
The transmitted pulses from objects in the surrounding environment are picked up by the radar receiver as echoes or reflections. The distance to the target is determined using the speed of light and the time interval between the pulse's transmission and the echo it receives.
- **Signal Processing:**
Following the reception of the echoes, information about the targets, including their size, velocity, and occasionally direction, is extracted using signal processing techniques. Target tracking algorithms, Doppler processing, and pulse compression are a few examples of these methods.

Pulsed Radars can be of 2 types,

- **Coherent Pulsed Radars:**
In coherent pulsed radar systems, the transmitted pulses maintain a fixed phase relationship with each other and with the received signals.
- **Non Coherent Pulsed Radars:**
Phase coherence between the transmit and received signals is not necessary for the detection of non-coherent pulsed radar. Given this, these Radars are

used in applications where precise measurements of target range and velocity are not required.

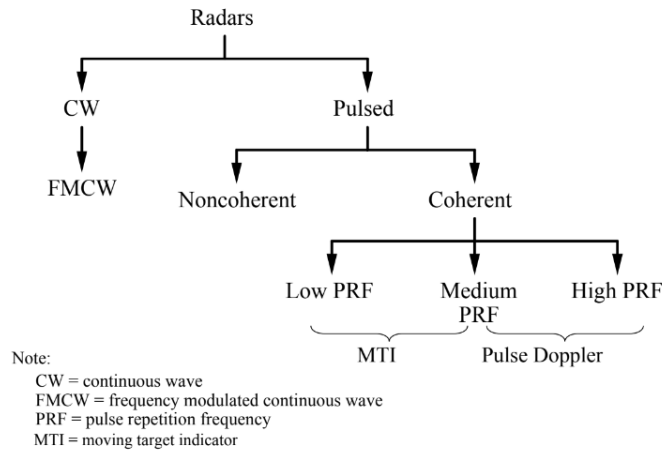


Figure 2.2: Classification of Radars

Now that the types of radar sensors and their operation have been discussed, the focus shifts to understanding some key radar principles.

2.4 Key Radar Principles

Reflectivity:

The object's reflectivity (γ), Radar Cross Section (RCS) (σ), and distance from the antenna all affect how much energy is returned to the antenna. When the relative permittivity of two mediums that the signal is travelling through differs, a reflection happens. Reflectivity (γ) is given by,

$$\gamma = \left(\frac{\sqrt{\epsilon_1} - \sqrt{\epsilon_2}}{\sqrt{\epsilon_1} + \sqrt{\epsilon_2}} \right)^2 \quad (2.4)$$

where, (ϵ_1) and (ϵ_2) are the relative permittivity at a certain frequency at the boundary of reflection.

Radar Cross Section (RCS):

The Radar Cross Section (RCS), denoted by σ , is a measure of how detectable an object is by radar. It quantifies the amount of electromagnetic energy that a target reflects back to the radar relative to an idealized target. RCS is critical in radar systems as it determines the strength of the radar return signal from a given target. The RCS of a target is defined as:

$$\sigma = \frac{\text{Power reflected back to radar per unit solid angle}}{\text{Power density of the incident wave at the target}} \quad (2.5)$$

The unit of RCS is square meters (m^2), representing an equivalent area that reflects energy back toward the radar.

Factors affecting RCS:

- **Size:** Larger objects generally have a higher RCS since they can reflect more electromagnetic energy.
- **Shape:** The geometry of the target significantly impacts how it scatters radar waves. Flat surfaces oriented toward the radar reflect more energy (e.g., a metallic plate). Curved or irregular surfaces scatter energy in multiple directions, reducing RCS.
- **Material Properties:** Conductive materials (e.g., metal) have a higher RCS because they reflect more electromagnetic energy while Non-conductive or absorbing materials (e.g., composites or stealth coatings) reduce RCS.
- **Orientation:** The angle between the radar beam and the target affects RCS. At certain angles, the RCS may peak (specular reflection), while at others, it may drop significantly.
- **Frequency/Wavelength:** The RCS varies with radar frequency or wavelength. Smaller wavelengths can resolve finer features on a target, potentially increasing RCS if these features reflect energy well.

In the radar range equation, σ represents how effectively the target reflects radar energy back toward the source. For a given transmitted power and range, the strength of the received signal depends directly on σ ,

$$P_r \propto \sigma \tag{2.6}$$

where P_r is the received power, and σ is the radar cross section.

A larger RCS means more energy is reflected back to the radar, resulting in a stronger return signal.

Doppler Effect:

The echo's frequency shifts when the target moves. The Doppler shift is this frequency shift. If the Target is moving towards the Radar there will be an increase in the frequency of the received signal while if the target is moving away from the radar, there will be a decrease in the frequency of the received signal.

- **Doppler Effect in continuous wave radar:**
In continuous-wave radar, the Doppler effect is observed directly as a frequency

shift. Let the transmitted continuous wave be represented as:

$$s_t(t) = A \cos(2\pi ft) \quad (2.7)$$

where:

- A is the amplitude,
- f is the frequency of the transmitted signal,
- t is time.

If the target is moving with a radial velocity v , the frequency of the received signal will shift by f_d , where:

$$f_d = \frac{2vf}{c} \quad (2.8)$$

where c is the speed of light.

The received signal $s_r(t)$ can be represented as:

$$s_r(t) = A \cos(2\pi(f + f_d)t) = A \cos(2\pi ft + 2\pi f_d t) \quad (2.9)$$

The radial velocity v of the target can be derived from the Doppler shift f_d as follows:

$$v = \frac{f_d c}{2f} \quad (2.10)$$

- **Doppler effect in Pulse Radar:** In pulsed radar, the Doppler effect is observed as a phase shift between successive pulses.

The transmitted signal can be represented as a pulse train:

$$s_t(t) = A \sum_{n=0}^{N-1} p(t - nT_p) \cos(2\pi ft) \quad (2.11)$$

where:

- $p(t)$ represents the pulse envelope (shape),
- T_p is the pulse repetition interval (PRI),
- N is the number of pulses in the pulse train.

For a target moving with velocity v , each pulse n experiences a round-trip distance change of $2vT_p$, causing a phase shift $\Delta\phi$ between successive pulses given by:

$$\Delta\phi = \frac{4\pi v f T_p}{c} \quad (2.12)$$

The phase shift $\Delta\phi$ between successive pulses results in an effective Doppler frequency f_d , which can be expressed as:

$$f_d = \frac{\Delta\phi}{2\pi T_p} = \frac{2vf}{c} \quad (2.13)$$

Each returned pulse experiences an accumulated phase shift, resulting in the received pulse sequence:

$$s_r(nT_p) = A \cos(2\pi f n T_p + n \Delta \phi) \quad (2.14)$$

The target's radial velocity v can be calculated as:

$$v = \frac{f_d c}{2f} \quad (2.15)$$

Therefore, while the Doppler effect in continuous-wave radar is observed directly as a frequency shift, in pulsed radar it is observed as a phase shift between pulses, which corresponds to the same Doppler frequency.

Range Resolution:

This is the radar's ability to distinguish between two closely spaced objects along the same line of sight. High range resolution is desirable to ensure a powerful Radar system. Range Resolution can be calculated as follows,

- **Based on Pulse Width (τ):** Pulse width is the duration of the radar pulse, typically measured in microseconds. Since the pulse travels at the speed of light c , the shorter the pulse, the smaller the distance between two objects that the radar can distinguish.

$$\text{Range Resolution} = \frac{c \cdot \tau}{2} \quad (2.16)$$

- **Based on Radar Bandwidth (B):** In modern radar systems, particularly those using pulse compression or frequency modulated continuous wave (FMCW) techniques, the bandwidth of the radar signal is used to determine range resolution.

A larger bandwidth provides better range resolution, as the system can differentiate between returns from objects at different distances more precisely.

$$\text{Range Resolution} = \frac{c}{2B} \quad (2.17)$$

Pulse Repetition Frequency (PRF): Pulse Repetition Frequency (PRF) refers to the number of pulses that a radar system transmits per second. It is a fundamental parameter in pulsed radar systems, determining how often the radar sends out a pulse to detect targets. PRF is typically measured in Hertz (Hz), where 1 Hz equals one pulse per second. The following are some key concepts related to PRF.

- **Pulse Repetition Interval (PRI):** The Pulse Repetition Interval (PRI) is the time between the start of one pulse and the start of the next. It is the

inverse of PRF.

$$\text{PRI} = \frac{1}{\text{PRF}} \quad (2.18)$$

- **Range Ambiguity:** The PRF affects the radar's **maximum unambiguous range**, which is the farthest distance a target can be accurately detected without ambiguity. If the next pulse is sent out before the previous pulse's echo returns, the radar might mistake the returning echo for one coming from a closer target.

Maximum unambiguous range (R_{\max}) is calculated as follows,

$$R_{\max} = \frac{c}{2 \cdot \text{PRF}} \quad (2.19)$$

High PRF results in shorter unambiguous range but better Doppler resolution, while low PRF allows for detection of targets at longer ranges but limits velocity measurement accuracy.

- **Trade-off Between Range and Velocity Measurement:** PRF also influences the radar's ability to measure the velocity of moving targets using the Doppler effect. Higher PRF improves the radar's ability to measure target velocity accurately, especially for fast-moving objects. However, it introduces range ambiguity, meaning targets at different ranges might appear indistinguishable.

This means that radar systems often have to balance between range and velocity measurement by choosing an appropriate PRF:

- **Low PRF:** Better for long-range detection, but not as good for velocity measurement due to Doppler ambiguity.
- **High PRF:** Better for measuring velocity and tracking fast-moving targets, but introduces range ambiguity.
- **Staggered PRF:** To overcome the limitations of both low and high PRFs, radars often use staggered PRF, where the PRF is varied from pulse to pulse. This helps to reduce both range and velocity ambiguities, improving overall performance.

Antenna Gain: Antenna Gain is a measure of how well an antenna converts input power into radio waves in a specified direction (for transmitting) or how effectively it receives radio waves from a specific direction (for receiving). It is expressed as the ratio of the power radiated by an antenna in a given direction to the power radiated by an ideal isotropic antenna (a theoretical antenna that radiates power uniformly

in all directions) at the same distance.

Antenna gain G is generally expressed in decibels (dB) and can be calculated as:

$$G = 10 \log_{10} \left(\frac{P_r}{P_i} \right) \quad (2.20)$$

Where:

- P_r is the power radiated in the direction of interest.
- P_i is the power radiated by an isotropic antenna.

Alternatively, gain can also be expressed in terms of directivity D and efficiency η as:

$$G = D \cdot \eta \quad (2.21)$$

- **Directivity:** Directivity is a measure of how focused the radiation pattern of an antenna is in a particular direction compared to an isotropic antenna (which radiates equally in all directions). Directivity describes the ability of an antenna to direct the power it radiates in a particular direction. It is dimensionless and typically greater than 1 for directional antennas. Directivity reflects how well the antenna focuses energy in the desired direction but does not account for energy losses.
- **Efficiency:** Efficiency is the ratio of the power radiated by the antenna to the total input power supplied to the antenna. It accounts for power losses due to factors like resistive heating, material imperfections, and impedance mismatches in the antenna structure.

Mathematically, efficiency is given by:

$$\eta = \frac{P_{\text{radiated}}}{P_{\text{input}}} \quad (2.22)$$

Where:

- P_{radiated} is the power radiated by the antenna.
- P_{input} is the total input power supplied to the antenna.
- Efficiency ranges from 0 to 1 (or from 0% to 100%)
 - * $\eta = 1$ (or 100%) means the antenna is perfectly efficient and radiates all of the power it receives.
 - * $\eta < 1$ indicates that some power is lost (e.g., as heat).

Radar Range Equation: The radar equation is a fundamental formula that expresses the relationship between the power transmitted by a radar system, the distance to a target, the characteristics of the target, and the power received by the

radar. It helps in understanding how different factors influence the performance of radar in detecting objects.

The gain of an antenna G is a measure of how effectively it directs radio frequency energy compared to an isotropic radiator. It is defined as the ratio of the power density S_d radiated in the maximum direction to the power density S_{iso} radiated by an isotropic antenna, for the same input power:

$$G = \frac{S_d}{S_{iso}} \quad (2.23)$$

For an isotropic antenna, the power density at a distance R is:

$$S_{iso} = \frac{P_t}{4\pi R^2} \quad (2.24)$$

where, P_t is the transmitted power. This comes from the fact that the power spreads spherically, and the surface area of a sphere is $4\pi r^2$. For a directional antenna with gain G , the power density S_d in the direction of maximum radiation becomes:

$$S_d = \frac{GP_t}{4\pi R^2}. \quad (2.25)$$

The effective aperture A_e of an antenna relates to the ability of the antenna to capture power from an incoming electromagnetic wave. If the wave has power density S , the total power P_r captured by the antenna is:

$$P_r = S \cdot A_e \quad (2.26)$$

Substituting S_d into the equation, the power received by the antenna is:

$$P_r = \frac{GP_t}{4\pi R^2} \cdot A_e \quad (2.27)$$

Now consider a radar system. The target reflects some portion of the incident power. This reflected power depends on the Radar Cross Section (RCS) σ , which is a measure of how much power the target scatters back toward the radar. The power density at the target is:

$$S_{\text{return}} = \frac{\left(\frac{P_t G_t}{4\pi R^2}\right) \sigma}{4\pi R^2}. \quad (2.28)$$

The radar antenna collects the reflected power based on its **effective aperture**, A_e . The received power P_r is:

$$P_r = S_{\text{return}} \cdot A_e \quad (2.29)$$

Substituting S_{return} into this equation:

$$P_r = \left(\frac{P_t G_t \sigma}{4\pi R^2} \right) A_e \quad (2.30)$$

Simplifying:

$$P_r = \frac{P_t G_t \sigma A_e}{(4\pi)^2 R^4} \quad (2.31)$$

Signal to Noise Ratio (SNR): Signal-to-Noise Ratio (SNR) is a key parameter in radar systems that quantifies the level of the desired signal relative to the background noise. It is crucial for assessing the performance of radar systems, as it directly affects the ability to detect, identify, and track targets. A higher SNR indicates a clearer and more discernible signal, leading to better target detection and classification.

SNR is defined as the ratio of the power of the desired received signal to the power of the background noise. Mathematically, it is expressed as:

$$\text{SNR} = \frac{P_r}{N} \quad (2.32)$$

Where:

- P_r : Power of the Received signal.
- N : Noise Power.

The noise power at the radar receiver is given by:

$$N = kT_s B_n \quad (2.33)$$

where,

- k : Boltzmann's constant (1.38×10^{-23} J/K),
- T_s : System noise temperature (in Kelvin),
- B_n : Receiver noise bandwidth (in Hz).

Using the radar range equation for P_r and noise power $N = kT_s B_n$:

$$\text{SNR} = \frac{\frac{P_t G_t G_r \sigma \lambda^2}{(4\pi)^3 R^4}}{kT_s B_n} \quad (2.34)$$

Simplifying:

$$\text{SNR} = \frac{P_t G_t G_r \sigma \lambda^2}{(4\pi)^3 R^4 kT_s B_n} \quad (2.35)$$

Factors affecting SNR:

- **Transmitted Power (P_t):** Increasing the power of the transmitted signal directly increases P_r thus improving the SNR.
- **Antenna Gain:** The gain of the transmitting and receiving antennas affects the effective power received from the target, impacting the SNR.
- **Range to the Target (R):** As the range increases, the power of the signal decreases due to the inverse square law, which reduces the SNR.
- **Radar Cross Section (σ):** The RCS of the target affects how much signal is reflected back to the radar. Larger RCS values lead to stronger return signals, enhancing SNR.
- **Noise Figure:** The inherent noise of the radar system and its components (like amplifiers) adds to the background noise level affecting SNR.
- **Environmental Conditions:** Atmospheric conditions, such as precipitation, can introduce additional noise and attenuation, impacting SNR.

Now that the key concepts of radar sensors have been covered, the focus shifts to the radar sensor used in this thesis.

2.5 Acconeer A111 Radar Sensor

It is a Pulsed Coherent Radar by Acconeer that operates at 60GHz that has been optimized for High precision and Ultra Low power with Antenna in Package(AiP) and Integrated Baseband. The following are some of the specifications available in Acconeer A111 datasheet that are relevant to this thesis.

Table 2.1 provides system parameters sourced from [4].

Table 2.1: System parameters of the radar sensor [4].

Parameter	Value
Minimum Range	2m with absolute accuracy in mm, relative accuracy in μm
Antenna Type	Folded Dipole
Operating Frequency Range	57GHz - 64GHz
TX HPBW- E plane	Typical value= 53°
TX HPBW- H plane	Typical value= 65°
EIRP	10dB
Bitrate of the system	16bps

2. Theory

Figure 2.3 and Figure 2.4 show the Radar chipset and its functional block diagram respectively.

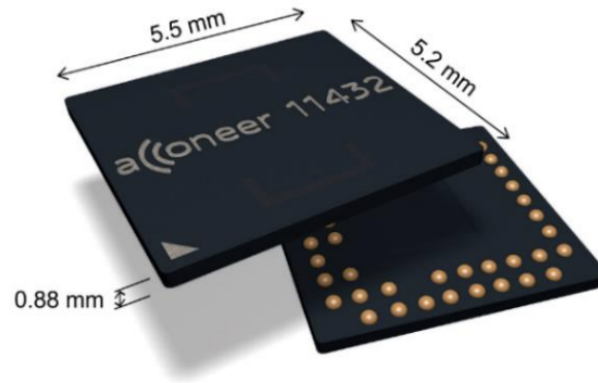


Figure 2.3: A111 Radar chipset [4]

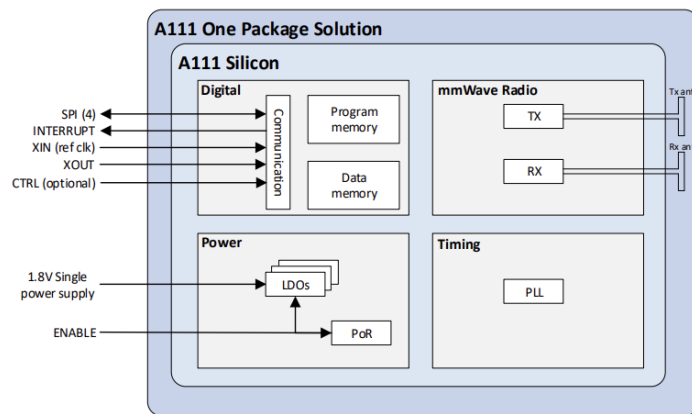


Figure 2.4: A111 functional block diagram [3]

The discussion now shifts to the theory behind the next essential element of this thesis: the robot and ABB RobotStudio, the interface used to program it.

2.6 ABB CRB 15000

The CRB 15000 robot also known as the ABB GoFa, is a collaborative robot (cobot) designed by ABB.

2.6.1 Key Features

Collaborative Robot:

The CRB 15000 is designed to work safely alongside human workers without needing physical barriers. It is equipped with sensors and safety features to detect humans and avoid collisions.

Payload and Reach:

It is a flexible, agile 6-axis articulated robot that can carry payloads upto 15 kilograms. The robot arm is designed to have a reach ranging from 0.95 metres to 1.51 metres making it quite flexible for use in the manufacturing industry.

Safety Features:

It has built-in safety-certified technology, allowing it to stop motion when it detects a collision or other hazardous situations. Also, its Power and force limiting (PFL) functions ensure the robot operates safely around humans, with low risk of injury during unintended contact.

Ease of use:

Designed for ease of programming and use, the GoFa can be programmed through ABB's Wizard Easy Programming. This drag-and-drop interface simplifies automation for users with little or no coding experience. It also supports ABB RobotStudio, an advanced software suite that allows for simulation and virtual commissioning. The Cobot also facilitates 'Lead Through programming' where users can move the robot physically to teach it positions and movements.

It is a highly versatile robot which is very easy to use with various applications like automated material handling, Intricate assemble tasks, machine tending for loading and unloading machines and packaging. This Cobot is also known to deliver high levels of precision and repeatability for tasks that demand consistency. Another good feature of this Cobot is that it is easily compatible with a number of end effectors making it very versatile.

Figure 2.5 shows the Cobot used.



Figure 2.5: ABB CRB 15000 Cobot [8]

2.7 Wizard Easy Programming

ABB's Wizard Easy Programming is a user-friendly, graphical programming interface that makes programming robots easier, especially for people without extensive programming knowledge or robotics expertise. It is often used with ABB's collaborative robots, such the CRB 1100 (YuMi) and the CRB 15000 GoFa. This was used to program the robot in this thesis.

The following are some of Wizard Easy Programming's key features.

Drag-and-Drop Interface:

- **No coding required:** Wizard Easy Programming uses a graphical, block-based system where users can create robot programs by simply dragging and dropping predefined function blocks. These blocks represent standard robot actions such as move commands, grasping objects, and conditional logic (e.g., "if-else" statements).
- **Predefined Function Blocks:** The interface comes with a set of ready-made blocks that allow users to program the robot's actions quickly. These include:

- **Move:** Commands for moving the robot to specific positions.
- **Wait:** For activating end-of-arm tools like grippers.
- **Grasp:** To introduce time delays.
- **Logic:** Such as loops, conditions, and wait-for-signal commands.

Intuitive User Experience:

- **Ease of use:** The interface is designed to be easy for operators with little to no prior programming knowledge. Users can see how their robot's actions will flow without needing to understand complex coding syntax.
- **Real-time feedback:** The user interface provides real-time feedback, showing where the robot is in its programmed sequence and allowing for adjustments spontaneously.

Built-In Safety: The Wizard system integrates safety features, which allow operators to ensure that the robot works safely around humans without complicated setup procedures. These safety functions can be automatically triggered by the program depending on the robot's movements and tasks.

Lead-Through Programming Support: Wizard Easy Programming often works alongside lead-through programming (also called hand-guiding), where users can manually guide the robot arm through the desired motion, and the robot will record these movements. This can make creating more complex motions straightforward for those without robotics expertise.

Simulation and Testing: Programs created with Wizard Easy Programming can be simulated in ABB's RobotStudio software before they are executed in the physical environment. This helps users test and verify the robot's movements, preventing mistakes or inefficiencies in the workflow.

Expandable and Customizable: Although Wizard Easy Programming is designed to be simple, it can also be expanded to accommodate more advanced programming needs. If users require more complex control, they can switch to RAPID, ABB's more advanced robot programming language, for fine-tuning.

Multilingual Support: Wizard Easy Programming is available in multiple languages, further supporting ease of use for operators worldwide.



Figure 2.6: Wizard Easy Programming on a controller

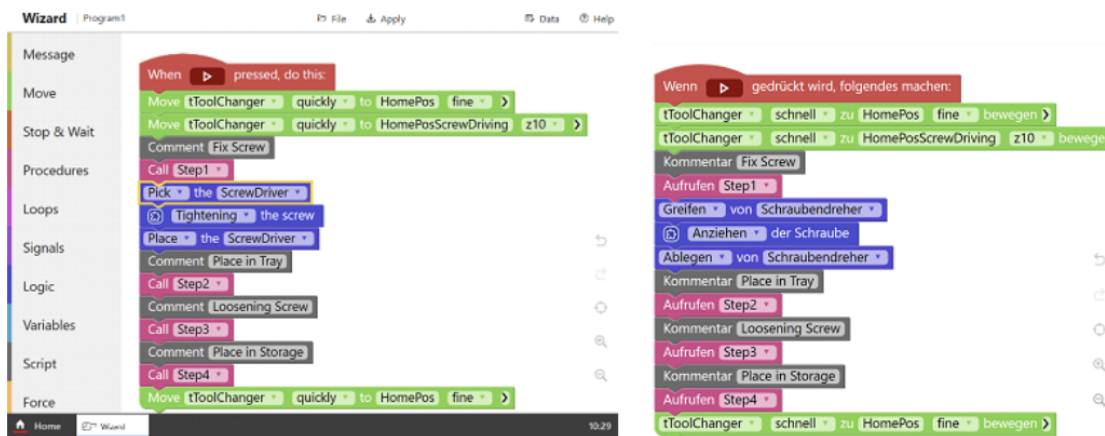


Figure 2.7: Wizard Easy Program Snippet- Multilingual

2.8 ABB Robotstudio

ABB RobotStudio is ABB's advanced software platform designed for robot simulation, programming, and virtual commissioning. The following are some of its key features.

Offline Programming:

- RobotStudio allows users to program ABB robots without disrupting production. Offline programming means that the robot does not need to be taken out of its production environment to write, test, or modify its programs.
- It uses ABB's RAPID programming language, the standard programming language for ABB robots.

Virtual Commissioning: RobotStudio enables users to virtually commission robotic cells and systems. This means you can fully simulate and test the behavior of the robot, its interactions with other machines, sensors, and the environment, ensuring

that everything works as expected before the physical robot is even installed. This reduces downtime during installation and helps avoid costly mistakes, as most issues can be identified and resolved virtually.

Motion Simulation:

- RobotStudio offers tools for simulating the motion of robots, including precise kinematics and dynamics calculations, to ensure robots will operate safely and efficiently in a production line.
- The software supports simulation of robot arms, grippers, tools, and conveyor belts, allowing users to fine-tune path planning and motion execution.

Real-Time Connection to Physical Robots: RobotStudio can connect in real-time to physical ABB robots via an Ethernet link, allowing users to control and monitor the actual robot's movements in the same virtual environment. This means that real-world robot performance can be tested and debugged remotely. This capability is useful for making small adjustments or troubleshooting without being physically present on-site.

Wizard Easy Programming Integration: For simpler applications, ABB's Wizard Easy Programming can be integrated into RobotStudio, providing an intuitive drag-and-drop interface for programming tasks. This makes it easier for non-experts to set up basic robotic actions within the platform.

3

Method

The whole project was carried out in two parts. Since the aim of this thesis is to validate the results from AVxcelerate (AVx) as a radar simulation tool, real radar measurements were recorded as a benchmark. This chapter describes the measurement scenarios and methods that are identical to both the simulation and real radar measurements.

3.1 Real Radar Data

3.1.1 Case 1- Stationary Target

The first set of range measurements were taken with a static corner reflector of dimensions (10 cm \times 10 cm \times 10 cm) as target. The idea behind this simple case was to ensure that the entire setup was error-free and had no calibration errors with respect to the range of the target.

Description of the Setup:

Figures 3.1, 3.3 and 3.4 depict the setup in which the measurements were taken. A makeshift 'Anechoic Chamber' was built using High frequency Absorbers to avoid multipath reflections. The corner reflector was placed such that its centre is at a distance of approximately 23.5 centimetres from the Radar sensor as shown in the images below. The A111 Radar sensor was powered up using a Raspberry Pi Module. The measurements were taken in two different formats of the A111 Exploration tool which have been explained in detail in Section 3.1.2.

Tetrahedral Corner Reflector as Target: A Tetrahedral corner reflector of dimensions (10 cm \times 10 cm \times 10 cm) was used as the sole target in this thesis.

A tetrahedral corner reflector is a type of passive radar reflector composed of three mutually perpendicular flat surfaces that form a shape resembling one corner of a cube, also called a trihedral corner reflector. When radar waves strike the reflector, the three flat surfaces cause the waves to bounce multiple times and return directly back toward the radar source, even if the waves hit the reflector at different angles.

- **Structure and Design:**

- **Three Perpendicular Surfaces:** The tetrahedral (or trihedral) corner reflector consists of three orthogonal surfaces that meet at a right angle, forming a corner.
- **Geometrical Symmetry:** The reflector's geometric configuration ensures that any radar signal incident on it is reflected multiple times between the three surfaces, and these reflections converge back in the direction of the incoming signal.

- **Reason for use as target:**

- **High Radar Cross Section:** A corner reflector, reflects incoming radar signals directly back toward the radar source. This is because each flat surface redirects the signal along the path it came from. The result is a much larger radar cross section (RCS) compared to other objects of similar physical size.
- **Consistent and Predictable Reflection:** A corner reflector provides a stable, consistent reflection regardless of the angle of incidence of the radar waves. When radar signals hit the surfaces of the corner reflector, they are reflected multiple times before returning to the source. This predictable reflection pattern allows radar systems to be accurately calibrated or evaluated, ensuring repeatable results.
- **Ideal for Radar Calibration:** Corner reflectors have a known radar cross section (RCS) that is easy to calculate based on their geometry. This makes them extremely useful for radar calibration purposes. Once the radar is calibrated with the corner reflector, measurements of other objects can be compared against the known reflection from the corner reflector.

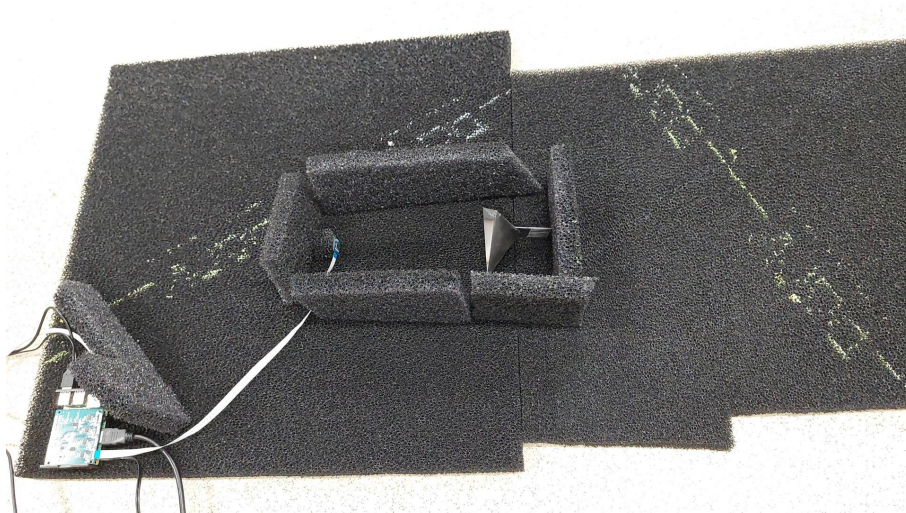


Figure 3.1: Static Corner Reflector in a Make-shift Anechoic Chamber

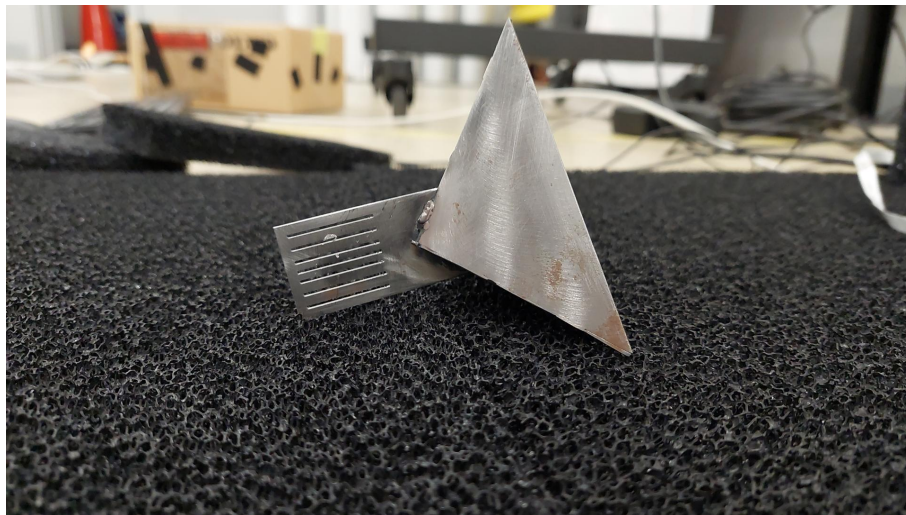


Figure 3.2: Corner Reflector of dimensions 10 cm \times 10 cm \times 10 cm



Figure 3.3: Corner Reflector placed at approximately 23.5 cms from the Radar

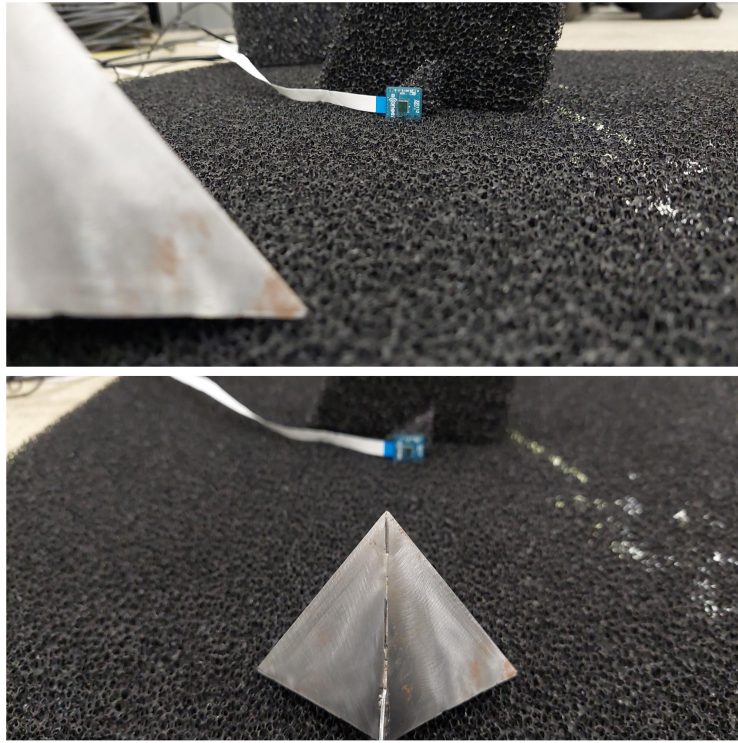


Figure 3.4: The Corner Reflector and the A111 Radar Sensor

3.1.2 Case 2- Target in Motion

With the radar free from calibration errors, the next step involved recording range measurements for a moving target.

Description of the setup: Figure 3.5 illustrates the experimental setup. To enable precise movement of the target relative to the stationary radar, a linear actuator was utilized. The corner reflector was mounted on the end effector of an ABB CRB 15000 robot, allowing for controlled positioning. Additionally, the A111 radar sensor, as depicted in the figures, was securely mounted on a custom wooden rig.

Motion of the Corner Reflector: A simple 3D motion was implemented in which the corner reflector traced the perimeter of a cuboid. This motion generated significant variations in the range measurements of the target, as it involved changes in coordinates along all three axes (x , y , and z), thereby adding value to the purpose of the experiment.

To achieve this motion, the ABB CRB 15000 robot arm was programmed. A code was developed using ABB's Wizard Easy Programming logic to specify the coordinates of the end effector, which holds the corner reflector, as well as the joint states of the six joints that define the robot's six degrees of freedom. The following is the code used.

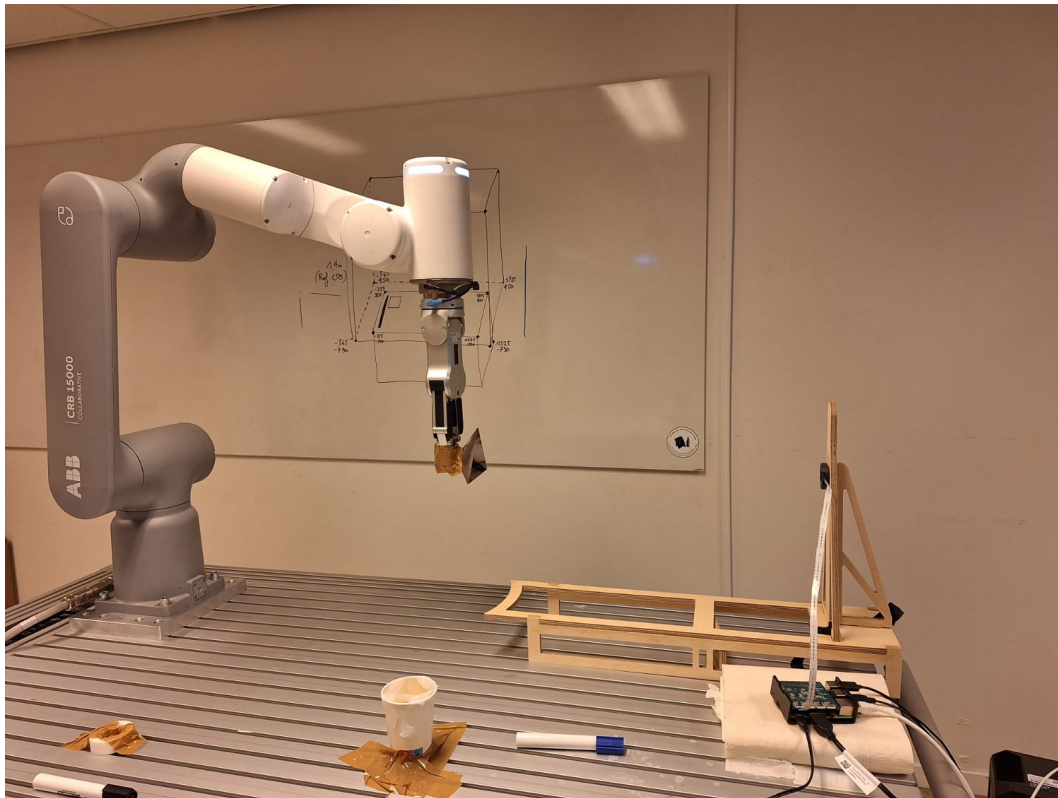


Figure 3.5: Setup for physical measurements for target in motion

Listing 3.1: ABB Wizard Easy Programming Code

```

1 MODULE Wizard
2
3 PERS robtarget Location1:=[[617.941, -170.798, 480.778],
   [0.00671051, -0.996749, 0.0802682, 0.00160228], [-1, 0, -3, 0],
   [9000000000, 9000000000, 9000000000, 9000000000, 9000000000,
   9000000000]];
4
5 PERS robtarget Location2:=[[637.665, 66.3496, 480.745],
   [0.00628522, -0.994329, -0.10612, 0.00285076], [0, 0, -3, 0],
   [9000000000, 9000000000, 9000000000, 9000000000, 9000000000,
   9000000000]];
6
7 PERS robtarget Location3:=[[628.165, 167.308, 480.703], [5.15033e
   -8, 0.994355, 0.106105, -3.318e-7], [0, 0, -2, 0], [9000000000,
   9000000000, 9000000000, 9000000000, 9000000000]];
8
9 PERS robtarget Location5:=[[602.851, 159.841, 561.111],
   [0.00000325706, -1, 0.000857853, -0.0000609413], [0, -1, -2, 0],
   [9000000000, 9000000000, 9000000000, 9000000000, 9000000000,
   9000000000]];
10
11 PERS robtarget Location6:=[[602.826, -160.077, 561.13], [6.69989e
   -7, 1, -0.000778003, 0.000085936], [-1, 0, -3, 0], [9000000000,
   9000000000, 9000000000, 9000000000, 9000000000]];
12
13 PERS robtarget Location7:=[[602.854, 159.808, 720.791],

```

3. Method

```

    [0.00000557019, 1, -0.000877797, 0.0000303285], [0, -1, -2, 0],
    [9000000000, 9000000000, 9000000000, 9000000000, 9000000000,
    9000000000]];
14
15 PERS robtarget Location8:=[[602.804, -160.036, 720.99],
    [0.00000470055, 1, -0.000754484, 0.000109071], [-1, 0, -3, 0],
    [9000000000, 9000000000, 9000000000, 9000000000, 9000000000,
    9000000000]];
16
17 PERS robtarget Location9:=[[442.385, 159.856, 561.122],
    [0.00000784572, -1, 0.000809498, -0.0000719538], [0, -1, -2, 0],
    [9000000000, 9000000000, 9000000000, 9000000000, 9000000000,
    9000000000]];
18
19 PERS robtarget Location11:=[[442.35, 159.84, 722.221],
    [0.0000116611, -1, 0.000817439, -0.0000805542], [0, -1, -2, 0],
    [9000000000, 9000000000, 9000000000, 9000000000, 9000000000,
    9000000000]];
20
21 PERS robtarget Location12:=[[442.177, -160.096, 561.139], [8.39672e
    -8, -1, 0.000822284, -0.0000863503], [-1, 0, -3, 0],
    [9000000000, 9000000000, 9000000000, 9000000000, 9000000000,
    9000000000]];
22
23 PERS robtarget Location14:=[[442.179, -160.101, 721.012],
    [0.00000812866, -1, 0.000820569, -0.0000674387], [-1, 0, -3, 0],
    [9000000000, 9000000000, 9000000000, 9000000000, 9000000000,
    9000000000]];
24
25 PERS robtarget Location15:=[[762.696802616119, 0.015378134776256,
    561.082601547241], [9.01022815469332e-7, -0.999999701976776,
    0.000778551737312227, -0.0000320407598337624], [0, 0, -2, 0],
    [9000000000, 9000000000, 9000000000, 9000000000, 9000000000,
    9000000000]];
26
27 PERS robtarget Location50:=[[602.829, 0.0541112, 561.093],
    [0.00000117862, -1, 0.000751413, -0.0000502279], [0, 0, -2, 0],
    [9000000000, 9000000000, 9000000000, 9000000000, 9000000000,
    9000000000]];
28
29 PERS robtarget Location100:=[[549.734, -0.0327211, 718.431],
    [1.43636e-10, 1, -0.000796161, -1.94989e-7], [-1, 0, -2, 0],
    [9000000000, 9000000000, 9000000000, 9000000000, 9000000000,
    9000000000]];
30
31 PROC main()
32
33 MoveL Location50, v50, fine, tool0\WObj:=wobj0;
34
35 MoveL Location5, v50, fine, tool0\WObj:=wobj0;
36
37 MoveL Location7, v50, fine, tool0\WObj:=wobj0;
38
39 MoveL Location8, v50, fine, tool0\WObj:=wobj0;
40
41 MoveL Location6, v50, fine, tool0\WObj:=wobj0;
```

```

42
43 MoveL Location50, v50, fine, tool0\WObj:=wobj0;
44
45 MoveL Location9, v50, fine, tool0\WObj:=wobj0;
46
47 MoveL Location11, v50, fine, tool0\WObj:=wobj0;
48
49 MoveL Location12, v50, fine, tool0\WObj:=wobj0;
50
51 MoveL Location14, v50, fine, tool0\WObj:=wobj0;
52
53 MoveL Location50, v50, fine, tool0\WObj:=wobj0;
54
55 MoveL Location15, v50, fine, tool0\WObj:=wobj0;
56
57 MoveL Location50, v50, fine, tool0\WObj:=wobj0;
58
59 Stop;
60
61 ENDPROC
62
63 ENDMODULE

```

In this code, all the positions of the robot are defined in the 'Module Wizard' block. The position definition includes the X,Y,Z coordinates of the tool center point (TCP) of the robot in millimeters, followed by the quaternion representation of the TCP, the configuration of the robot's arm to avoid elbow flips and wrist singularities and finally external axes configurations. The robot uses external axes such as positioners or tracks to enable more complex movements. High values for these axes indicate that they are not required for this simple motion. The 'MoveL' commands in the main function are executed sequentially. Therefore the order must be perfect to produce the desired motion.

Joint States of the Robot: To replicate the motion in the simulation, the simplest method was to record the robot's joint states at regular intervals throughout the entire movement. This was achieved using the 'Signal Analyzer' feature in ABB Robotstudio, which allows for virtual operation of the robot and enables the motion to be simulated. The Signal Analyzer tool provides real-time signal monitoring during both simulations and actual robot movements. Since the six joints of the robot are part of the 'Custom Signal Selection' in Signal Analyzer, they were specifically selected to capture the joint positions (angles) at intervals of 0.024 seconds during the simulation. Figure 3.6 shows the Signal Analyzer window. J1 through J6 indicate each of the 6 joints in the CRB 15000.

Acconeer Exploration Tool and Radar measurements With the motion of the corner reflector sorted, the next important step was setting up the radar for measurements. A111 radar sensor has Acconeer Exploration Tool which is the interface for sensor configuration, real time data visualization and measurement recording. Figure 3.7, shows the interface. The A111 sensor's IP address is used in the interface to link it with the physical sensor. The Range interval for measurements is defined

3. Method

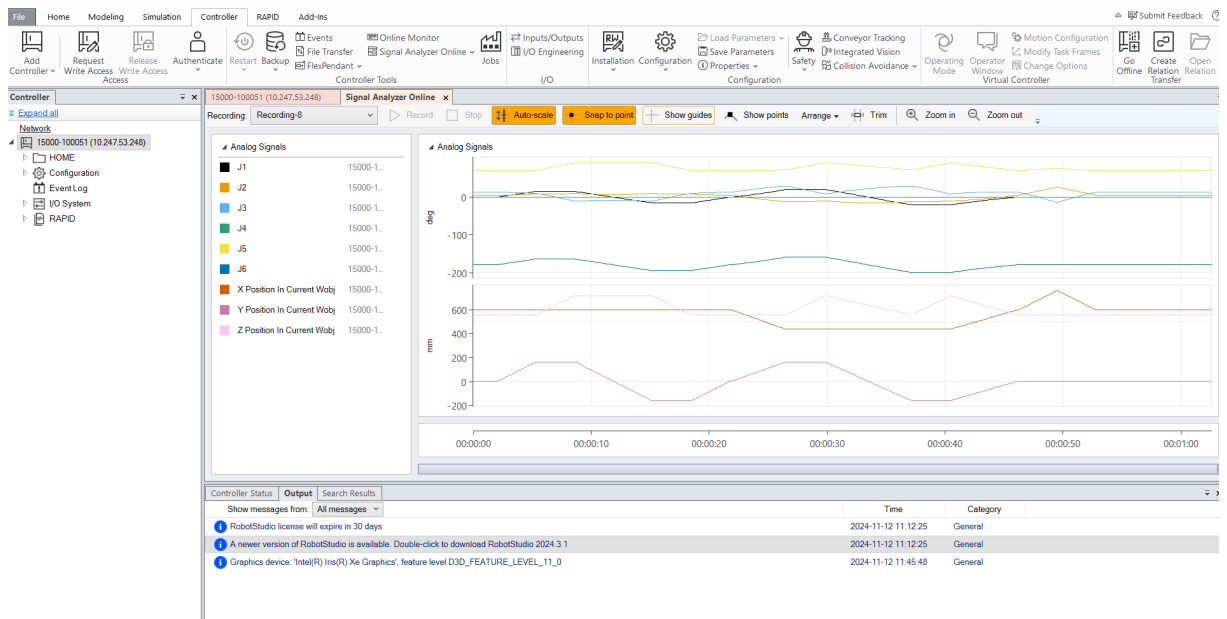


Figure 3.6: Signal Analyzer window from ABB Robotstudio

in the 'Range interval [m]' option. There are several other settings like 'Profile' which defines the length of the radar pulse. In this case, since a shorter pulse was required, the 'Profile' parameter was set accordingly. There is a 'Gain' parameter as well which defines the receiver gain. This exploration tool offers multiple formats to record measurements. Two such formats used are:

- **Envelope:** The Envelope Service provides a filtered amplitude representation of the radar reflections. This is useful for detecting objects, measuring distances, and monitoring changes in the environment. The service strips away the complex details of the raw radar data, focusing on the amplitude of the signal that corresponds to the reflection of objects in the sensor's field of view. The sensor output is therefore amplitude values at different distances. In Figure 3.7, the Amplitude versus depth plot shows the signal for 1 frame of the corner reflector's motion.

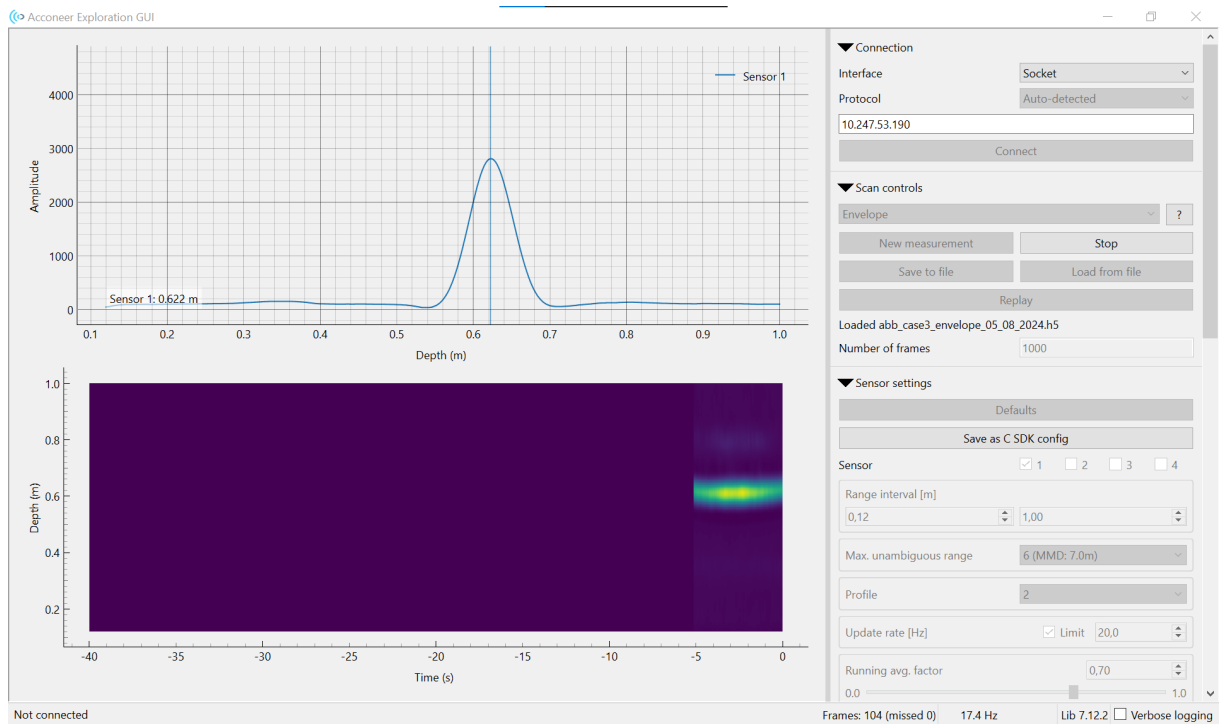


Figure 3.7: Corner reflector's motion in Envelope format

- **Sparse:** Sparse Service works best for low power systems. The signals are sampled in 6cm intervals and is therefore only used in cases where the target is moving. The Sparse Service is optimized to respond to changes in amplitude or distance that indicate movement. This means it is designed to trigger detection events based on the temporal changes in the radar reflections rather than relying on static signals that provide more stable (but potentially insufficient) information.

The output is a number of sweeps N_s per data frame sampled one after the other. Each sweep has data points sampled at every 6cm interval with each sweep interval T_s as shown in figure 3.8. Figure 3.9 shows the measurements of motion collected in the sparse format.

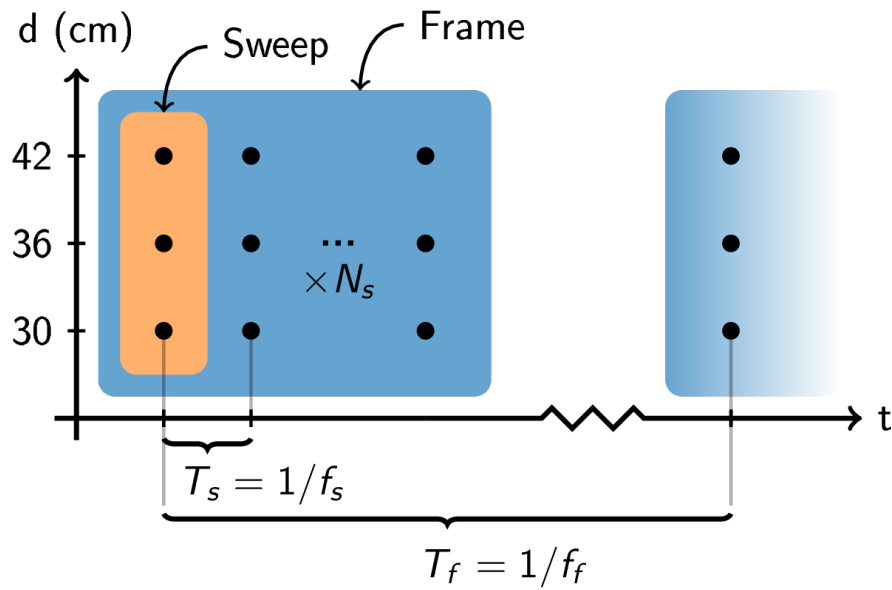


Figure 3.8: Sparse service format [1]

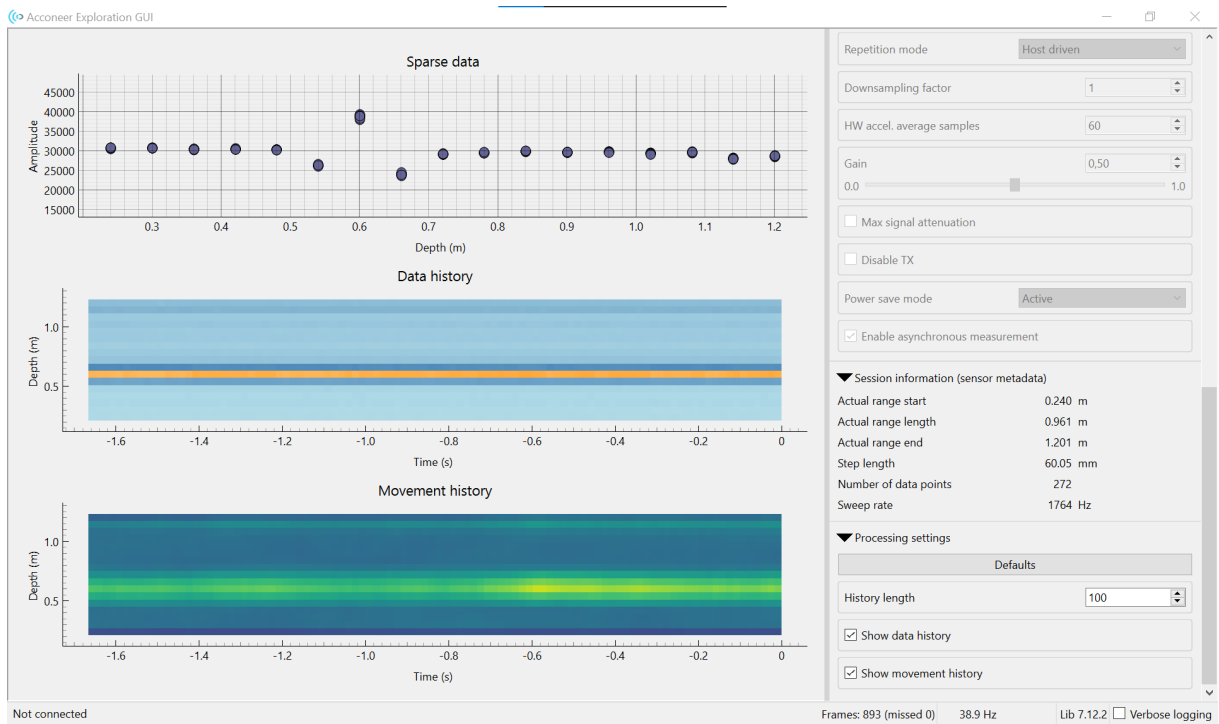


Figure 3.9: Corner's reflector's motion in Sparse format

3.2 Simulated Radar Data

3.2.1 Case 1- Stationary Target

A simulation was conducted in Ansys AVx, replicating conditions similar to those observed in real radar measurements. In order to to successfully carry out the sim-

ulation, a number of steps had to be carried out.

A pre-existing 3D model of the physical corner reflector, used for real-world radar measurements, was imported into AVx. To ensure precise spatial alignment, the model's coordinates were first adjusted using Blender. Once configured, the model was successfully imported from Blender into the AVx virtual environment.

The next key component integrated into the simulation was the A111 radar sensor, previously configured in Sensor Modeler by AVx Sensor Labs. Figures 3.10 and 3.11 display the sensor parameters within the AVx Sensor Labs interface. As shown, the center frequency is set to 60.5 GHz, aligning with the transmission frequency of the real radar system. During each coherent processing interval (CPI), the radar transmits 64 pulses, with each pulse containing 256 samples. The transmission power is specified at 1 W and the pulse interval is set to 0.0769 μ s.

- **Coherent Processing Interval:** Refers to the duration over which the radar processes a series of consecutive pulses to extract target information while maintaining phase coherence. During this interval, the radar system assumes that the target's motion (e.g., velocity) and the environment remain relatively constant.
 - During the CPI, the radar transmits a sequence of short pulses. Each pulse is emitted at a specific interval known as the Pulse Repetition Interval (PRI).
 - These pulses are coherent, meaning they maintain a fixed phase relationship with one another, which is essential for Doppler processing.
 - After transmitting each pulse, the radar switches to the receiving mode to collect echoes from targets.
 - The time between transmitting a pulse and receiving its echo corresponds to the distance of the target (range). This is based on the time-of-flight of the radar signal.
 - The radar preserves phase information for the echoes received during the CPI. This allows the system to measure the Doppler shift, which provides information about the radial velocity of the target.
 - Since all pulses within a CPI are processed together, the radar assumes that the environment and target motion remain constant during this interval.

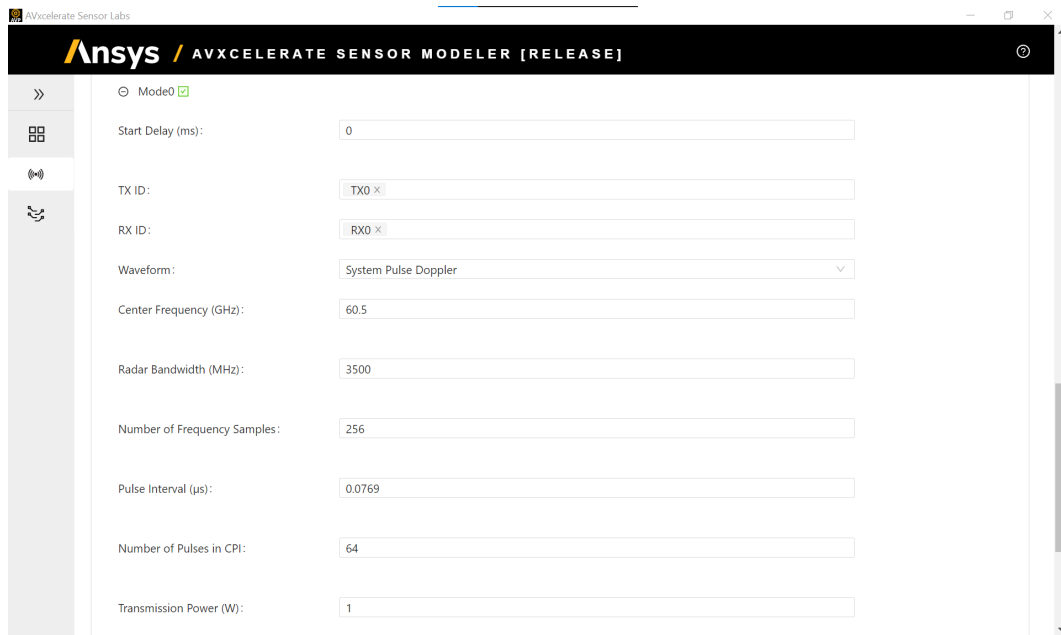


Figure 3.10: A111 Sensor parameters in AVx Sensor Labs

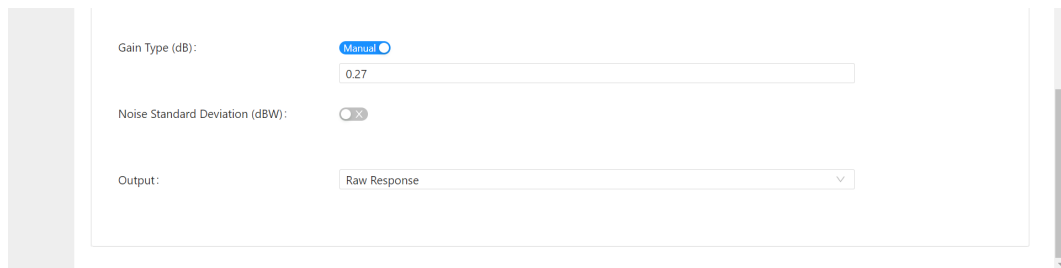


Figure 3.11: A111 Sensor parameters in AVx Sensor Labs

In the following step, a sensor layout was configured in the AVx Sensor Layout Configurator to include only the A111 radar sensor, as it is the sole sensor used in the simulation. Figure 3.12 displays the sensor layout window in AVx. As shown, the Translation and Orientation parameters are set to zero, positioning the corner reflector at the origin of the AVx coordinate system. This file was then saved to be used in the program to run the simulation.

With the sensor file prepared, the next step involved creating an AVx project, into which the 3D model of the corner reflector was imported as a geometry. This model was subsequently exported as an asset for use in the program to facilitate the simulation. Figure 3.13 illustrates the AVx virtual environment window.

A Python script that integrates all components of the simulation environment was executed to conduct the simulation. The number of frames for the simulation was set to 625, matching the number of frames generated by the actual radar measurements. The results of the simulation are presented as follows. Figure 3.14 illustrates the view of the static corner reflector as perceived by the A111 radar sensor through-

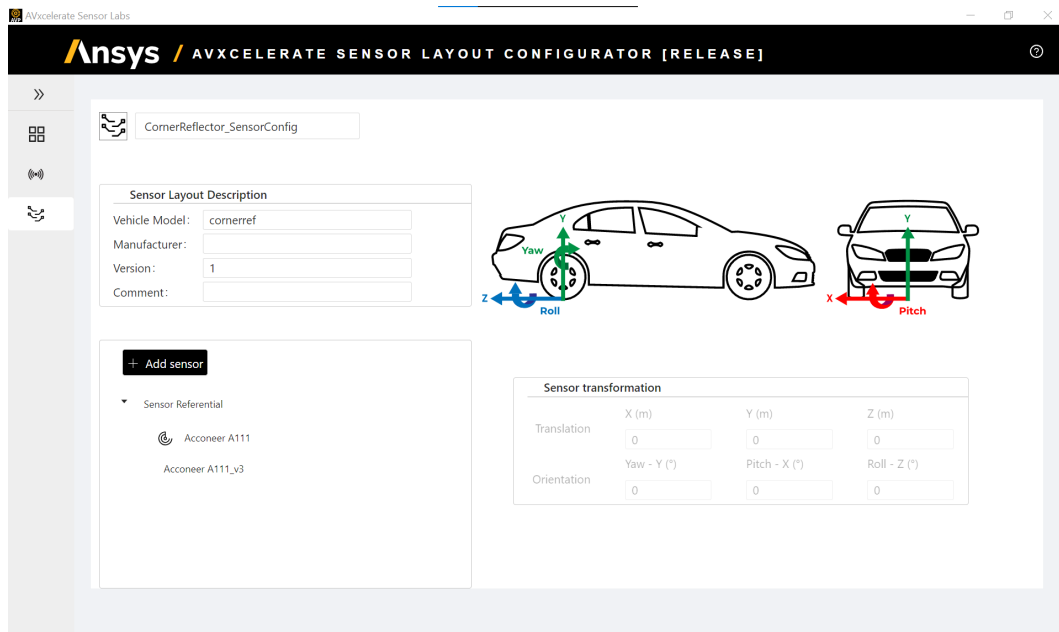


Figure 3.12: Sensor Layout for static corner reflector in AVx Sensor Labs

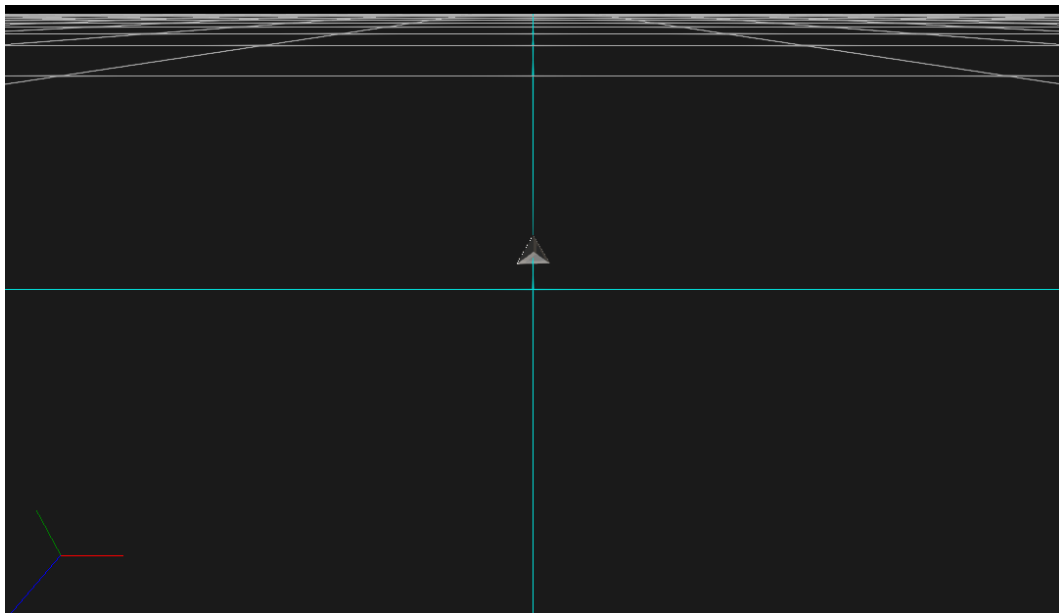


Figure 3.13: Stationary corner reflector in AVx Asset Prep Environment

out the simulation. Since the corner reflector is static, this view remains consistent across all frames. Figure 3.15 displays the output values from one frame of the simulation. A total of 625 frames were produced, with each frame containing 64 pulses, each represented by 256 samples. The values shown in Figure 3.15 correspond to the frequency domain representation of the received signal power where every 256 rows of the output file correspond to the received power of a pulse.

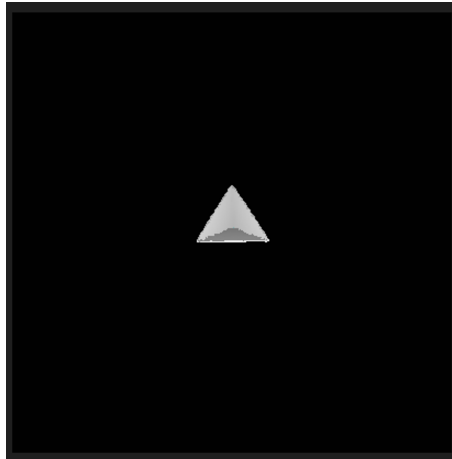


Figure 3.14: Static corner reflector as seen by the A111 radar in the simulation

```
Acconeer A111_000000010_Mode000_frequencypulse.txt - Notepad
File Edit Format View Help
1 1 64 256 0 0 -2.4608E-06 2.3839E-06 5.875E+10 6.223632812E+10
0.004695256706 -0.003205807181
0.004229680169 -0.003810048336
0.003686804092 -0.004347072449
0.003076273249 -0.004807060584
0.002408974105 -0.005181570072
0.001696828869 -0.005463688169
0.0009525925852 -0.005648157094
0.0001896167669 -0.005731476471
-0.0005783882807 -0.005711964332
-0.001337602502 -0.005589790177
-0.002074342221 -0.005366978236
-0.002775312169 -0.005047366489
-0.003427841468 -0.004636546131
-0.004020115361 -0.00414175773
-0.004541387782 -0.003571764566
-0.004982178565 -0.002936696401
-0.005334444344 -0.002247867407
-0.005591728259 -0.001517575583
-0.005749265198 -0.0007588770823
-0.005804091226 1.464148409E-05
-0.005755078048 0.0007891121786
-0.005602972116 0.001550634857
-0.005350369494 0.002285525901
-0.005001675338 0.002980565652
-0.004563032184 0.003623237135
-0.004042202141 0.004201952368
-0.003448430914 0.004706260283
-0.002702288549 0.005127039272
-0.002085472224 0.00545665808
-0.001340600778 0.005689115661
-0.0005709885154 0.005820154678
0.0002095962554 0.005847333465
0.0009871767834 0.00577007141
0.001747820177 0.005589671899
0.002477883827 0.005309276283
0.003164268564 0.004933836404
0.003794648685 0.004470008891
0.004357696977 0.003926042933
```

Figure 3.15: Simulation output for stationary reflector

3.2.2 Case 2- Target in Motion

To simulate the 'Target in Motion' condition, the steps taken for the stationary target were replicated, along with a few additional procedures. The following outlines the process undertaken.

An integrated 3D model of the ABB CRB 15000 robot and the corner reflector (used in the previous test case) was imported into AVx, along with a wooden rig, as

shown in Figure 3.20, which holds the A111 sensor and a flat surface or floor for the entire setup. Surface properties were meticulously added to replicate the dielectric characteristics of the actual materials used in the measurements. Given that this scenario involved the robot, the simulation was more complex than in the case of the stationary corner reflector. The following points outline the additional steps taken to manage the robot before running the simulation.

3.2.2.1 Getting ABB CRB 15000 ready for simulation

As mentioned previously, the joint states of the robot were captured throughout its motion using the Signal Analyzer feature in ABB RobotStudio. These joint states were then exported as an Excel file from ABB RobotStudio. Figure 3.16 displays a screenshot of the file. The joints J1 through J6 correspond to the six degrees of freedom of the robot arm, and the recorded values represent angles in degrees relative to the robot's base, sampled at intervals of 0.024 seconds.

Time	J1	J2	J3	J4	J5	J6	Current Wobj	Current Wobj	Current Wobj
0,000	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,024	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,048	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,072	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,096	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,120	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,144	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,168	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,192	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,216	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,240	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,264	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,288	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,312	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,336	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,360	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,384	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,408	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,432	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,456	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,480	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,504	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093
0,528	0,005	5,455	13,055	0,000	71,484	-179,909	602,829	0,054	561,093

Figure 3.16: Joint states of CRB15000 captured by Signal Analyzer

In Figure 3.7, the sensor configuration used in the physical measurements is shown. The A111 radar sensor operates at an update rate of 20 Hz, corresponding to a frame interval of 0.05 seconds. However, since the data recordings were acquired at intervals of 0.024 seconds, it was necessary to interpolate the joint state data to align with the radar's 20 Hz update rate.

Following the interpolation of the joint states, the next step involved converting the joint angles from degrees to radians. This conversion was implemented using a MATLAB script. The resulting interpolated and converted data was then saved in a JSON file for subsequent processing and analysis.

3. Method

The next critical step was to transform the joint state values from the interpolated data to align with the AVxcelerate coordinate system. This transformation was performed using Blender to ensure compatibility with the AVx system conventions.

ABB Robotstudio and Blender share a similar coordinate system where the Z-axis points upward. In both software environments, the rotation conventions are as follows: yaw around the Z-axis, pitch around the Y-axis, and roll around the X-axis. Specifically, yaw represents rotation around the vertical axis, producing clockwise or counterclockwise turns; pitch refers to rotation around the lateral axis, resulting in an upward or downward tilt; and roll involves rotation around the longitudinal axis, producing a side-to-side tilt.

In contrast, AVxcelerate (AVx) employs a Y-up coordinate system. Here, yaw is defined along the Y-axis, roll along the Z-axis, and pitch along the X-axis. Additionally, in AVx, the sensor is oriented to always face the positive Z direction. Thus, the axis mapping from Blender/ABB RobotStudio to AVx is shown in Table 3.1. Also, Figure 3.17 shows the robot in Blender with the coordinate system transformed from Blender's Z-up coordinate system to AVx's Y-up coordinate system.

Blender	AVx
X	-Z
Y	-X
Z	Y

Table 3.1: Axes mapping from Blender to AVx

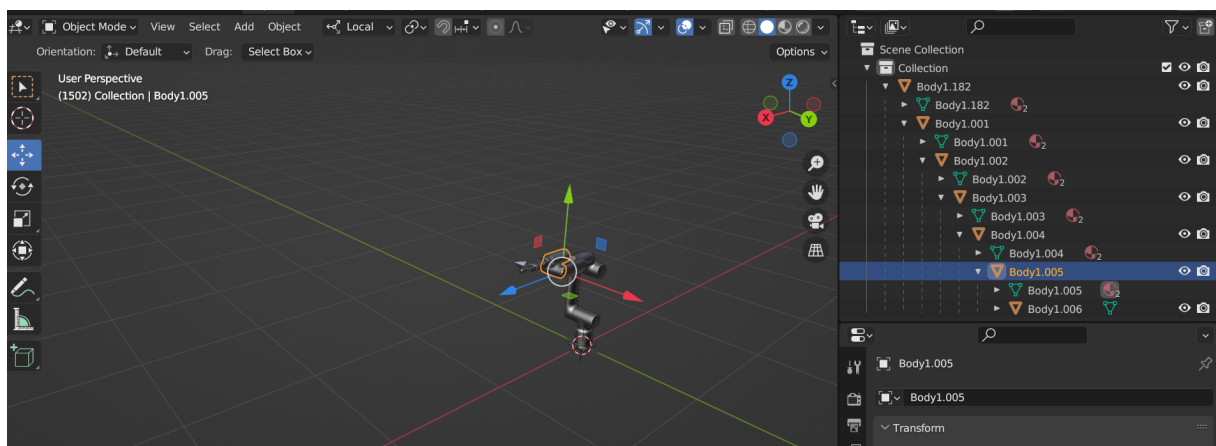


Figure 3.17: ABB CRB 15000 in Blender

To understand the conversion, it is essential to understand the following:

- **Local angles:** Local angles describe the rotation of each joint relative to its own coordinate system or to the coordinate system of the parent joint, not relative to the entire scene or world coordinate system.

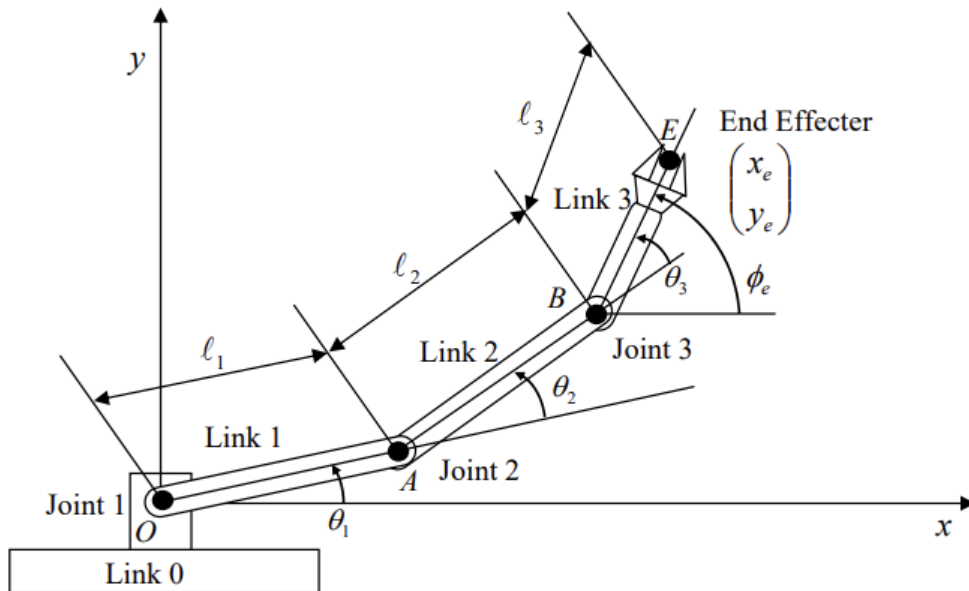


Figure 3.18: Local joint angles in a robot arm [11]

In this diagram, the robot is a 3-DOF arm with three revolute joints. Joint 1 forms an angle of θ_1 with the x-axis. θ_2 represents the angle between Link 1 and Link 2, specifically between the directions from O to A and from A to B. Similarly, θ_3 is the angle between Link 2 and Link 3, specifically between the directions from A to B and from B to E. Therefore, the local joint angles are θ_1 , θ_2 , and θ_3 , each denoting the angle of rotation at their respective joints. These angles define the configuration of the arm within its workspace.

- **Global coordinate system:** The global coordinate system is an absolute reference frame used to describe the position and orientation of objects in space. In robotics, this is often a fixed system, such as the world frame, that doesn't change with the robot's movements.
- **Global angles:** Global angles define the orientation of the end effector and the different joints of the robot relative to the fixed global coordinate system regardless of its local joint configurations.

The global equivalent of the local angles in a robotic arm, like the one shown in Figure 3.18, will be represented with respect to the global coordinate system

or reference frame that is typically fixed at the base of the robot, specifically at Joint 1 (point O). In this coordinate system:

- θ_1 represents the orientation of Link 1 with respect to the global frame centered at O.
- The global orientation of Link 2, which is the angle it makes with the global x-axis, is the sum of the angles θ_1 and θ_2 . This is because θ_2 represents the relative angle between Link 1 and Link 2, so it must be added to θ_1 to determine the orientation of Link 2 in the global frame. Therefore, the global angle of Link 2 is given by $\theta_1 + \theta_2$.
- The global orientation of Link 3, which is the angle it forms with the global x-axis, is the cumulative sum of the angles θ_1 , θ_2 , and θ_3 . Since θ_3 represents the relative angle between Link 2 and Link 3, it is added to determine the orientation of Link 3 in the global frame. Thus, the global angle of Link 3 is given by $\theta_1 + \theta_2 + \theta_3$, which also corresponds to the end-effector angle, θ_e , with respect to the global frame.

In Blender, the initial step involved setting each joint angle of the robot to zero. Subsequently, the coordinate axes at each joint were reoriented to align with a Y-up system, where the Y-axis points upward, the Z-axis points forward, and the X-axis points to the right. This configuration was chosen to synchronize the joint orientation with the AVx coordinate system, ensuring consistency across reference frames.

After configuring the joint orientations, frame-by-frame local rotations from the provided JSON file were applied to the robot's joints. These local rotations were then converted to global angles, effectively decoupling each joint's rotation from the rotations of its parent joints. This approach ensures that each joint's rotation remains independent, facilitating precise and consistent tracking of each joint's position throughout the animation sequence. Figure 3.21 shows a flowchart of the algorithm used.

With the robot's joint states aligned to the AVx coordinate system, the next phase involved running the simulation. All 3D geometries were imported into a new AVx project, and appropriate material properties were applied to each geometry to accurately represent their physical characteristics. Static objects, including the floor, wooden rig, and radar holder, were all exported as a single static asset to optimize the simulation setup.

The robot holding the corner reflector, was imported as a separate asset to allow independent manipulation and tracking. Additionally, two cameras were positioned strategically within the scene to provide optimal viewpoints for observing the simulation. Lighting elements were also added to ensure sufficient scene illumination, enhancing visibility and rendering quality throughout the simulation.

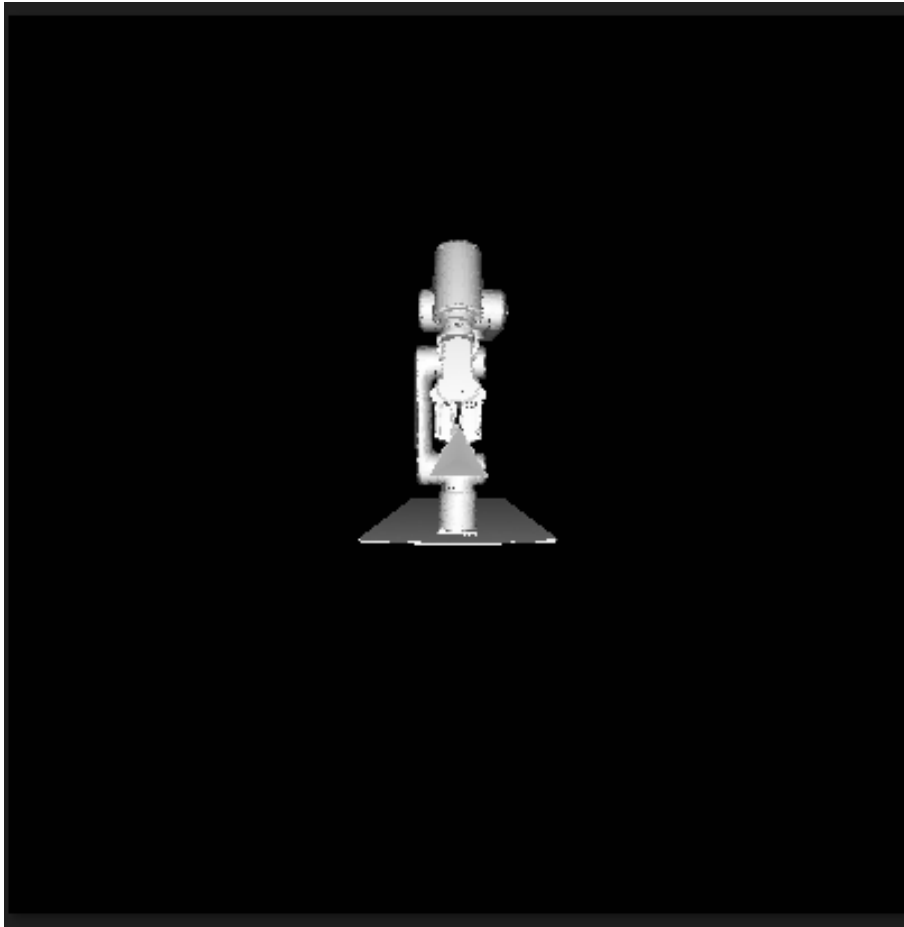


Figure 3.19: A111 sensor’s view in a simulation frame

A Python script integrating all elements of the simulation was executed, following a structure similar to the script used in the static corner reflector scenario. This script integrates three key components: the sensor definition file, the motion file, and the sensor parameters file.

The sensor definition file specifies the configuration of all sensors in the simulation, including the radar and cameras, along with their precise positions within the AVx project scene. The motion file, in JSON format, contains the converted joint states of the robot exported from Blender with each joint having corresponding yaw, pitch and roll values at every time step of motion, providing frame-by-frame rotational data to animate the robot accurately. Lastly, the sensor parameters file stores pre-defined parameters for each sensor, ensuring consistent sensor behavior throughout the simulation. The number of frames is not explicitly set to a specific value in this case; instead, it is determined by the time step defined in the joint angles JSON file. Figure 3.20 illustrates the AVx scene configuration for this simulation scenario.

Running this code initiates the simulation in AVx, generating outputs as shown in Figure 3.15, which displays complex received power values in the frequency domain.

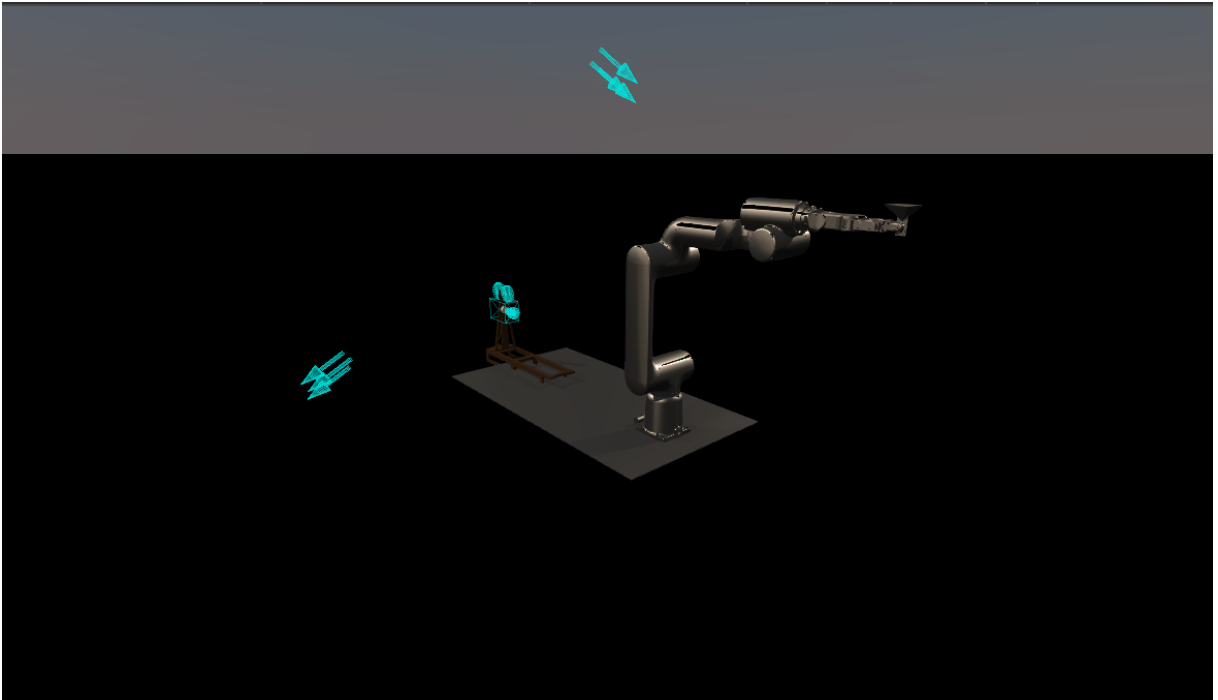
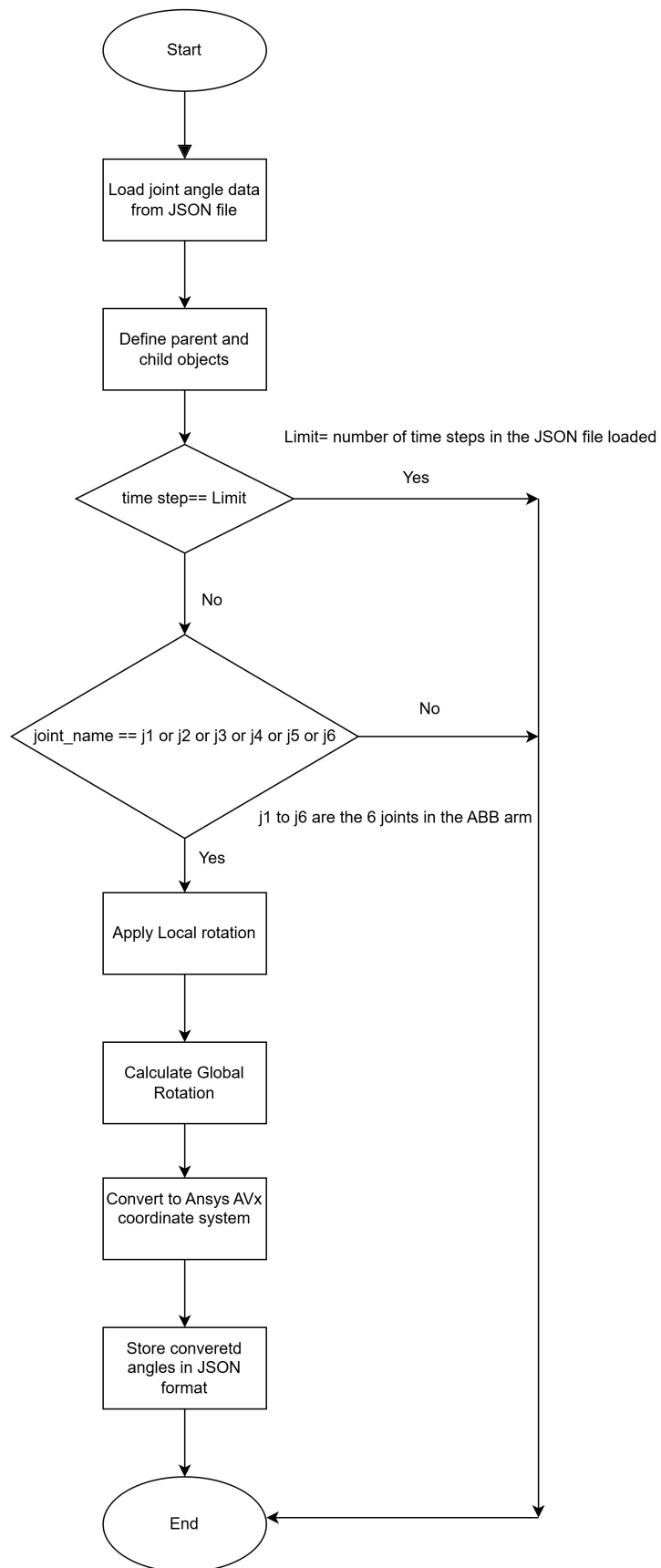


Figure 3.20: Simulation scene from AVx

The first 256 rows correspond to frequency samples for the first pulse, with each subsequent pulse also represented by 256 frequency samples. Each frame generated consists of a total of 64 pulses.

Figure 3.19 presents the debug camera view, illustrating what the radar 'sees' during the simulation, captured here for a single frame.

**Figure 3.21:** Blender flowchart

4

Results

The chapter describes the results compiled from both the simulation and the real radar measurements.

4.1 Results from Case 1- Stationary Target

The Data from the Real Radar measurements is as shown in the figure below.

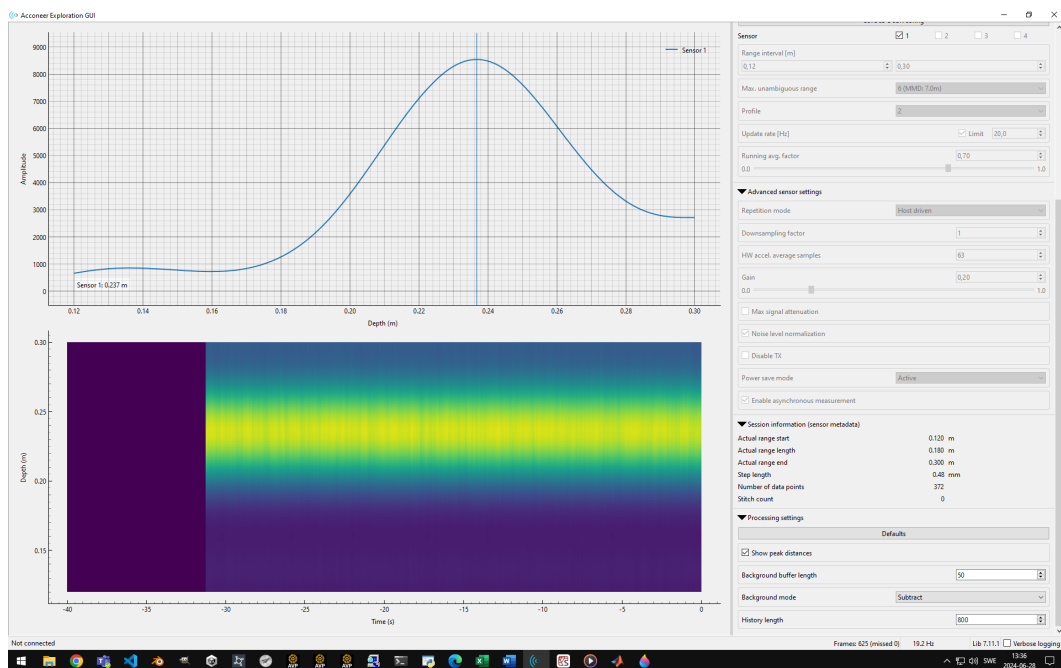


Figure 4.1: Radar Range plot from Real Radar Data

Since the target, a stationary corner reflector, remains fixed, the Range plot shows a single peak corresponding to its position. In the plot, the X-axis represents the range in meters, while the Y-axis displays discrete digital values computed by the Signal Processor in the A111 radar sensor. The peak's position along the X-axis indicates the corner reflector's location, measured here at 0.237 meters (or 23.7 centimeters), with a corresponding Y-axis value of 8590.

4. Results

Given the same scenario where the target remains stationary, the simulation data is shown in the figure below.

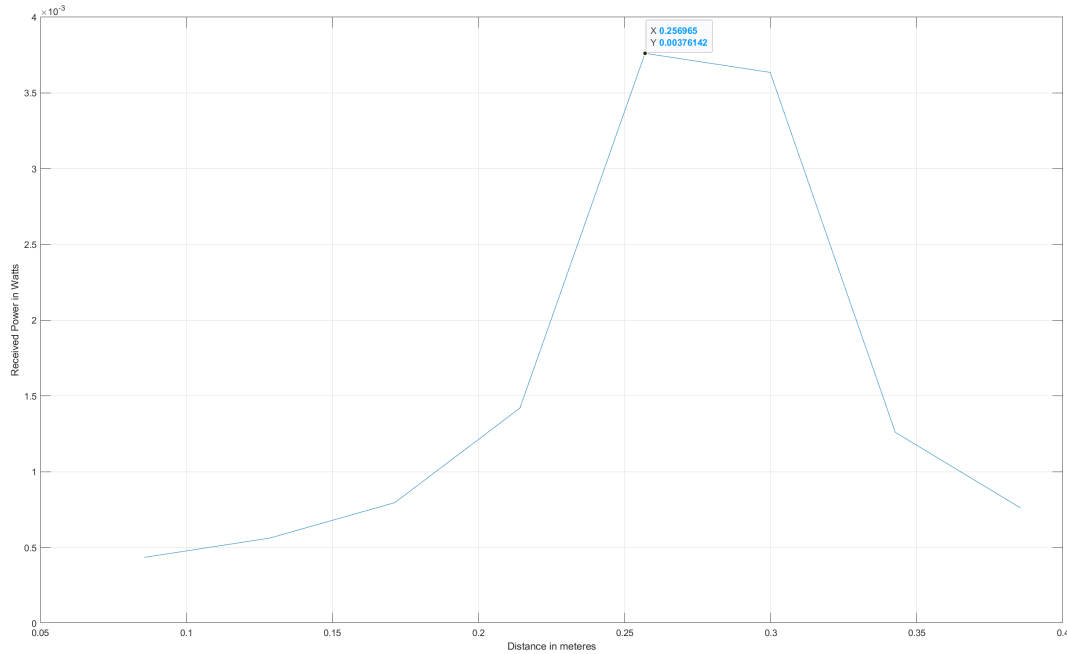


Figure 4.2: Radar Range plot from Simulated Data

The Range plot displays range values in meters on the X-axis and received power in watts on the Y-axis. The peak in the plot, representing the position of the corner reflector, appears at 0.257 meters (25.7 centimeters), which closely matches the value obtained from real radar measurements. Since the range values in both the real and simulated readings are nearly identical, it is reasonable to equate the magnitudes of their corresponding peak values.

$$\text{Magnitude of peak from Real Data} = 8590$$

$$\text{Magnitude of peak from Simulated Data} = 0.00376142$$

Thus, the scaling factor is calculated as:

$$\text{Scaling factor} = \frac{0.00376142}{8590} = 0.43737 \times 10^{-6} \quad (4.1)$$

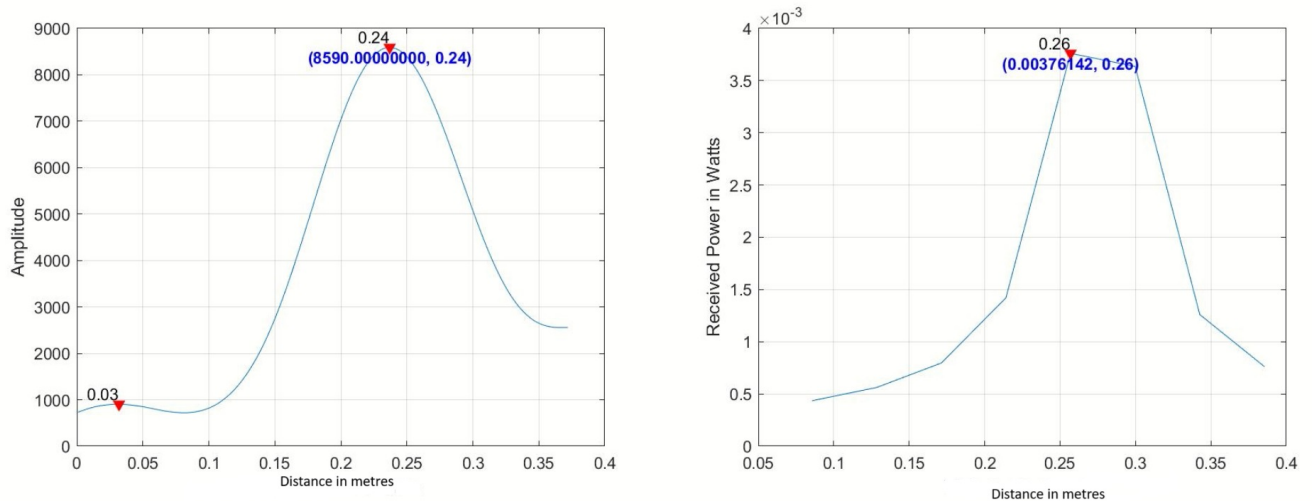


Figure 4.3: Radar Range plots from Simulated Data and Measurements for Stationary corner reflector

4.2 Results from Case 2- Target in Motion

4.2.1 Results using Sparse format

As discussed in Section 3.1.2, the Acconeer Exploration Tool, which serves as Acconeer’s sensor configuration and measurement recording interface, offers multiple formats for recording measurements. The initial results presented were based on the Sparse format. Figures 3.8 and 3.9 in Section 3.1.2 illustrate this format.

To recap, the signals are sampled at 6 cm intervals, yielding a number of sweeps denoted as N_s . Each sweep consists of data sampled at these 6 cm intervals, with T_s as the time between each sweep or bin.

Figure 4.4 presents the bin values derived from the physical measurements. Through research and analysis, it was determined that these values represent the amplitude (or magnitude) of the reflected radar signals received from each range bin. Although Acconeer has not provided specific documentation regarding this, it is reasonable to infer that these values are generated by the radar’s analog-to-digital converter (ADC), which digitizes the analog signals of the radar reflections. Typically, higher amplitude values indicate stronger reflections, which may correspond to objects that are either closer to the radar or possess greater reflectivity.

Figure 4.5 displays the bin values obtained after processing the raw output from the AVX simulation. As shown in Figure 4.6, the raw output consists of a range of complex values representing the received power in the frequency domain. The following steps summarize the data processing procedure:

The simulation produced multiple `.h5` files, with each file representing a distinct frame.

Each frame, or `.h5` file, contains multiple rows of complex received power values. Given that the number of frequency samples was set to 256 and the number of pulses in the coherent processing interval (CPI) was set to 64, the total number of rows in each `.txt` file is $64 \times 256 = 16,384$. This means that each batch of 256 rows corresponds to a single pulse.

The frequency domain data was converted to the time domain by applying an inverse fast Fourier transform (IFFT). This time domain representation reflects the instantaneous power received at each time instance. So, the complex values obtained represent the amplitude and phase information of the signal at discrete frequency components. Each complex value can be expressed as:

$$C_k = |C_k|e^{j\phi_k} \quad (4.2)$$

where $|C_k|$ is the magnitude (amplitude) and ϕ_k is the phase angle of the frequency component.

The IFFT takes these complex frequency-domain samples and reconstructs them into a time-domain signal. Mathematically, the IFFT is defined as:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} C_k e^{j\frac{2\pi}{N}kn} \quad (4.3)$$

where:

- N is the total number of frequency samples.
- $x[n]$ is the time-domain signal at sample n .
- C_k are the complex frequency-domain samples.

The resulting time-domain signal $x[n]$ represents the **instantaneous value of the signal** at each time sample n . When analyzing power, the resulting signal can be interpreted as the **instantaneous power response** of the system, varying over time based on the contributions from all frequency components.

At the end of this process, the instantaneous power response for every pulse in every frame was obtained.

The next step involved obtaining the range values. To achieve this, the time indices axis from the IFFT plot was multiplied by the range resolution of the radar system. Range R is,

$$R[n] = \frac{c \cdot t[n]}{2} \quad (4.4)$$

where:

- $R[n]$ is the range corresponding to the time sample n ,
- $t[n]$ is the time associated with the n -th sample of the IFFT output.

This was done for all the pulses in every frame.

The next step involved selecting power values for each pulse and frame at 6 cm intervals, adhering to the Sparse format defined by Acconeer.

It was observed that all pulses within each frame produced identical power values since it is an ideal system with no noise. Consequently, averaging the power values would yield the same result as selecting the power values from any single pulse.

These power values were then stored in 'power bins' for each frame as shown in Figure 4.5.

With the datasets prepared from both the simulation and physical measurements, the next step involved comparing these two sets of data. To facilitate this comparison, the idea was to use the following relation, Power and Amplitude are related as follows:

$$P_{\text{calculated}} = k \cdot (A^2) \quad (4.5)$$

where:

- $P_{\text{calculated}}$ is the calculated power.
- k is the scaling factor that relates amplitude to power.
- A is the amplitude of the received signal.

The scaling factor k is a constant that must be determined during the calibration process, which was previously addressed in Equation 4.1.

This approach was unsuccessful because the ADC values could not be converted into equivalent amplitude values, which are necessary for calculating power using Equation 4.5.

Another possibility could have been the use of the following equation,

$$P_{\text{received}} = \frac{\left(\frac{\text{ADC}_{\text{value}}}{2^N - 1} \times V_{\text{ref}}\right)^2}{R \cdot G_{\text{linear}}^2} \quad (4.6)$$

where,

- P_{received} : Received power in watts.
- $\text{ADC}_{\text{value}}$: Raw digital output from the ADC.
- V_{ref} : Reference voltage of the ADC (in volts).
- N : Resolution of the ADC in bits (16 in this case).
- $G_{\text{linear}} = 10^{\frac{G}{10}}$: Gain of the radar system in linear scale, where G is the gain in decibels (dB).

4. Results

- R : Impedance of the radar receiver (typically in ohms, e.g., 50 Ω).

This equation requires detailed knowledge of the A111 hardware, information that is not available in the Acconeer documentation.

1	2	3	4	5	6	7	8	9	10	11	12	13
30850	30752	30728	30616	30678	30808	30721	30799	30548	30539	30653	30724	30638
30749	30598	30637	30893	30684	30604	30758	30797	30728	30694	30824	30526	30702
30448	30420	30316	30353	30340	30386	30332	30455	30418	30290	30292	30509	30377
30490	30536	30518	30413	30443	30502	30602	30430	30528	30484	30450	30567	30501
30305	30190	30245	30270	30230	30265	30446	30175	30322	30243	30303	30239	30316
30302	30362	30376	30378	30431	30382	30457	30153	30310	30305	30423	30372	30416
30770	30780	31163	31238	31682	31711	32002	32201	32211	32639	32673	33010	32891
40229	39987	39750	39837	40080	39866	39591	39551	39351	39051	38934	38559	38472

Figure 4.4: Sparse result from physical measurements

8x64 double												
1	2	3	4	5	6	7	8	9	10	11	12	13
6.7519e-05	6.7519e-05	6.7519e-05	6.7519e-05	6.7519e-05	6.7519e-05	6.7519e-05	6.7519e-05	6.7519e-05	6.7519e-05	6.7519e-05	6.7519e-05	6.7519e-05
7.9019e-05	7.9019e-05	7.9019e-05	7.9019e-05	7.9019e-05	7.9019e-05	7.9019e-05	7.9019e-05	7.9019e-05	7.9019e-05	7.9019e-05	7.9019e-05	7.9019e-05
9.4357e-05	9.4357e-05	9.4357e-05	9.4357e-05	9.4357e-05	9.4357e-05	9.4357e-05	9.4357e-05	9.4357e-05	9.4357e-05	9.4357e-05	9.4357e-05	9.4357e-05
1.1401e-04	1.1401e-04	1.1401e-04	1.1401e-04	1.1401e-04	1.1401e-04	1.1401e-04	1.1401e-04	1.1401e-04	1.1401e-04	1.1401e-04	1.1401e-04	1.1401e-04
1.7690e-04	1.7690e-04	1.7690e-04	1.7690e-04	1.7690e-04	1.7690e-04	1.7690e-04	1.7690e-04	1.7690e-04	1.7690e-04	1.7690e-04	1.7690e-04	1.7690e-04
6.7421e-04	6.7421e-04	6.7421e-04	6.7421e-04	6.7421e-04	6.7421e-04	6.7421e-04	6.7421e-04	6.7421e-04	6.7421e-04	6.7421e-04	6.7421e-04	6.7421e-04
2.6769e-04	2.6769e-04	2.6769e-04	2.6769e-04	2.6769e-04	2.6769e-04	2.6769e-04	2.6769e-04	2.6769e-04	2.6769e-04	2.6769e-04	2.6769e-04	2.6769e-04
1.2387e-04	1.2387e-04	1.2387e-04	1.2387e-04	1.2387e-04	1.2387e-04	1.2387e-04	1.2387e-04	1.2387e-04	1.2387e-04	1.2387e-04	1.2387e-04	1.2387e-04

Figure 4.5: Sparse result from AVx simulation

4.2.2 Results using Envelope format

Since the sparse format could not be used for evaluation, the next best option was the envelope format. As discussed in Section 3.1.2, the envelope format provides a filtered amplitude representation of radar reflections, outputting amplitude values at various distances. To evaluate the results with this format, two different approaches were considered.

4.2.2.1 Method 1

In this approach, the goal was to generate range plots for the first pulse in each frame, as shown in Figure 4.8, and compile these plots into a video. This video would then be compared with a similar compilation created from the plots of physical measurements.

The plot generation algorithm closely follows the one used for the sparse format, with only minor adjustments. The process is summarized in the following steps:

The same raw output file from the AVx simulation used in the sparse format was processed, producing multiple .txt files, each representing a single frame. The frequency domain data in each file was then converted to time domain data by applying

```

1 1 64 256 0 0 -2.4608E-06 2.3839E-06 5.875E+10 6.223632812E+10
-0.002198661445 -0.004074266646
-0.003494828008 -0.003045921912
-0.004313590005 -0.001625110046
-0.004557961598 -3.652132727E-05
-0.004235261586 0.001492328476
-0.003435638268 0.002775099361
-0.002296136692 0.003689461388
-0.0009648432606 0.004176183138
0.0004204279103 0.004221849144
0.001734390273 0.003841914004
0.002858646447 0.003077367553
0.003684597556 0.002002003836
0.004129364621 0.0007248045877
0.004152967595 -0.0006231229636
0.003761941334 -0.001911685918
0.00299620186 -0.003029334126
0.00191094121 -0.003882860532
0.0005723818322 -0.004382475745
-0.0009229173302 -0.004432919901
-0.002420399105 -0.003954501823
-0.003706093179 -0.002933824668
-0.00455294596 -0.001469696406
-0.004796110094 0.0002266545343
-0.004391296301 0.001885222388
-0.003423655871 0.003253805451
-0.002070679329 0.004157638643
-0.0005456772633 0.004522290081
0.0009524545749 0.00436290307
0.002268592129 0.003753996687
0.003296637209 0.002797395224
0.003969698213 0.001602713601
0.004250803962 0.0002817324421
0.004130619112 -0.001054746215
0.003623663913 -0.002303944435
0.002757307375 -0.003372085514
0.001567548024 -0.004160042387
0.0001204075597 -0.004549343605
-0.001450374606 -0.004420462996
-0.002930859104 -0.00371141918
-0.004072265234 -0.002478257054
-0.004676160868 -0.0009040019355
-0.004662090447 0.0007577491342
-0.004075630568 0.002265124815
-0.00304486067 0.003444214351
-0.001725325477 0.004198676907
-0.0002669490641 0.004488611128
0.001190622803 0.00430813944
0.002510883613 0.003680320457

```

Figure 4.6: AVx Simulation Raw output for target in motion

an Inverse Fast Fourier Transform (IFFT).

This Inverse Fourier transform generated plots of instantaneous power values for each pulse within every frame as shown in Figure 4.9.

Range values for each pulse from the simulation were computed using Equation 4.4, and it was observed that the range vector was consistent across all pulses within a single frame because it is an ideal system with no noise. Therefore, to create the range plots for the video compilation, the range vector from one pulse per frame was used.

To ensure consistency with the simulated data, slight modifications were made to the recorded data from the .h5 file generated by the Acconeer Exploration tool. Specifically, the X-axis was scaled to represent range in meters, aligning the recorded data with the simulated data in terms of distance measurements.

To facilitate analysis, peaks in each plot were identified and highlighted as shown in Figures 4.7 and 4.8.

4. Results

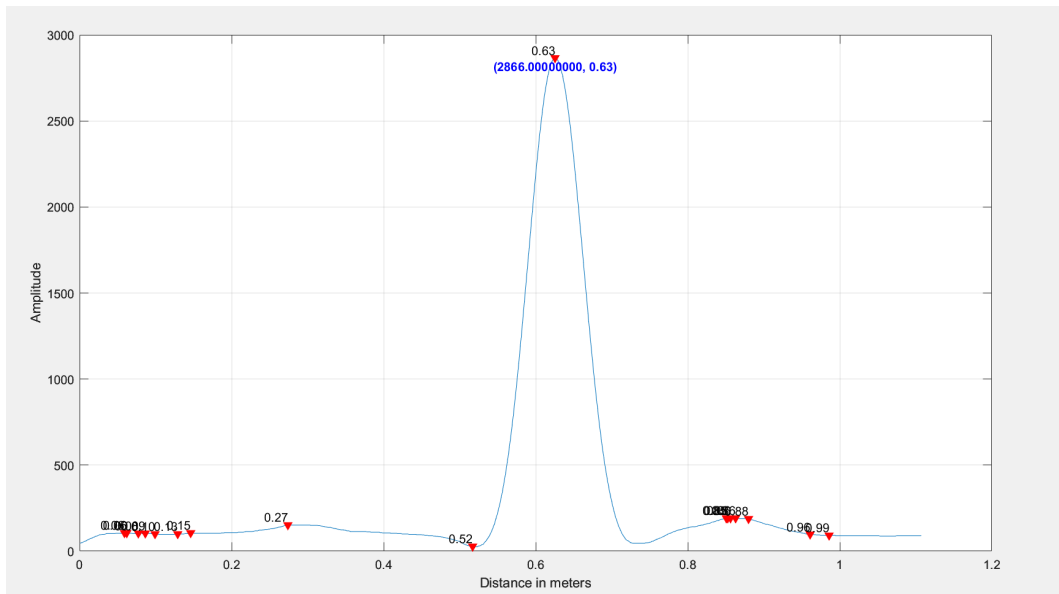


Figure 4.7: First frame from the real motion

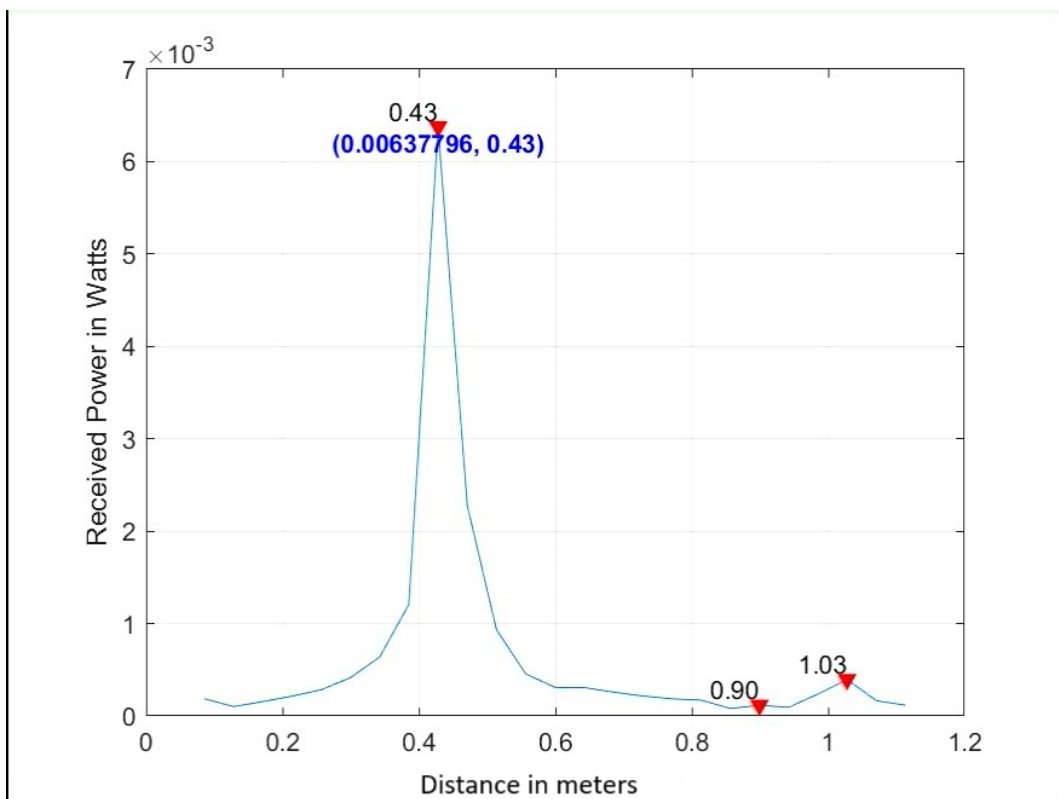


Figure 4.8: First frame from the simulated motion

The objective was to analyze the motion patterns in each video and determine whether the peaks aligned across both datasets. Although the Y-axes differ—as shown in Figures 4.7 and 4.8., where the simulation data represents Received Power and the measurement data represents 'Amplitude' values as defined by Acconeer—the goal was to identify peaks at corresponding ranges and assess whether the scaling

factor from Equation 4.1 remained consistent.

This method was set aside due to a considerable difference in frame counts between the simulation and the real data. While it was possible to address this through interpolation or by rerunning the simulation with a modified frame count, playing the video compilations individually revealed differences in range values (as discussed in the next subsection) and, more importantly, in motion patterns. These discrepancies indicated that a frame-by-frame comparison would not yield reliable results.

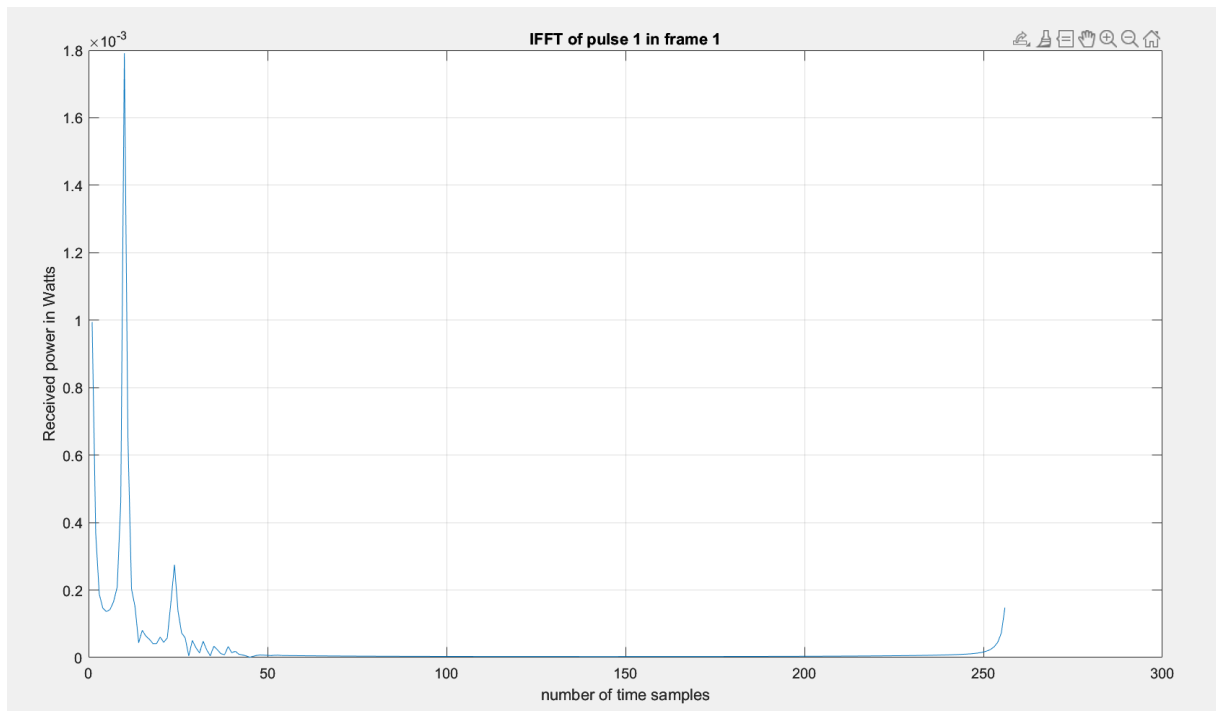


Figure 4.9: Inverse Fourier Transform plot for Pulse 1 in Frame 1 of Simulation

4.2.2.2 Method 2

Analysis of the previous method revealed differences in motion patterns between the simulated and real data. To verify these differences and gain visual insights, range plots were generated for both datasets. This was done by extracting location of the peak values of received power (or amplitude) from each frame.

This algorithm is developed based on earlier methods, with the process summarized in the following steps.

Starting with the raw output files from the AVx simulation—where each .txt file represents the frequency-domain returns for a single frame—time-domain data was derived by applying an Inverse Fast Fourier Transform (IFFT).

The Inverse Fourier Transform yielded instantaneous power values for each pulse within every frame.

Range values for the simulation were calculated using Equation 4.4. To ensure consistency, the X-axis of the real data was also scaled to represent range in meters, as described in Method 1.

Once range values were calculated across all frames for both simulated and real data, peaks in received power were identified.

The range values corresponding to the identified peaks were then plotted across all frames for both the simulation and real data, as shown in Figures 4.14 and 4.11.

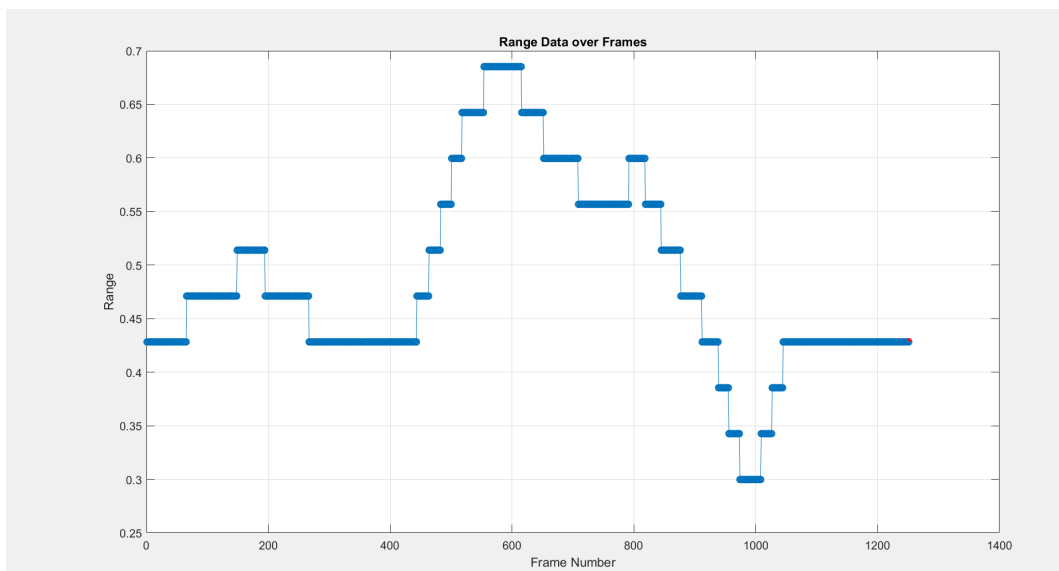


Figure 4.10: Frame versus Range plot- Simulation

Figure 4.14 illustrates the simulation results. While there are noticeable differences in appearance and range values between the graphs, the overall patterns remain

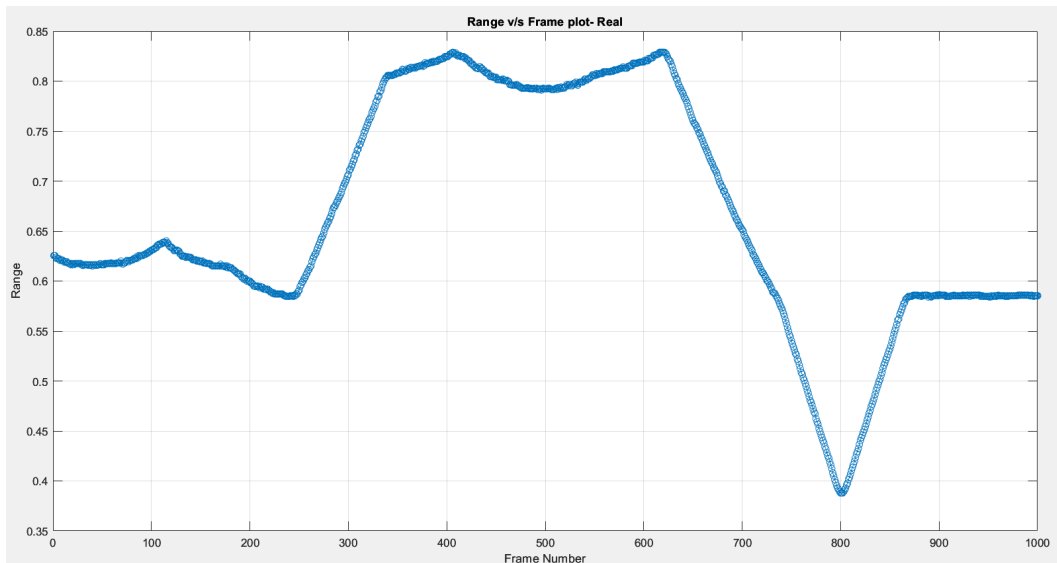


Figure 4.11: Frame versus Range plot- Real motion

quite similar. A closer look at the simulation plot reveals that the radar moves in discrete steps of 0.0428 meters, which aligns with the standard range resolution of the A111 sensor. Range resolution is defined as the minimum distance between two objects that enables the radar to distinguish them as separate targets. This is directly related to the system’s bandwidth, which, in this case, is 3.5 GHz. The range resolution is calculated as follows:

$$\text{Range Resolution} = \frac{3 \times 10^8 \text{ m/s}}{2 \times 3.5 \times 10^9 \text{ Hz}} = \frac{3 \times 10^8}{7 \times 10^9} \text{ m} = 0.042857 \text{ m} \quad (4.7)$$

The radar plot derived from real measurements shows a continuous movement pattern, suggesting that the radar is operating with finer steps than its conventional range resolution of 0.0428 meters. This is supported by the sensor configuration data from the Acconeer exploration tool, which indicates a sampling distance of 0.48 mm as highlighted in Figure 4.12—implying that the radar achieves a range resolution on the order of millimeters. Achieving such high range resolution with the system settings shown in Figures 3.10 and 3.11 would typically be unattainable using standard radar processing techniques, where range resolution is directly tied to the bandwidth of the system.

Further investigation revealed that the Acconeer A111 radar utilizes a unique signal processing algorithm, enabling its range resolution to not be dependent entirely on the system bandwidth. Although the exact signal processing method used by the radar is not fully clear, it is speculated that the radar utilizes a unique correlation-based technique. By correlating the received signal with delayed versions of the transmitted signal and leveraging phase information, the radar appears able to detect even minute changes in distance. This method allows for enhanced resolution and continuity in tracking, which likely explains the radar’s apparent ability to resolve movement at sub-millimeter accuracy.

4. Results

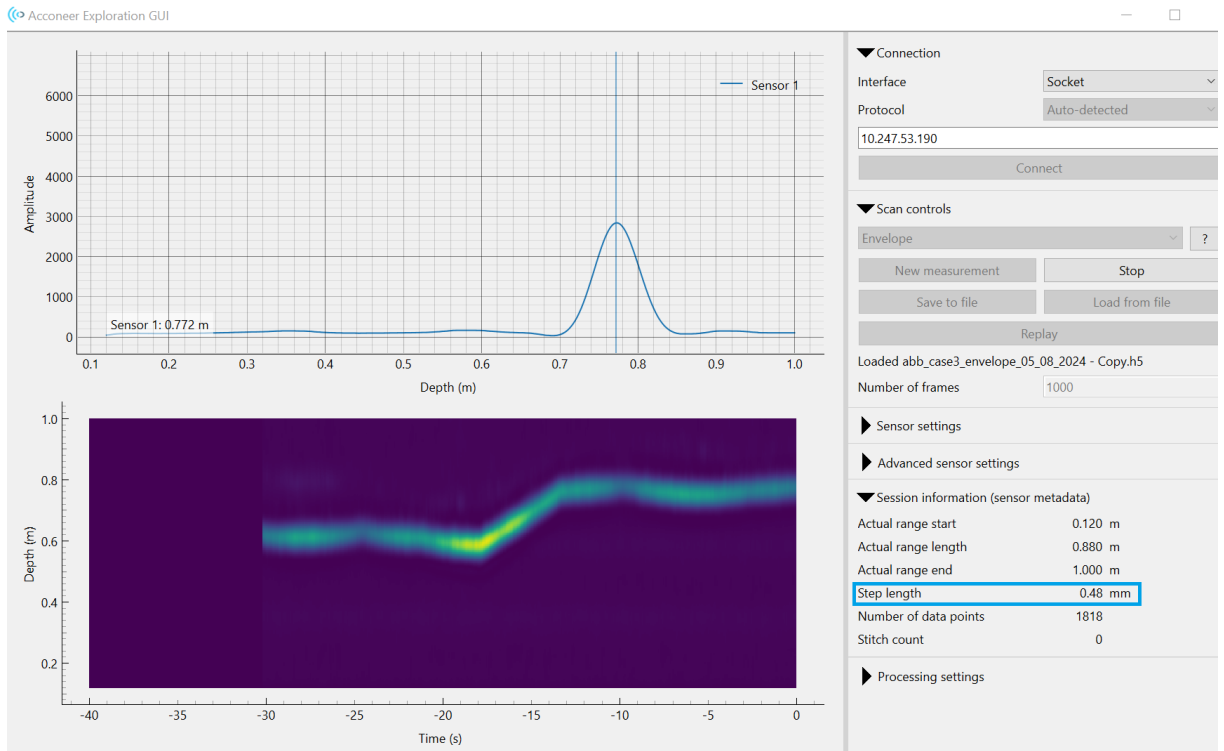


Figure 4.12: Sensor Metadata Information from real motion recording

Distance from phase can be calculated as follows,

For a sinusoidal wave with frequency f , the transmitted radar signal can be represented as:

$$S(t) = A \cos(2\pi ft + \phi_0) \quad (4.8)$$

where:

- A is the amplitude of the signal,
- f is the frequency,
- t is time,
- ϕ_0 is the initial phase of the signal.

After transmitting, this signal travels to the target and reflects back, introducing a time delay Δt in the received signal. This delay causes a phase shift $\Delta\phi$ relative to the transmitted signal.

The phase shift $\Delta\phi$ due to a time delay Δt can be determined based on the relationship between time and phase:

- Each full cycle of the wave (one period T) corresponds to a phase change of 2π radians.
- The frequency f of the signal is related to the period by $T = \frac{1}{f}$, so the signal's phase changes by 2π radians every T seconds.

The phase change $\Delta\phi$ corresponding to a time delay Δt can be expressed as the fraction of the wave's period that Δt represents, multiplied by 2π :

$$\Delta\phi = 2\pi \cdot \frac{\Delta t}{T} \quad (4.9)$$

Substitute $T = \frac{1}{f}$ to get:

$$\Delta\phi = 2\pi \cdot \Delta t \cdot f \quad (4.10)$$

which simplifies to:

$$\Delta\phi = 2\pi f \Delta t \quad (4.11)$$

This equation shows that the phase shift $\Delta\phi$ is proportional to the frequency f and the time delay Δt .

In radar systems, the time delay Δt is related to the distance d to the target. Since the radar signal travels at the speed of light c , the round-trip time delay Δt is:

$$\Delta t = \frac{2d}{c} \quad (4.12)$$

Substituting this expression for Δt into the phase shift equation gives:

$$\Delta\phi = 2\pi f \cdot \frac{2d}{c} = \frac{4\pi df}{c} \quad (4.13)$$

Thus, the phase shift $\Delta\phi$ is directly related to the distance d to the target, the frequency f of the radar signal, and the speed of light c .

Therefore,

$$\Delta\phi = \frac{4\pi df}{c} \quad (4.14)$$

Achieving such high range resolution without knowing the exact algorithm Acconeer uses was the next challenge. None of the sensor parameters shown in Figures 3.10 and 3.11 could be changed since it would then no longer represent the original sensor used in the real measurements. After some study and experimentation, it was noticed that the results from the simulation closely matched the graph from the physical measurements when the length of the raw data was increased. The following steps summarize the algorithm used.

As illustrated in Figure 4.14, the range values in the initial simulation exhibited an offset of approximately 20 cm when compared to real measurements. Upon further investigation, it was determined that this discrepancy was due to a 20 cm positional offset in the radar configuration within the simulation. To address this, the radar's position was corrected as shown in Figure 4.13, and the simulation was subsequently rerun to align more closely with the real-world measurements.

Using the updated raw simulation output, which provided complex received power values in the frequency domain, an inverse Fourier transform was applied to convert the data into the time domain.

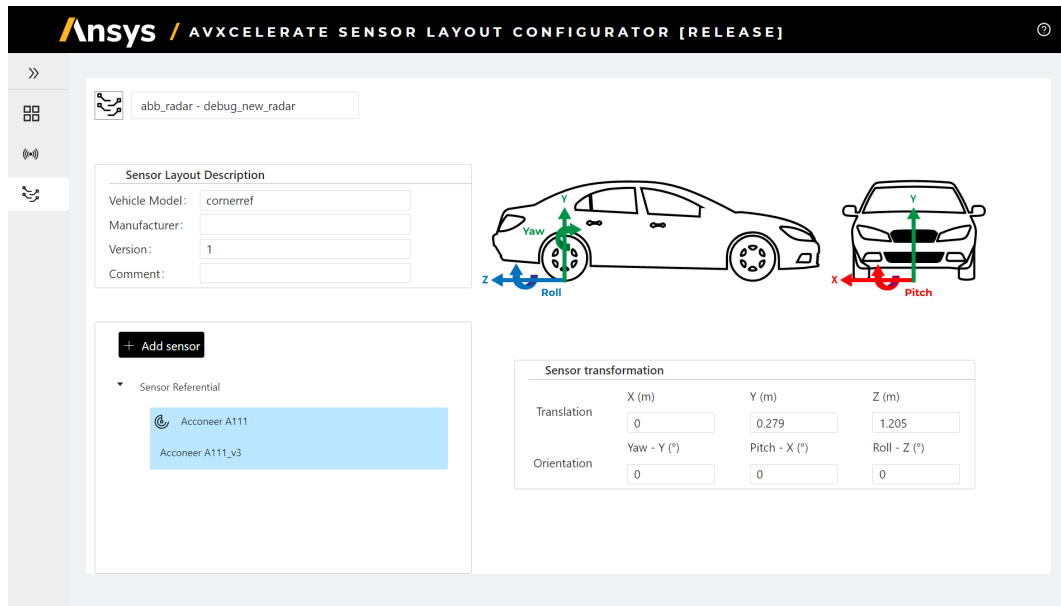


Figure 4.13: Sensor layout for target in motion with adjusted radar position

During this conversion to time domain, the frequency domain data was first 'Zero Padded'.

Zero padding involves adding zeros to the end of a signal in the frequency domain before performing an IFFT. If the original signal has N samples, the data is extended by adding zeros to reach a larger number of samples, say M (where $M > N$), before applying the IFFT. This padding does not add any new frequency content to the signal—it simply increases the number of points in the time domain after the IFFT.

This higher sample count adds more data points between peaks, which makes the resulting curve appear smoother when plotted. More frequency bins are created which makes the peaks appear sharper and allows better estimation of frequencies. This therefore helps with identifying the center of a target's peak with more precision within the range resolution limit. This method therefore helps in determining a target's location with greater precision.

Range was then calculated for all the pulses by multiplying the resultant time indices axis with the new apparent range resolution.

The locations of received power peaks for each frame were then plotted, as illustrated in Figure 4.14.

This plot now shows a continuous profile, closely resembling the measurement data depicted in Figure 4.11. The range values align closely between the simulation and the measurements, and the motion trajectory is similarly consistent, though with some observable differences.

The observed discrepancy can be attributed to the movement of the joints in AVx.

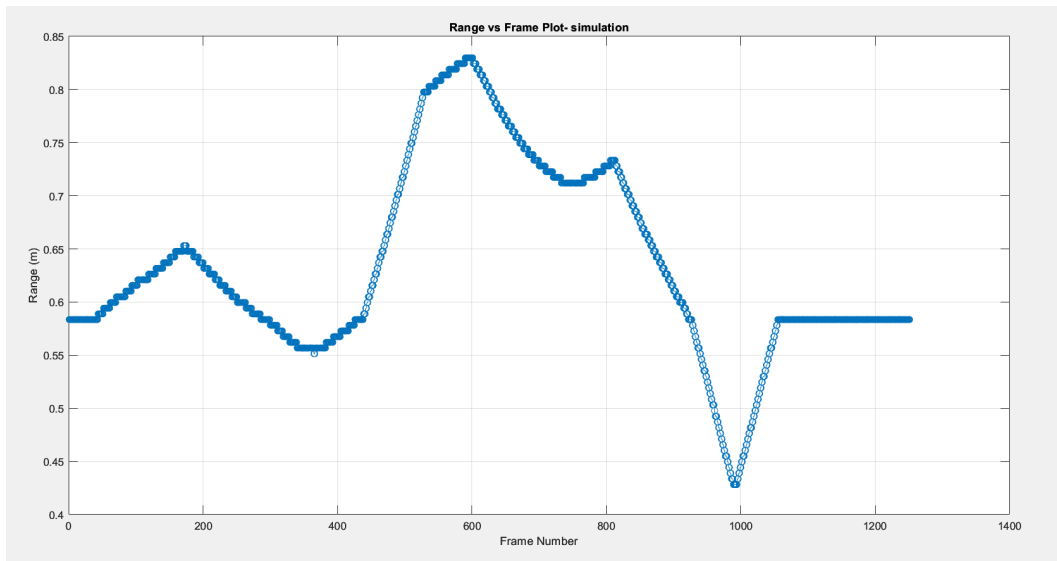


Figure 4.14: Frame versus Range plot- Simulation

As described in the previous section, the robot's motion is defined by the joint states of its arm throughout its trajectory. During the conversion from the Z-up coordinate system to AVx's Y-up coordinate system, as well as the transformation of local angles to global angles for AVx, it was noted that a certain joint combination did not particularly align as expected with the conversion process. While the simulation suggested that the motion of the corner reflector closely matched that in the real world, the differences in motion trajectories, as seen in Figures 4.11 and 4.14, suggest that the misalignment of the joint combination may have contributed to the discrepancies. Given the time-consuming nature of the process, which requires extensive effort and experimentation, it was decided to retain the simulation in its current form.

Considering the similarity between the two plots as shown in Figure 4.15, it is reasonable to conclude that AVx is accurately simulating the Acconeer A111.

4. Results

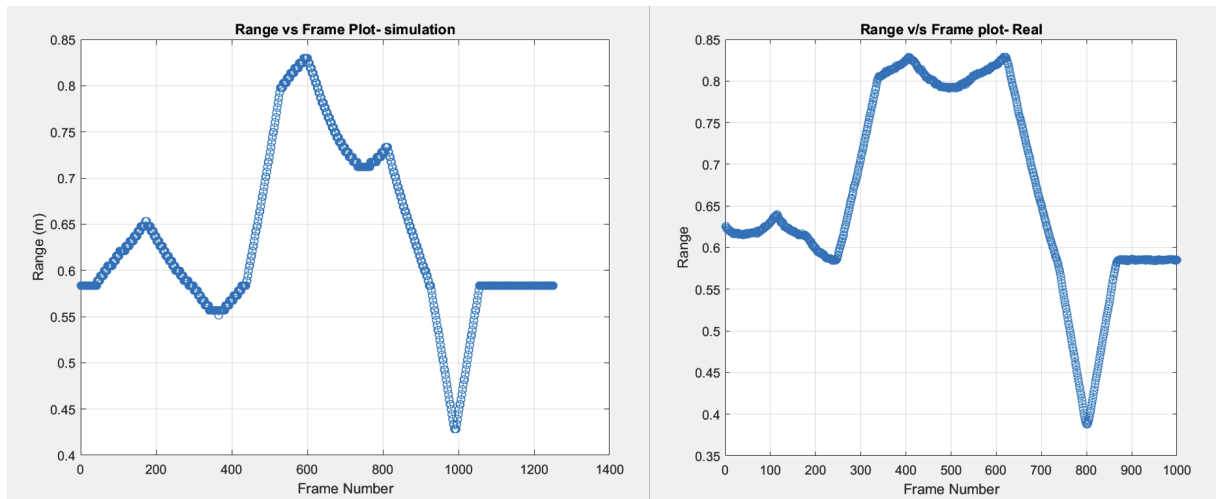


Figure 4.15: Range versus Frame plots- Simulation v/s Real motion

5

Conclusion and Future Work

5.1 Conclusion

This Master's thesis presents a thorough approach to evaluating simulation results from Ansys AVxcelerate, aimed at achieving both accuracy and consistency. The thesis systematically explores various alternative methods for evaluating these results, critically examining each in terms of its methodology and identifying their limitations. After a detailed assessment, the approach that produced the most reliable outcomes was selected as the optimal solution. Since the simulations needed to be benchmarked against predefined standards, special care was taken to ensure that real-time measurements were both accurate and easily reproducible in the simulation environment, allowing for a precise and meaningful comparison.

5.2 Evaluation and Findings

The results from both stationary and moving target scenarios highlighted several critical insights:

Accuracy of Simulated Data: For the stationary target scenario, the AVxcelerate simulation demonstrated a close alignment with the real radar data, particularly in terms of range accuracy and radar reflections. This confirms that the simulation software effectively replicates simple, static scenarios with high precision.

Challenges in Dynamic Scenario: In the moving target scenario, minor differences were observed between the simulated and real radar data, despite using frame interpolation in the simulation and closely aligning with the real-time data's sampling intervals. These discrepancies were mainly due to small variations in motion between the real measurements and the simulated model. Additionally, replicating the real motion in the simulation proved challenging, as the joint states had to be converted into equivalent angles in AVx. While this transformation was mostly successful, some issues affected the simulated motion, leading to the observed differences. However, the nearly identical range values suggest that the simulated radar can accurately capture range measurements. Furthermore, the motion pattern in the simulation appears promising, indicating that with more accurate motion replication, the trajectory could match the real motion.

System Calibration and Real-Time Measurement Precision: The thesis

further underscored the importance of precise calibration and high-resolution measurement data, both in real-world and simulated environments, for obtaining reliable results. The calibration efforts ensured minimal error in real-time measurements, serving as a solid benchmark for simulation comparison.

5.3 Future Work

Enhanced Motion Simulation and Complex Target Scenarios: While the current study tested the radar system's response to a moving target with relatively simple, controlled motions, future work could expand on this by simulating more complex motion trajectories and target configurations. This could involve:

- **Complex Motion Patterns:** Introducing a variety of movement patterns, including faster speeds, sudden stops, and multi-directional changes, could better emulate real-world conditions where interior objects might shift unpredictably.
- **Multiple Targets in Motion:** Incorporating multiple moving targets would add complexity to the scene, better reflecting real-life scenarios such as passenger monitoring in vehicles. This could allow the system to distinguish between different moving targets in close proximity and evaluate its capability to maintain accuracy in more crowded environments.
- **High-Speed Movements:** Testing the radar's responsiveness to high-speed motions would help identify potential lag or loss of resolution in tracking fast objects, especially relevant for applications requiring quick responses, such as detecting sudden passenger movements in vehicles.

Incorporation of Additional Environmental Factors: Expanding the simulation to account for a broader range of environmental factors could provide a more robust validation of radar performance:

- **Multipath and Clutter:** Realistic interior sensing often involves signal reflections from surfaces like windows, seats, or metallic components. Simulating such multipath effects could provide insights into how radar systems cope with signal noise and reflections, enhancing reliability in cluttered environments.
- **Different Material Properties and Textures:** To more closely replicate the real-world variety of objects and surfaces inside a vehicle, future simulations could incorporate materials with varying dielectric properties. This would help test how well radar can differentiate between soft materials (such as upholstery) and harder surfaces (such as seat frames), refining object detection capabilities.

- **Variable Environmental Conditions:** Factors like temperature, humidity, or lighting can indirectly affect radar performance. For example, heating elements in seats or variable lighting conditions might interact with radar reflections. By simulating these environmental variables, future work could gauge the radar's robustness under diverse conditions, particularly for applications requiring high precision in fluctuating interior settings.

Bibliography

- [1] Acconeer Sparse Format, https://docs.acconeer.com/en/latest/a111/radar_data_and_control/services/sparse.html.
- [2] Acconeer Envelope Format, https://docs.acconeer.com/en/latest/a111/radar_data_and_control/services/envelope.html.
- [3] Acconeer System Overview, https://docs.acconeer.com/en/latest/a111/radar_data_and_control/system_overview.html
- [4] Acconeer A111 – Pulsed Coherent Radar (PCR) Datasheet v2, <https://developer.acconeer.com/home/a111-docs-software/>.
- [5] Fawwaz T. Ulaby and David G. Long, *Microwave Radar and Radio Metric Remote Sensing*.
- [6] MIT Lincoln Laboratory, *Introduction to Radar Systems*, <https://www.ll.mit.edu/sites/default/files/outreach/doc/2018-07>.
- [7] J. C. Toomay and Paul J. Hannen, *Radar Principles for the Nonspecialist*, 4th Edition, SciTech Publishing, 2004.
- [8] ABB GoFa CRB 15000, <https://new.abb.com/products/robotics/robots/collaborative-robots/crb-15000>
- [9] ABB, *Product Manual CRB 15000, GoFa*, <https://library.abb.com>.
- [10] Blender Documentation, <https://docs.blender.org/manual/en/latest/advanced/appendices/rotations.html>.
- [11] H. Harry Asada, *Introduction to Robotics, Planar Kinematics, Department of Mechanical Engineering, MIT*, <https://ocw.mit.edu/courses/2-12-introduction-to-robotics-fall-2005>.
- [12] R. Siegwart and I. Nourbakhsh, *Mobile Robot Kinematics*, Carnegie Mellon University and EPFL, <https://www.cs.cmu.edu/~rasc/Download/AMRobots3.pdf>.
- [13] Ashley Stevens, *Forward Kinematics*, Clemson University, <https://opentextbooks.clemson.edu/wangrobotics/chapter/forward-kinematics/>.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY