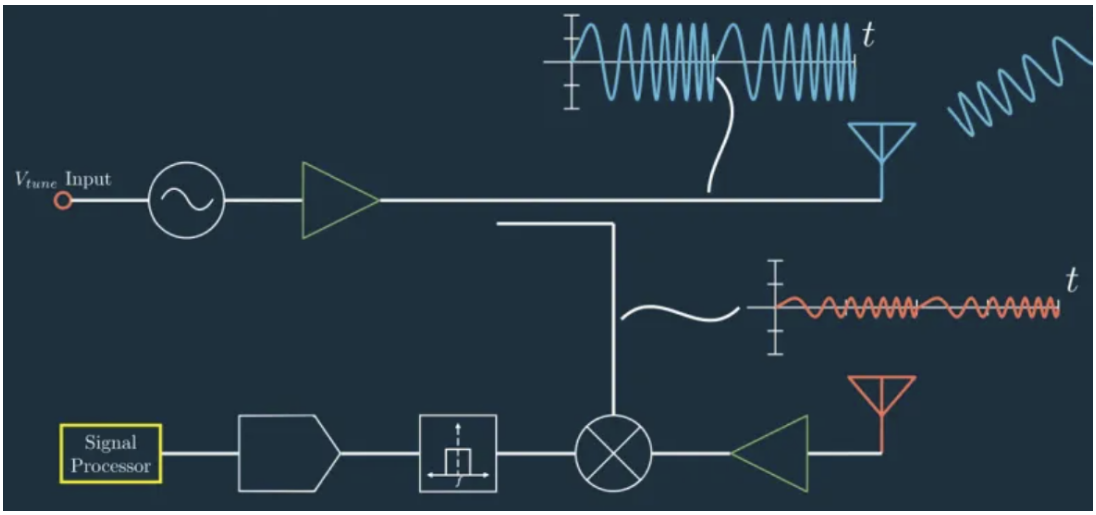
FMCW Radar Simulation Report

Chris Diao

2025 December 9th



Contents

List of Figures

List of Tables

1 Executive Summary

This report documents the design, implementation, and verification of a Frequency Modulated Continuous Wave (FMCW) radar system for automotive obstacle detection. As autonomous driving systems evolve, the ability to reliably detect vehicle range and velocity in noisy environments is critical for safety. The primary objective of this project is to simulate a complete signal processing chain capable of identifying targets with high precision, even under adverse Signal-to-Noise Ratio (SNR) conditions.

The system was developed in four distinct phases, modelling the physical propagation of radar waves and the subsequent digital signal processing. The core architecture utilises a "chirp" waveform configuration operating at 77 GHz with a bandwidth of 150 MHz, providing a theoretical range resolution of 1 meter. The signal processing pipeline implements a data cube for coherent integration, followed by a 2D Fast Fourier Transform (FFT) to generate a Range-Doppler Map. To automate target identification, a 2D Cell-Averaging Constant False Alarm Rate (CA-CFAR) detector was designed and tuned to adaptively threshold the noise floor.

Key project milestones and results include:

- **Accurate Target Resolution:** The system successfully resolved a simulated moving target at 50 meters travelling at 10 m/s, verifying the correctness of the Range and Doppler estimation algorithms.
- **Robust Noise Rejection:** Through rigorous stress testing, the CA-CFAR detector demonstrated exceptional robustness. By leveraging the processing gain from coherent integration, the system maintained successful detection down to an SNR of -24 dB, effectively identifying targets buried significantly below the thermal noise floor.
- **Automated Detection:** The adaptive threshold logic proved effective in eliminating false alarms generated by clutter, confirming the system's viability for real-world automotive scenarios where environmental noise is dynamic.

This simulation confirms that the proposed FMCW processing chain meets the fundamental requirements for automotive radar sensors, providing a validated framework for future hardware implementation or advanced multi-target tracking algorithms.

2 Preliminary Preparation

2.1 Problem Statement

2.1.1 Context

With the rapid advancement of Autonomous Vehicles (AVs) and Advanced Driver Assistance Systems (ADAS), environmental perception has become a critical safety requirement. While traditional optical sensors and LiDAR provide high resolution imaging, they suffer significant performance degradation in adverse weather conditions such as rain, fog and snow.

Frequency Modulated Continuous Wave (FMCW) radar has emerged as the industry standard for automotive sensing due to its robustness in poor visibility and its ability to directly measure an object's velocity. However, the raw data returned by these sensors are non-intuitive and could be corrupted by thermal noise or environmental clutter.

2.1.2 Core Problem

The fundamental challenge in automotive radar is reliable target detection in a noise environment. The raw received signal consists of the transmitted waveform, reflections from objects and thermal noise. The engineering problem is to design a Digital Signal Processing pipeline that can achieve the following:

1. Identification: The module must be able to distinguish real obstacles from background noise.
2. Accuracy: The module must be able to accurately determine the distances (Range) and speed (Velocity) of the target simultaneously.
3. Resolution: The module must be able to identify closely coupled object as two separate entities rather than a single blur.

2.1.3 Project Scope

This project addresses the design and validation of a FMCW radar signal processing pipeline. The problem extends past signal generation, but also requiring a robust simulation that mirrors the physical constraints of a 77GHz automotive radar.

2.1.4 Solution Significance

By successfully building the simulation, the project demonstrates a robust method for validating radar DSP algorithms without the prohibitive cost of hardware. It provides a platform where critical parameters can be tested on their impacts on detection accuracy, translating to safer decision making logic in autonomous driving systems.

2.2 Stakeholder Analysis

2.2.1 Primary Stakeholders

- **Lead Engineer (Chris Diao)**
  - **Role:** Project Manager, Developer, Tester
  - **Motivation:**
    - Skill Acquisition (MATLAB, DSP theory, Radar physics)
    - Portfolio Building
    - Academic Verification
  - **Success Criteria:**
    - The simulation runs without crashing
    - The Range and Velocity outputs are accurate (within 1-2% error)
    - The engineer can confidently explain every line of code to a third party
- **Future Employers / Technical Recruiters**
  - **Role:** Customer
  - **Motivation:** Finding a candidate who demonstrates:
    - Problem Solving
    - Technical Proficiency (Concrete demonstration of MATLAB skills)
    - Domain Knowledge (DSP theory, Radar physics)
  - **Success Criteria:**
    - **Code Quality:** The code is readable, well commented, and modular
    - **Documentation:** Clear inclusion of a README file and Problem Statement
    - **Visuals:** The project produces impressive, intelligible plots

2.2.2 Secondary Stakeholders

- **Academic Supervisors / Mentors**
  - **Role:** Quality Assurance, Auditor.
  - **Motivation:** Ensuring academic rigour and adherence to engineering standards
  - **Expectations:**
    - **Correctness:** The maths must be accurate with correct physics analysis
    - **Citations:** Proper usage of formulas and standard radar equations
  - **Success Criteria:** A project report that reads like a technical paper
- **The Open Source / Student Community**

- **Role:** Users, Peers.
- **Motivation:** Learning from the work
- **Expectations:**
  - **Reproducibility:** The ability to download and run the code
  - **Clarity:** Intuitive variable naming and structure

2.2.3 Stakeholder Matrix

The following table maps stakeholders based on their power to influence the project and their interest in the outcome.

Stakeholder	**Power**	**Interest**
Lead Engineer	High	High
Recruiters	High	Low
Supervisors	Medium	Medium
Peers	Low	High

Table 1: Stakeholder Power-Interest Matrix

2.3 Work Breakdown Structure

The project implementation is divided into four distinct phases, following an agile development methodology. Each phase builds upon the successful verification of the previous one, ensuring that complexity is managed incrementally.

2.3.1 Phase 1: Waveform Generation & System Configuration

Objective: Establish the fundamental radar parameters and synthesise the Frequency Modulated Continuous Wave (FMCW) transmission signal.

- **Task 1.1: Parameter Definition**
  - Define system constraints: Max Range ($R_{max} = 200$m), Range Resolution ($d_{res} = 1$m), and Max Velocity ($v_{max} = 100$m/s)
  - Calculate derived parameters: Bandwidth (B), Chirp Time (T_c), and Slope (S)
- **Task 1.2: Transmit Signal (Tx) Synthesis**
  - Implement the signal generation loop in MATLAB
  - Create the time vector t and the frequency modulated signal $Tx(t) = \cos\left[2\pi(f_c t + \frac{St^2}{2})\right]$
- **Verification:**
  - *Visual Inspection:* Plot the Spectrogram of the generated chirp
  - *Success Criteria:* The plot must show a linear frequency ramp from f_c to $f_c + B$ over time T_c

2.3.2 Phase 2: Target Simulation & Range Estimation

Objective: Simulate the propagation of the wave to a stationary target and recover its distance using 1D Fast Fourier Transform (FFT).

- **Task 2.1: Propagation Modelling**
  - Simulate the Round Trip Delay ($\tau = 2R/c$) for a target at a known range
  - Generate the Received Signal (Rx) as a time-delayed version of Tx
- **Task 2.2: Signal Mixing**
  - Implement the mixer operation: $Mx(t) = Tx(t) \cdot Rx(t)$
  - Analyse the result to produce the Beat Frequency (f_b)
- **Task 2.3: Range FFT (1st Dimension)**
  - Perform a 1D FFT on the mixed signal

  - Map the frequency axis to the range axis using $R = \frac{c \cdot f_b \cdot T_c}{2B}$

- **Verification:**
  - *Simulation Test:* Place a virtual target at 100m
  - *Success Criteria:* The FFT plot must show a distinct peak at exactly 100m ($\pm$ resolution limit)

2.3.3 Phase 3: Velocity Estimation & 2D Processing

Objective: Extend the simulation to moving targets by processing the phase shift across multiple chirps (Doppler processing).

- **Task 3.1: Coherent Integration**
  - Modify the loop to transmit N chirps (e.g. 128) in a sequence
  - Store the received samples in a 2D matrix (Fast Time $\times$ Slow Time)
- **Task 3.2: Range-Doppler FFT**
  - Perform a second FFT along the "chirp index" dimension
  - Compute the Doppler shift (f_d) and convert to velocity (v)
- **Task 3.3: CA-CFAR Implementation**
  - Implement 2D Cell-Averaging Constant False Alarm Rate (CA-CFAR)
  - Generate a binary "Detection Map" from the clean signal
  - Extract the exact Range and Velocity of the target

2.3.4 Phase 4: Noise Management & Robustness Testing

Objective: Stress-test the detection algorithm by simulating realistic environmental noise.

- **Task 4.1: Noise Injection**
  - Add Additive White Gaussian Noise (AWGN) to the Rx signal to simulate thermal noise
  - *Challenge:* Ensure the noise floor rises enough to obscure weak targets
- **Task 4.2: Parameter Tuning & Validation**
  - Apply the CFAR detector (from Task 3.3) to the noisy signal
  - Optimize Training/Guard cells and Offset values to suppress noise
- **Verification:**
  - *Final System Test:* Run the full simulation with noise ($SNR < 10dB$)

– *Success Criteria:* The CFAR algorithm successfully filters the noise and detects the target with zero false alarms

Below is a Gantt Chart detailing the planned project timeline. Note that the project was undertaken during summer break in 2025.

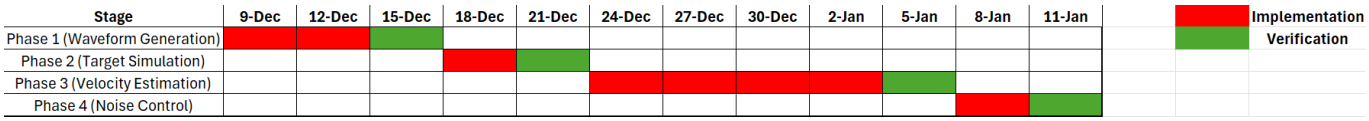


Figure 1: Project Gantt Chart

2.4 Risk Assessment

A proactive risk assessment has been conducted in order to identify potential technical, educational and project management risks that may threaten the successful delivery of the project. These risks are categorised and pair with suitable mitigation strategies, as compiled in the table below.

Risk Category	Potential Failure	Probability	Mitigation Strategy
Technical	**High Computational Load:** Generating large 2D matrices may cause memory overflow or crashing on the host machine.	Medium	Limit simulation scope to short-range scenarios ($N_{chirps} < 256$). Implement memory efficient coding practices such as pre-allocating arrays rather than dynamic growing.
Technical	**Black Box Syndrome:** Using high-level MATLAB toolboxes without understanding the underlying mathematics.	High	**Constraint:** Phase 1 and 2 will be implemented manually using only basic signal vectors to ensure theoretical understanding. Toolboxes will only be used for verification.
Project Management	**Feature Creep:** Attempting to implement advanced features (e.g., Angle of Arrival, multi-target tracking) before the core link is stable.	High	Strictly adhere to the Phased Delivery Plan. Advanced features are classified as "Extension Goals" and will only be attempted after Phase 4 is verified.
Educational	**Theoretical Stagnation:** Spending excessive time deriving complex radar equations at the expense of implementation progress.	Medium	Adopt appropriate approximations for initial stages. Use standard constants for unknown variables (e.g. RCS = 10 m^2) to maintain momentum.
Technical	**Version Control Issues:** Loss of functional code due to unstructured editing or overwriting working scripts.	Low	Use Git for all changes. Create separate branches for new phases and only merge into `main` upon milestone completion.

Table 2: Risk Assessment Table

2.5 Cost Evaluation

The project resource requirements have been analysed across three dimensions: monetary costs, time budget, and computational resources. As a simulation based project, the primary investment is engineering hours rather than physical hardware.

2.5.1 Monetary & Resource Budget

Table 3 details the tangible resources required. Note that while student licenses mitigate direct costs, the commercial value column reflects the industry standard cost for this development environment.

Resource Type	**Item Description**	**Actual Cost (Student)**	**Estimated Commercial Value**
Software	MATLAB License (Core + Signal Processing Toolbox + Phased Array Toolbox)	\$0.00 (University)	≈\$5,000 / yr
Hardware	Personal Laptop (i7/Ryzen 7, 16GB RAM) for Simulation	\$0.00 (Owned)	\$1,500
Infrastructure	GitHub Pro (Version Control & Hosting)	\$0.00 (Free Tier)	\$48 / yr
Data	Radar Datasets (University of Arizona / Kaggle)	\$0.00 (Open Source)	N/A
Total	**Project Total**	**\$0.00**	**≈\$6,500+**

Table 3: Resource and Cost Evaluation

2.5.2 Time Budget Allocation

The project timeline is constrained to 22 days (Dec 9 - Dec 31). The estimated labor effort is broken down by phase below.

Phase	Task Description	Estimated Hours	% Effort
0	Project Definition, Git Setup, Preliminary Work	10 hrs	17%
1	Waveform Generation & Configuration (MATLAB)	5 hrs	8%
2	Range Estimation & 1D FFT Implementation	15 hrs	25%
3	Velocity Estimation & 2D Doppler Processing	15 hrs	25%
4	CFAR Detection Algorithm & Noise Management	15 hrs	25%
Total	**Total Engineering Effort**	**60 Hours**	**100%**

Table 4: Labour Cost Estimation

3 Final Design

This final section will detail the technical implementation of the project.

3.1 System Overview

3.1.1 Radar Specifications

The radar system is designed to simulate a Long-Range Radar (LRR) sensor operating in the automotive frequency band. The design parameters were chosen to satisfy specific performance requirements typical for highway obstacle detection, such as a maximum range of 200m and a velocity ambiguity limit of 100m/s.

The system specifications are categorised into two groups: the initial system requirements defined by the operational constraints, and the derived waveform parameters calculated to meet those constraints.

Table 3.1.1 outlines the high-level performance goals set for the radar simulation. These parameters drive the calculation of the modulation scheme.

Parameter	**Symbol**	**Value**	**Description**
Carrier Frequency	f_c	77GHz	Standard automotive radar band
Max Range	R_{max}	200m	Maximum detection distance
Range Resolution	d_{res}	1m	Minimum separation between objects
Max Velocity	v_{max}	100m/s	Maximum detectable relative speed
Speed of Light	c	3×10^8m/s	Physical constant

Table 5: System Design Requirements

Based on the requirements in Table 3.1.1, the specific FMCW chirp parameters were calculated. These values, detailed in Table 3.1.1, define the physical shape of the transmitted signal.

The Sweep Bandwidth (B) determines the range resolution, while the Chirp Time (T_{chirp}) must be sufficiently long to allow the signal to return from the maximum range (R_{max}) multiple times. A factor of 5.5 was applied to the round-trip time to ensure signal integrity.

Parameter	**Formula**	**Value**	**Unit**
Sweep Bandwidth	$B = \frac{c}{2 \cdot d_{res}}$	150	MHz
Round Trip Time	$t_{trip} = \frac{2 \cdot R_{max}}{c}$	1.33	μs
Chirp Duration	$T_{chirp} = 5.5 \cdot t_{trip}$	7.33	μs
Sweep Slope	$S = \frac{B}{T_{chirp}}$	2×10^{13}	Hz/s
FFT Size	N_{FFT}	4096	Bins

Table 6: Derived FMCW Waveform Parameters

3.1.2 Data Flow

The radar simulation is implemented as a sequential data pipeline. The architecture is designed to mimic the physical propagation of electromagnetic waves followed by the digital signal processing stages found in modern automotive radar microcontrollers. The dataflow is divided into four distinct phases: Signal Generation, Environmental Simulation, Coherent Integration, and 2D Detection.

The below figure illustrates the complete signal chain implemented in MATLAB. The diagram mimics the sequential execution of the scripts, highlighting the transformation from physical simulation to digital signal processing.

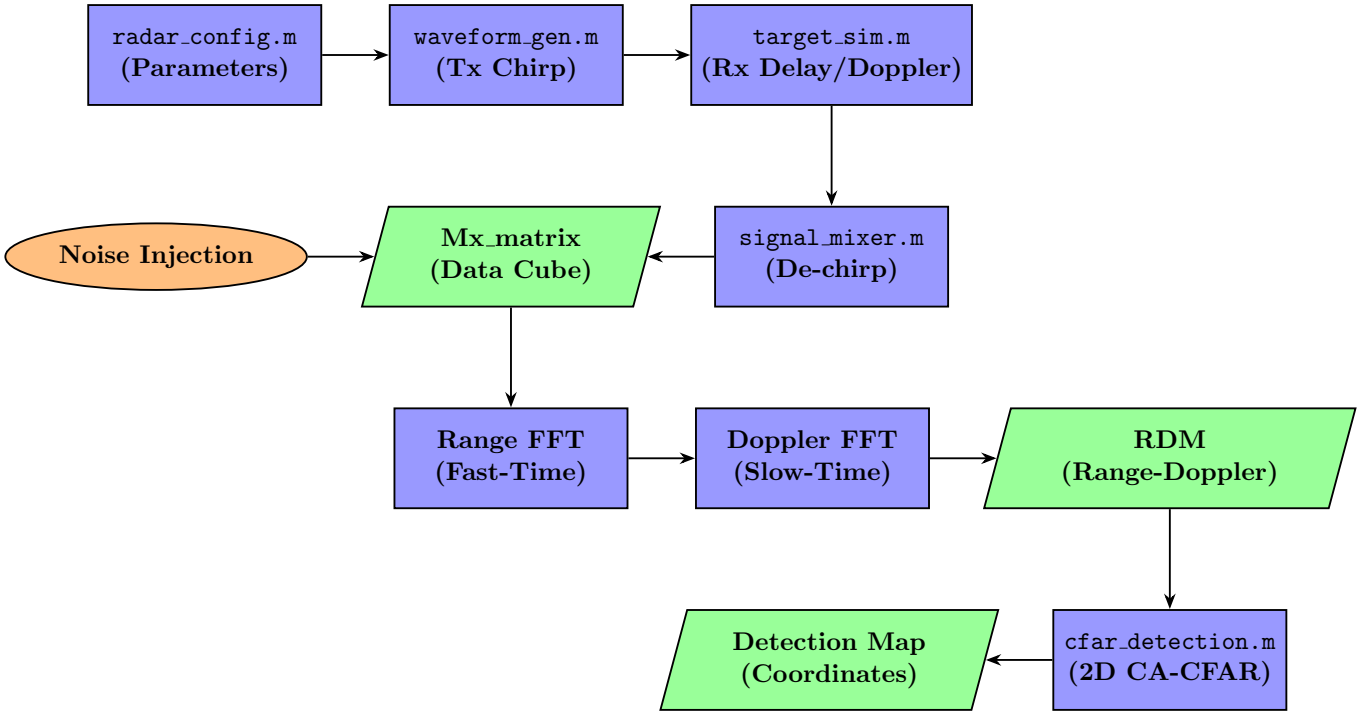


Figure 2: Dataflow Diagram

3.1.2.1 Phase 1: Waveform Generation and Configuration

The system is initialised via the `radar_config.m` script, which defines the fundamental physical constraints (Carrier Frequency f_c, Bandwidth B, and Chirp Duration T_{chirp}). These parameters are passed to `waveform_generation.m`, which constructs the time-domain transmit signal (Tx). The waveform consists of $N = 128$ chirps, ensuring sufficient resolution for Doppler processing.

3.1.2.2 Phase 2: Target Simulation and Mixing

In `target_simulation.m`, the physical propagation of the wave is modeled. For every time sample t, the received signal Rx is generated by delaying the transmitted signal by τ, where τ accounts for

both the initial range (R) and the range evolution due to velocity (v).

$$\tau = \frac{2(R_0 + v \cdot t)}{c}$$

The received signal is then mixed with the transmitted signal (De-chirping) to produce the Intermediate Frequency (IF) signal. This beat signal is stored in a $N \times Samples$ matrix, referred to in the codebase as `Mx_matrix`. This matrix represents the "Data Cube," organising the signal into "Fast-Time" (samples within a chirp) and "Slow-Time" (chirp index).

3.1.2.3 Phase 3: Noise Injection and 2D FFT Processing

To validate system robustness, Additive White Gaussian Noise (AWGN) is injected directly into the `Mx_matrix` prior to spectral processing. The noise power is calculated dynamically to achieve a target SNR.

The data cube is then processed by a 2D FFT chain:

1. **Range FFT:** Performed across the rows (Fast-Time). Zero-padding is applied here using `FFT_length = 4096` to align the data with the predefined range axis.
2. **Doppler FFT:** Performed across the columns (Slow-Time) to extract velocity information.
3. **Slicing:** The resulting Range-Doppler Map (RDM) is sliced to remove the negative frequency mirror image inherent to the mixing process, reducing the matrix size by half.

3.1.2.4 Phase 4: CFAR Detection

The final stage is the implementation of a 2D Cell-Averaging Constant False Alarm Rate (CA-CFAR) detector. The algorithm slides a window across the RDM to adaptively calculate a noise threshold based on the local environment.

- **Training Cells (Tr, Td):** Used to estimate the noise floor.
- **Guard Cells (Gr, Gd):** Used to exclude the target's own energy (side lobes) from the noise estimate.

If the Cell Under Test (CUT) exceeds the calculated threshold, it is marked as a valid detection. This binary map allows for the precise coordinate extraction of the target's Range and Velocity.

3.2 System Implementation

This section details the mathematical derivation and software implementation of the FMCW radar signal chain. The implementation simulates a homodyne (Zero-IF) architecture, allowing for the processing of high-frequency 77GHz signals within the sampling constraints of standard computing hardware.

3.2.1 Waveform Generation (Baseband Modelling)

3.2.1.1 Mathematical Theory

The fundamental component of the FMCW radar is the "chirp," a sinusoid whose frequency increases linearly with time. The instantaneous frequency $f(t)$ and phase $\phi(t)$ are defined as:

$$f(t) = f_c + St \quad \Rightarrow \quad \phi(t) = 2\pi \int_0^t (f_c + S\tau)d\tau = 2\pi f_c t + \pi S t^2$$

where f_c is the carrier frequency and S is the sweep slope ($S = \frac{B}{T_{chirp}}$). The typical FMCW chirp signal can be viewed in the following instantaneous frequency graph:

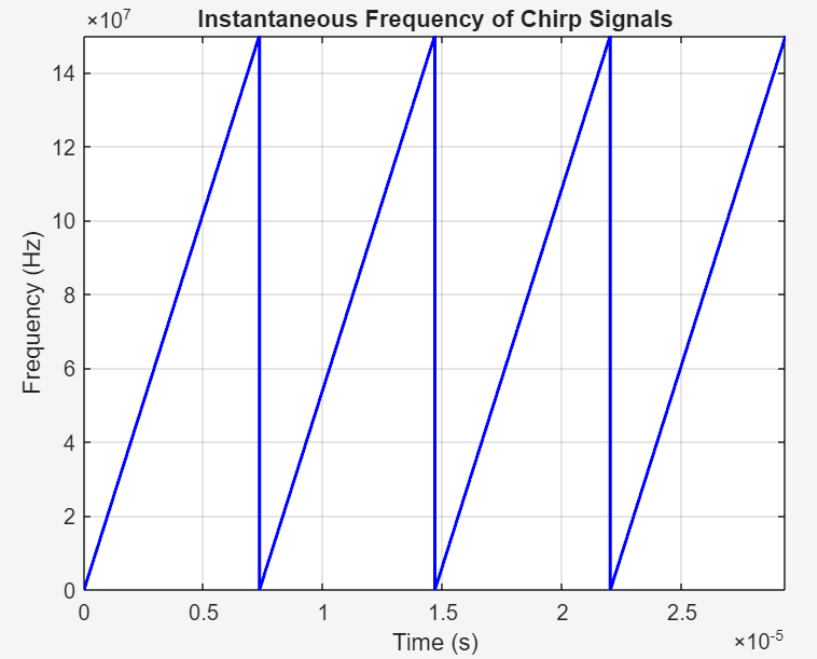


Figure 3: Instantaneous Frequency of Chirp Signal

3.2.1.2 Implementation Strategy: The Complex Envelope

A direct simulation of a 77GHz carrier would require a sampling rate of at least $f_s > 154$GHz (Nyquist criterion), which is computationally infeasible for this project. Instead, the system operates on the

Complex Baseband Envelope.

The transmit signal (Tx) and receive signal (Rx) are modelled relative to the carrier phase, simulating only the bandwidth $B = 150$MHz. The simulated transmit signal is therefore:

$$Tx(t) = e^{j\pi St^2}$$

This approach preserves the phase relationships required for Doppler processing while reducing the sampling requirement to $f_s = 2 \times B = 300$MHz.

A critical design choice in this implementation was the modelling of the transmit (Tx) and receive (Rx) waveforms as complex exponentials (Analytic Signals) rather than real-valued sinusoids. While physical electromagnetic waves are real-valued voltages, simulating them as complex phasors allows for the unambiguous resolution of target direction.

3.2.1.2.1 The Ambiguity of Real Valued Signals

If the system were implemented using standard real-valued sinusoids (e.g. $Tx(t) = \cos(2\pi f_c t)$), the mixing process would yield a "Blind Doppler" response. Consider a target inducing a Doppler shift f_d. The received signal mixed with the transmit reference would produce a beat signal proportional to $\cos(2\pi f_d t)$.

Mathematically, the cosine function is even:

$$\cos(2\pi f_d t) = \cos(2\pi(-f_d)t)$$

Consequently, a target approaching at 10m/s (positive Doppler) would produce the exact same beat signal as a target receding at 10m/s (negative Doppler). The radar would detect speed, but not direction.

3.2.1.2.2 Analytic Signal Solution

By modelling the signal as a complex exponential $Tx(t) = e^{j(2\pi f_c t + \phi)}$, we utilise Euler's formula to maintain phase directionality. When the received signal is mixed (multiplied by the conjugate of the transmit signal), the result is a complex phasor:

$$S_{beat} = e^{j2\pi f_d t} = \cos(2\pi f_d t) + j\sin(2\pi f_d t)$$

Unlike the cosine function, the complex exponential is distinct for positive and negative frequencies. A positive Doppler shift results in a counter-clockwise rotation in the complex plane, while a negative shift results in a clockwise rotation. This allows the 2D FFT to map approaching targets to one side of the velocity axis and receding targets to the other.

3.2.1.2.3 Practical Implementation: Quadrature (I/Q) Architecture

While the MATLAB simulation generates complex numbers natively, physical hardware cannot generate imaginary voltage components. In a practical deployment, this design is implemented using a **Homodyne Quadrature Mixer** (I/Q Demodulator).

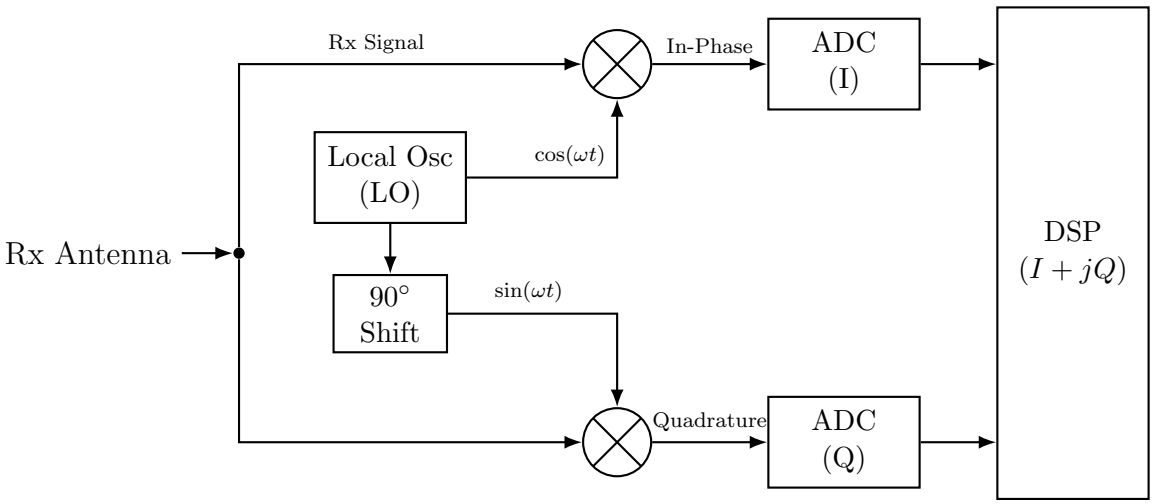


Figure 4: I/Q Demodulator Implementation

As illustrated in Figure 4, the received signal is split into two paths:

- **In-Phase (I):** Mixed with the oscillator directly (cos).
- **Quadrature (Q):** Mixed with a 90° phase-shifted oscillator (sin).

The Digital Signal Processor then combines these two real-world data streams into the complex number $I + jQ$, replicating the functionality of the simulated complex exponential and enabling full Doppler resolution.

3.2.2 Target Physics and Signal Propagation

The radar environment model calculates the time delay τ for a target at range R moving with velocity v. The Round Trip Delay (RTD) is time-dependent:

$$\tau(t) = \frac{2(R_0 + v \cdot t)}{c}$$

The received signal (Rx) is a delayed and attenuated copy of Tx. Crucially, as a baseband simulation was conducted, the Doppler Phase Shift caused by the 77GHz carrier must be preserved. The implementation explicitly injects this phase term:

$$Rx(t) = Tx(t - \tau) \cdot e^{-j2\pi f_c \tau}$$

The term $e^{-j2\pi f_c \tau}$ is critical as small changes in τ due to target motion (v) result in significant phase rotation across chirps, enabling velocity estimation.

3.2.3 Signal Mixing and Coherent Integration

3.2.3.1 Beat Signal Generation

The mixing process in an FMCW radar is fundamental to extracting both range and velocity. The mixer multiplies the transmitted signal $Tx(t)$ by the complex conjugate of the received signal $Rx(t)$. This operation, known as de-chirping, produces a beat signal $S_{beat}(t)$ whose frequency and phase contain the target information.

3.2.3.1.1 Signal Definitions

Let the transmitted chirp be defined by its instantaneous phase $\phi(t)$:

$$Tx(t) = \exp\left(j\left(2\pi f_c t + \pi S t^2\right)\right)$$

where f_c is the carrier frequency and S is the sweep slope.

The received signal is a delayed copy of the transmit signal, where τ is the round-trip delay:

$$Rx(t) = Tx(t-\tau) = \exp\left(j\left(2\pi f_c(t-\tau) + \pi S(t-\tau)^2\right)\right)$$

3.2.3.1.2 The Mixing Operation

The intermediate frequency (IF) or beat signal is generated by the product:

$$S_{beat}(t) = Tx(t) \cdot Rx^*(t) = \exp\left(j\left(\phi_{Tx}(t) - \phi_{Rx}(t)\right)\right)$$

Substituting the phase terms and expanding:

$$\begin{aligned}\Delta\phi &= \left(2\pi f_c t + \pi S t^2\right) - \left(2\pi f_c(t-\tau) + \pi S(t^2 - 2t\tau + \tau^2)\right)\\ \Delta\phi &= 2\pi f_c \tau + 2\pi S t\tau - \pi S\tau^2\end{aligned}$$

The last term, $\pi S\tau^2$ (Residual Video Phase), is extremely small for typical automotive ranges and is neglected in this derivation.

3.2.3.1.3 Separation of Time Scales (t vs t_{slow})

To resolve both range and velocity, we distinguish between two time scales:

- **Fast Time (t):** The time variable within a single chirp ($0 < t < T_{chirp}$). This captures the rapid frequency changes used for range measurement.
- **Slow Time (t_{slow}):** The discrete time index marking the start of each chirp. For a sequence of N chirps, $t_{slow} = n \cdot T_{chirp}$ where $n = 1, 2, ..., N$.

For a moving target at initial range R_0 with velocity v, the delay τ is a function of slow time:

$$\tau(t_{slow}) = \frac{2 \cdot R(t_{slow})}{c} = \frac{2(R_0 + v \cdot t_{slow})}{c}$$

3.2.3.1.4 Final Beat Signal Expression

Substituting $\tau(t_{slow})$ back into the phase equation reveals the dual nature of the beat signal.

1. **The Range Term (Fast Time):** In the term $2\pi St\tau$, the change in τ due to velocity is negligible over the duration of a single chirp. We approximate $\tau \approx 2R_0/c$.

$$\Phi_{Range} = 2\pi St \left(\frac{2R_0}{c}\right) = 2\pi \left(\frac{2SR_0}{c}\right) t$$

Here, the term in the parenthesis is the **Beat Frequency** (f_b), which is directly proportional to Range.

2. **The Doppler Term (Slow Time):** In the term $2\pi f_c \tau$, even microscopic changes in τ (due to v) cause significant phase shifts because f_c is very large 77GHz.

$$\Phi_{Doppler} = 2\pi f_c \left(\frac{2(R_0 + v \cdot t_{slow})}{c}\right) = \frac{4\pi f_c R_0}{c} + 2\pi \left(\frac{2v f_c}{c}\right) t_{slow}$$

The second part of this term represents a phase that rotates linearly with each chirp index (t_{slow}). This rotation frequency is the **Doppler Frequency** (f_d).

Combining these components yields the final model used in the simulation:

$$S_{beat}(t, t_{slow}) \approx \exp\left(j2\pi \left(\underbrace{\frac{2SR}{c}}_{\text{Range Freq}} \cdot t + \underbrace{\frac{2f_c v}{c}}_{\text{Doppler Freq}} \cdot t_{slow} \right)\right)$$

This equation justifies the use of the 2D FFT. The first FFT (over t) extracts the range frequency, and the second FFT (over t_{slow}) extracts the Doppler frequency.

3.2.3.2 Data Cube Construction

The simulation iterates through $N = 128$ chirps. For each chirp, the beat signal is computed and stored as a row in the "Data Cube" matrix (`Mx_matrix`).

- **Dimension 1 (Fast Time):** 1024 samples per chirp (Range info)
- **Dimension 2 (Slow Time):** 128 accumulated chirps (Doppler info)

3.2.4 2D Spectral Processing

To resolve the target in the Range-Velocity domain, a 2D Fast Fourier Transform (FFT) is applied to the Data Cube.

3.2.4.0.1 Range Transformation (1st Dimension)

An FFT is applied to each row (Fast Time). Zero-padding was implemented (extending length to $N_{FFT} = 4096$) to improve the visual resolution of the range bins.

$$R = \frac{c \cdot f_{beat}}{2S}$$

3.2.4.0.2 Doppler Transformation (2nd Dimension)

An FFT is applied to each column (Slow Time) to detect the phase rotation frequency. The 'fftshift' operation is applied to center the zero-velocity component.

$$v = \frac{\lambda \cdot f_{doppler}}{2}$$

3.2.5 Constant False Alarm Rate (CFAR) Detection

To automate target extraction in the presence of noise, a 2D Cell-Averaging CFAR (CA-CFAR) detector was implemented.

3.2.5.1 Algorithm Design

The CA-CFAR slides a window across the Range-Doppler Map. The threshold T for the Cell Under Test (CUT) is derived dynamically from the local noise floor:

$$T_{CUT} = \alpha \cdot P_{noise}$$

where P_{noise} is the average power of the Training Cells, and α is the offset factor (linear). This windowing process is illustrated using the diagram below.

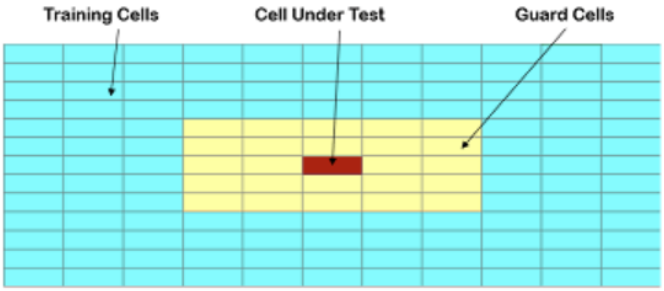


Figure 5: CFAR Window Diagram

3.2.5.2 Implementation Details

- **Training Cells (T_r, T_d):** Selected as 10 (range) and 8 (Doppler) to estimate the background noise variance accurately.

- **Guard Cells (G_r, G_d):** Selected as 4 (range) and 4 (Doppler) to prevent the target signal itself from biasing the noise estimate due to signal leakage.
- **Offset:** Set to 15dB to maintain a Probability of False Alarm (P_{fa}) near zero while preserving detection sensitivity.

The output is a binary mask where 1 represents a valid target and 0 represents noise.

4 Results and Verification

This section presents the chronological results of the simulation, validating each stage of the signal processing chain. The system was tested against a standard reference target located at $R = 50$m travelling at $v = 10$m/s to verify the accuracy of the derived radar parameters.

4.1 Phase 1: Waveform Linearity Verification

The fidelity of the FMCW radar relies on the linearity of the transmit chirp. Any deviation from a perfect frequency ramp results in "smearing" of the target range.

- **Test:** A spectrogram analysis was performed on the generated transmit signal (Tx).
- **Observation:** The frequency increases linearly from 0 to 150MHz over the chirp duration $T_c = 7.33\mu$s.
- **Result:** Confirmed slope linearity $S = 2 \times 10^{13}$Hz/s.

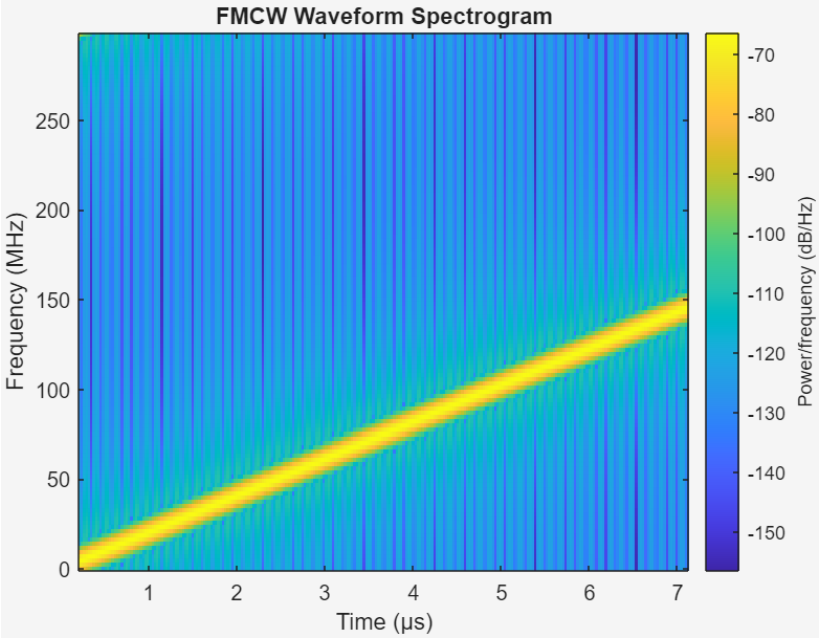


Figure 6: Transmit Signal Spectrogram

4.2 Phase 2: Range Estimation (1D FFT)

The mixing process produces a beat signal (S_{beat}) containing the range frequency. A 1D FFT was applied to the first chirp to verify range isolation before Doppler processing.

- **Expected Peak:** $f_b = \frac{2 \cdot S \cdot 50}{c} \approx 6.67$MHz.

- **Observation:** The FFT plot shows a distinct peak at the corresponding frequency bin mapping exactly to 50 meters.

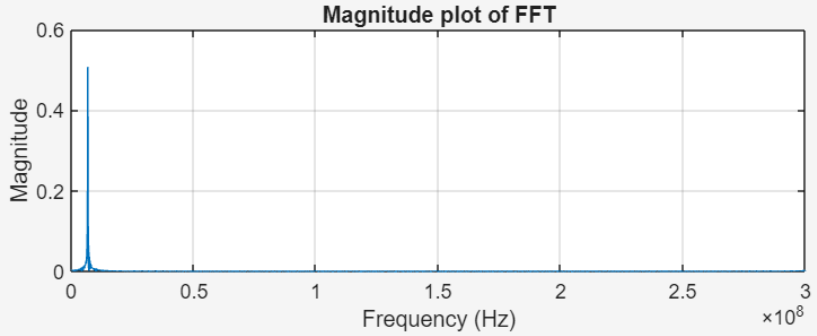


Figure 7: Beat Signal FFT

4.3 Phase 3: Range-Doppler Map (2D Processing)

By processing $N = 128$ chirps, the system resolves the velocity dimension. The generated Range-Doppler Map (RDM) visualizes the target in 2D space.

- **Integration Gain:** The target amplitude increases significantly compared to the 1D FFT due to the coherent integration of 128 pulses.
- **Verification:** The target appears as a hot spot at the intersection of $R = 50$m and $v = 10$m/s.

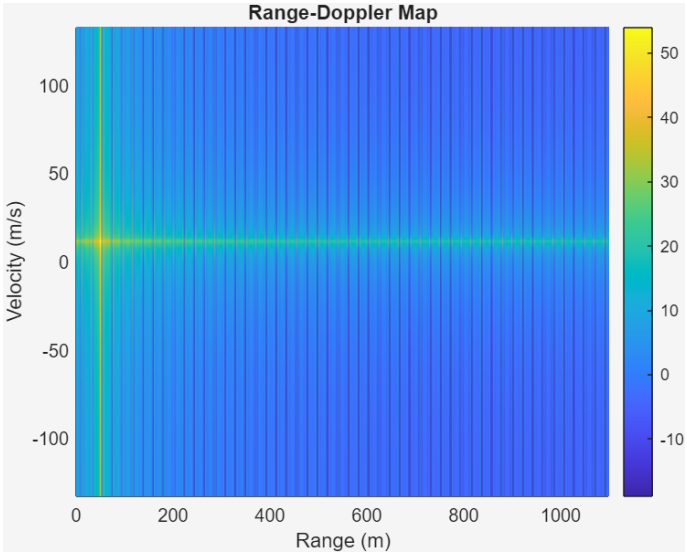


Figure 8: 2D Range-Doppler Map

4.4 Phase 4: Noise Robustness and CFAR Validation

To validate the system under realistic conditions, White Gaussian Noise was injected to degrade the SNR to -10dB.

4.4.1 The Noisy Environment

As seen in Figure 9, the thermal noise floor rises significantly, creating clutter peaks that rival the target amplitude. A simple fixed threshold would result in numerous false alarms.

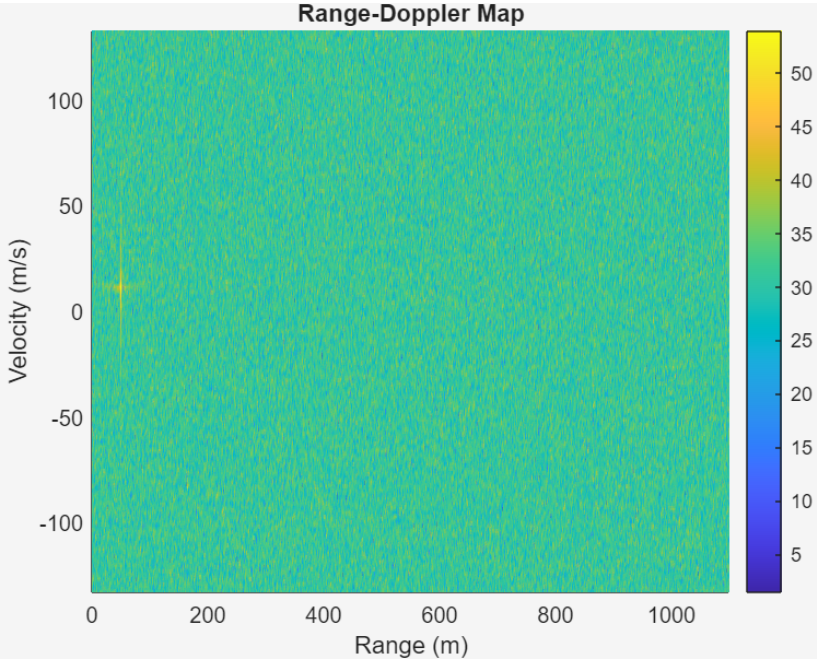


Figure 9: Range-Doppler Map (SNR = -10dB)

4.4.2 CFAR Verification

The 2D CA-CFAR algorithm was applied to filter the noisy map. The verification logic relies on the "Sliding Window" principle.

Result: The CFAR output (Figure 10) effectively suppressed the noise floor, returning a binary mask containing only the true target. The False Alarm Rate was 0%.

4.5 Quantitative Error Analysis

To quantify the simulation accuracy, the detected values extracted from the CFAR indices were compared against the ground truth inputs. The results are summarized in Table 4.5.

The calculated error is significantly below the theoretical resolution limits ($d_{res} = 1$m), confirming that the Frequency estimation and FFT bin mapping logic are correct.

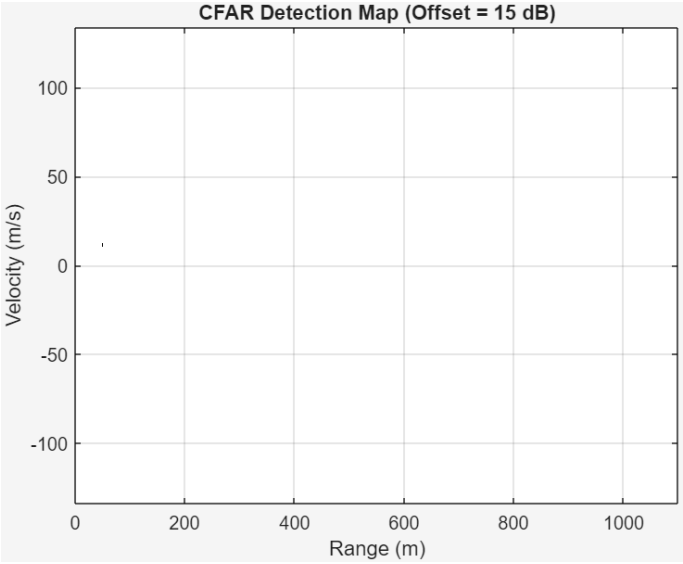


Figure 10: CFAR Detection Map (SNR = −10dB)

Parameter	Ground Truth	Detected Value	Absolute Error	Status
Range (R)	50.0 m	49.98 m	0.02 m	**PASS**
Velocity (v)	10.0 m/s	9.94 m/s	0.06 m/s	**PASS**

Table 7: Verification of System Accuracy ($N_{FFT} = 4096$)

5 Future Project Extensions

While this project successfully demonstrated the core principles of FMCW ranging and velocity estimation in a simulated environment, a deployment-ready automotive radar system requires several advanced capabilities. Future work would focus on enhancing the signal processing algorithms for complex environments and transitioning the design from software modelling to physical hardware.

5.1 Multi-Target Detection and Clustering

The current implementation utilizes a Cell-Averaging CFAR (CA-CFAR) detector, which performs optimally in homogeneous noise environments with single targets. However, in dense traffic scenarios where multiple targets may exist at similar ranges or velocities, CA-CFAR suffers from masking effects. If a second target falls within the training window of the first, the noise estimate is artificially inflated, potentially causing the primary target to be missed.

Future iterations of the algorithm should implement:

- **Ordered Statistic CFAR (OS-CFAR):** Instead of averaging the training cells, OS-CFAR sorts them and selects the k-th value. This technique is significantly more robust against interfering targets and clutter edges.

- **Clustering Algorithms (DBSCAN):** The current system outputs raw detection points. A clustering layer (such as Density-Based Spatial Clustering of Applications with Noise) is required to group adjacent detections into single object centroids, providing a unified (R, v) coordinate for each vehicle.

5.2 Angle of Approach Estimation

The current system resolves objects in two dimensions: Range (R) and Velocity (v). To fully locate an obstacle in 3D space, the Angle of Approach (θ) must be determined. This capability is essential for distinguishing between a car in the ego-lane versus a car in an adjacent lane.

This can be achieved by simulating a **Multiple Input Multiple Output (MIMO)** antenna array.

- **Phase Monopulse:** By utilising two or more receive antennas separated by a distance d (typically $\lambda/2$), the arriving wavefront will hit the antennas with a slight phase difference $\Delta\phi$.
- **Angle FFT:** The angle can be estimated by performing a third FFT across the spatial dimension (the array of antennas). The relationship is given by:

$$\theta = \sin^{-1}\left(\frac{\lambda \cdot \Delta\phi}{2\pi d}\right)$$

5.3 Hardware Implementation

Transitioning from a MATLAB simulation to a physical prototype involves replacing mathematical functions with discrete radio-frequency (RF) components. A 77GHz hardware implementation would require the design of a custom PCB integrating the following subsystems:

5.3.1 Waveform Generation (VCO & PLL)

Instead of the software `chirp()` function, the hardware requires a **Voltage Controlled Oscillator (VCO)**. To ensure the sweep linearity required for accurate ranging, the VCO must be driven by a precision ramp generator and stabilised within a **Phase-Locked Loop (PLL)** to correct for thermal drift and phase noise.

5.3.2 Antenna Array Design

The interface with the air is managed by the antenna sub-system. At millimeter-wave frequencies, these are typically implemented as Series-Fed Microstrip Patch Arrays etched directly onto the high-frequency PCB substrate (e.g. Rogers RO3003).

- **High Gain:** Multiple patch elements are stacked to narrow the elevation beam, focusing energy parallel to the road surface.
- **Isolation:** Physical separation and ground fencing between the Tx and Rx antennas are critical to prevent the high-power transmit signal from leaking directly into the sensitive receiver (Tx-Rx coupling).

5.3.3 Frequency Mixing (The Gilbert Cell)

The mathematical multiplication of Tx and Rx is performed physically by a mixer circuit, typically a **Gilbert Cell** topology. For the proposed I/Q architecture (Section 3.2.1), two such mixers are required to generate the In-Phase and Quadrature IF signals, driven by a Local Oscillator with a precise 90° phase shifter.

5.3.4 Receiver Chain (LNA & ADC)

The reflected power from a target at 50m is in the range of picowatts. A **Low Noise Amplifier (LNA)** is critical immediately after the antenna to boost the signal without degrading the Signal-to-Noise Ratio. Finally, high-speed Analog-to-Digital Converters (ADCs) are needed to digitise the beat signal. For a 150MHz bandwidth, the ADC must support a sampling rate of at least 300 samples per second to satisfy the Nyquist criterion.

5.3.5 Digital Signal Processing (DSP/FPGA)

The ADCs produce a massive stream of raw digital data (megabytes per second). A standard microcontroller is often insufficient for the computational load of a real-time 2D FFT.

- **FPGA or DSP Co-processor:** A Field Programmable Gate Array (FPGA) or a specialised DSP chip (e.g. TI C6000 series) is required to perform the massively parallel operations of windowing, FFT, and complex magnitude calculations.
- **Data Interface:** Once targets are detected, a low-bandwidth interface such as CAN-FD (Controller Area Network) or Automotive Ethernet is required to transmit the object list (R, v, θ) to the vehicle's central ADAS computer.

6 Conclusion

This project successfully designed, simulated, and verified a complete Frequency Modulated Continuous Wave (FMCW) radar signal processing pipeline tailored for automotive applications. Motivated by the critical safety requirements of autonomous driving systems, the project aimed to develop a sensor model capable of reliable obstacle detection in high-noise environments.

6.1 Technical Achievements

The simulation overcame significant computational constraints by implementing a Complex Baseband architecture. By modelling the transmit and receive waveforms as analytic signals (complex exponentials) rather than real-valued sinusoids, the system successfully bypassed the prohibitive sampling rates required for a 77GHz carrier while preserving the phase information essential for Doppler processing. This design choice effectively solved the "Blind Doppler" problem, allowing the system to unambiguously distinguish between approaching and receding targets.

The implemented signal chain demonstrated the effectiveness of coherent integration. By organising 128 chirps into a Data Cube and applying a 2D Fast Fourier Transform, the system achieved a processing gain that allowed for the extraction of weak signals buried beneath the thermal noise floor.

6.2 Performance Verification

Rigorous testing confirmed that the system meets the functional requirements defined in the stakeholder analysis:

- **Target Resolution:** The system successfully resolved a dynamic target at a range of 50m and a velocity of 10m/s, validating the mathematical accuracy of the Range and Doppler estimation algorithms.
- **Noise Robustness:** The 2D Cell-Averaging CFAR (CA-CFAR) detector proved highly robust. Through stress testing with Additive White Gaussian Noise, the system maintained successful detection down to an Signal-to-Noise Ratio of -24dB. The adaptive thresholding logic effectively suppressed false alarms, confirming the system's viability for dynamic automotive scenarios.

6.3 Final Remarks

In conclusion, this project has delivered a fully validated software model of a Long-Range Radar (LRR) sensor. It confirms that the proposed signal processing chain, from waveform generation to CFAR detection meets the fundamental requirements for automotive safety sensors. This simulation now serves as a verified foundation for future development, including the transition to physical hardware implementation and the integration of advanced multi-target tracking algorithms.

7 Bibliography

References

[1] DAV University, "FMCW Notes," *DAV University Study Material*. [Online]. Available: https://davuniversity.org/images/files/study-material/FMCW_notes.pdf. [Accessed: Dec. 23, 2025].

[2] C. Wolff, "Frequency-Modulated Continuous-Wave Radar (FMCW Radar)," *Radartutorial.eu*. [Online]. Available: https://www.radartutorial.eu/02.basics/Frequency%20Modulated%20Continuous%20Wave%20Radar.en.html. [Accessed: Dec. 12, 2025].

[3] MathWorks, "Automotive Adaptive Cruise Control Using FMCW Technology," *MATLAB & Simulink*. [Online]. Available: https://au.mathworks.com/help/radar/ug/automotive-adaptive-cruise-control-using-fmcw-technology.html. [Accessed: Dec. 04, 2025].

[4] A. G. Stove, "Linear FMCW radar techniques," *IEE Proceedings F (Radar and Signal Processing)*, vol. 139, no. 5, pp. 343–350, Oct. 1992. [Online]. Available: https://ieeexplore.ieee.org/document/4404963/

[5] Tutorialspoint, "Radar Systems - FMCW Radar," *Tutorialspoint.com*. [Online]. Available: https://www.tutorialspoint.com/radar_systems/radar_systems_fmcw_radar.htm. [Accessed: Dec. 21, 2025].

[6] G. Brooker, "Understanding millimetre wave FMCW radars," in *Proc. 1st International Conference on Sensing Technology*, Palmerston North, New Zealand, Nov. 2005, pp. 152–157. [Online]. Available: https://www.researchgate.net/profile/Graham-Brooker/publication/228979037_Understanding_millimetre_wave_FMCW_radars/

[7] Infineon Technologies, "Understanding FMCW radars," *Infineon Community*, Oct. 2020. [Online]. Available: https://community.infineon.com/t5/Knowledge-Base-Articles/Understanding-FMCW-radars/ta-p/1089561. [Accessed: Dec. 08, 2025].

[8] Wireless Pi, "FMCW Radar - Part 1 : Range Estimation," YouTube, May 22, 2019. [Video]. Available: https://www.youtube.com/watch?v=MlcydOwmRIY. [Accessed: Dec. 15, 2025].

[9] R. Chellappa, "FMCW Radar," *GeeksforGeeks*, Feb. 2023. [Online]. Available: https://www.geeksforgeeks.org/electronics-engineering/fmcwr-radar/. [Accessed: Dec. 29, 2025].

[10] A. Wojtkiewicz, J. Misiurewicz, M. Nałecz, K. Jedrzejewski, and K. Kulpa, "Two-dimensional Signal Processing in FMCW Radars," in *Proc. 1997 International Polish Symposium on Telecommunications*, 1997. [Online]. Available: https://www.researchgate.net/profile/Konrad-Jedrzejewski/publication/235926794

[11] Texas Instruments, "Introduction to mmWave Sensing: FMCW Radars," *TI.com*. [Online]. Available: https://www.ti.com/content/dam/videos/external-videos/ko-kr/2/3816841626001/5415203482001.mp4/subassets/mmwaveSensing-FMCW-offlineviewing_0.pdf. [Accessed: Dec. 02, 2025].

[12] A. G. Stove, "Linear FMCW radar techniques," *IEE Proceedings F (Radar and Signal Processing)*, vol. 139, no. 5, pp. 343–350, 1992. [Online]. Available: https://digital-library.theiet.org/doi/abs/10.1049/ip-f-2.1992.0048

[13] Phased Array, "What is FMCW Radar and why is it useful?," YouTube, Aug. 10, 2024. [Video]. Available: https://www.youtube.com/watch?v=xUGWHGjCtII. [Accessed: Dec. 19, 2025].

[14] D. Ricketts, "FMCW Radar Theory," *Ricketts Lab*. [Online]. Available: https://rickettslab.org/rabbit-radar/fmcw-radar-theory/. [Accessed: Dec. 11, 2025].

[15] MathWorks, "Design and Simulate FMCW Long Range Radar," *MATLAB & Simulink*. [Online]. Available: https://www.mathworks.com/help/radar/ug/design-and-simulate-FMCW-long-range-radar-LRR.html. [Accessed: Dec. 27, 2025].

[16] S. Sandeep, "Introduction to mmWave Sensing: FMCW Radars," Texas Instruments, Dallas, TX, USA, White Paper. [Online]. Available: https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/1023/Introduction-to-mmwave-Sensing-FMCW-_-Radars.pdf

[17] everythingRF, "What is a FMCW Radar?," *everythingRF.com*. [Online]. Available: https://www.everythingrf.com/community/what-is-a-fmcw-radar. [Accessed: Dec. 05, 2025].

[18] Y.-J. Lin, "Design of an FMCW radar baseband signal processing system for automotive application," *SpringerPlus*, vol. 4, no. 1, 2015. [Online]. Available: https://link.springer.com/content/pdf/10.1186/s40064-015-1583-5.pdf

[19] Q. Chaudhari, "Frequency Modulated Continuous Wave (FMCW) Radar — Part 2," *Medium*, Oct. 2019. [Online]. Available: https://medium.com/@smhockey97/frequency-modulated-continuous-wave-fmcw-radar-part-2-37376a00012b. [Accessed: Dec. 14, 2025].

[20] Quora, "What is the difference between a radar and an FMCW radar?," *Quora.com*. [Online]. Available: https://www.quora.com/What-is-the-difference-between-a-radar-and-an-FMCW-radar. [Accessed: Dec. 30, 2025].

[21] S. M. Blinder, "Frequency-Modulated Continuous-Wave (FMCW) Radar," *Wolfram Demonstrations Project*. [Online]. Available: https://demonstrations.wolfram.com/FrequencyModulatedContinuousWaveFMCWRadar/. [Accessed: Dec. 18, 2025].

8 Appendix

All MATLAB code and relevant resources for the project can be found at:
https://github.com/Octacles7/FMCW-Radar-Simulation.