



Faculty of Engineering
Cairo University

Electronics and Electrical Communications Department

Second Year – First Semester (Fall)

Analysis of Discrete and Continuous Time Signals



FMCW RADAR DESIGN

*For Target Detection
and Velocity Estimation*

EECG-231 Final Project

Under Supervision of:

Dr. Mohammed Motaz Abdelghany

Dr. Michel Milik

Eng. Sayed Kamel

Muhammad Sameer Abdelhamid
91240662

Mohamed Waleed Hosny
91241081

Table of Contents

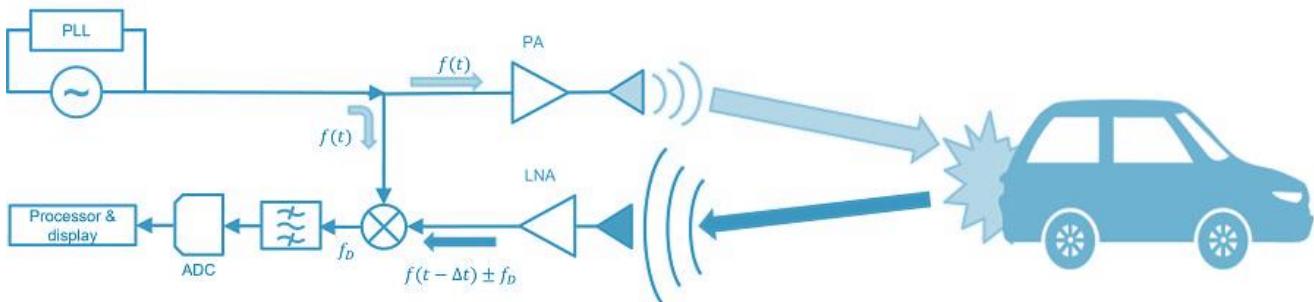
Introduction	2
Radar Systems and FMCW Radar (Concept)	3
System Specifications	5
Signal Processing Chain	6
Mapping Each Code Segment to Its Corresponding Processing Step	8
Test Scenarios.....	21
Test Outputs & Plots: Scenario 1 (Normal / Default Case)	21
Test Outputs & Plots: Scenario 2	25
Test Outputs & Plots: Scenario 3	28
Test Outputs & Plots: Scenario 4	31
Test Outputs & Plots: Scenario 5	34
Test Outputs & Plots: Scenario 6	37
Test Outputs & Plots: Scenario 7	40
Test Outputs & Plots: Scenario 8	43
Test Outputs & Plots: Scenario 9	46
Test Outputs & Plots: Scenario 10	49
Design Limitations.....	50
Conclusion	51
References.....	52
Appendix.....	52

Introduction

The modern automotive and industrial sector depends on **Frequency Modulated Continuous Wave (FMCW)** radar systems because these systems enable precise target range and velocity measurement of multiple targets at the same time. The measurement of distance in FMCW radars differs from pulsed radar systems because they use a continuous signal with changing frequency known as a **chirp**, this means more power but a better resolution and results.

The system contains a complete digital signal processing system which operates at **77 GHz for Long-Range Radar (LRR)** applications. The simulation environment uses a high-bandwidth waveform (1 GHz) which spans 512 pulses during the coherent processing interval. The main goal of this project involves creating a complete radar system which will be tested through simulation and validation of its entire operation from signal generation to environmental simulation and detection system development.

The system performance evaluation includes ten operational tests which check **Range and Doppler resolution boundaries** and **Constant False Alarm Rate (CFAR)** detection in noisy conditions and object identification between approaching and receding targets. The project shows essential trade-offs between hardware parameters and signal processing results through its implementation of theoretical radar equations in MATLAB.



Radar Systems and FMCW Radar (Concept)

Basic Radar Principle

A basic principle of radar is based on emitting Radio Detection and Ranging signals towards a target and detecting an echo reflected from it. Using parameters measured in the received echo, namely its delay, frequency deviation, or amplitude, it's possible to calculate coordinates, speed, and size of observed targets. This principle has been used to create radar technology applications in various fields.

FMCW Radar Operation

The FMCW radar emits a continuous wave whose frequency varies in a linear fashion with time, with a corresponding name of **Linear Frequency Modulation (LFM)** or **Chirp**.

1. The Chirp Signal

The transmitted frequency starts at a base frequency (f_{min}) and sweeps across a bandwidth (B) during the chirp time (T_{chirp}). The rate of this change is defined as the **Slope (S)**:

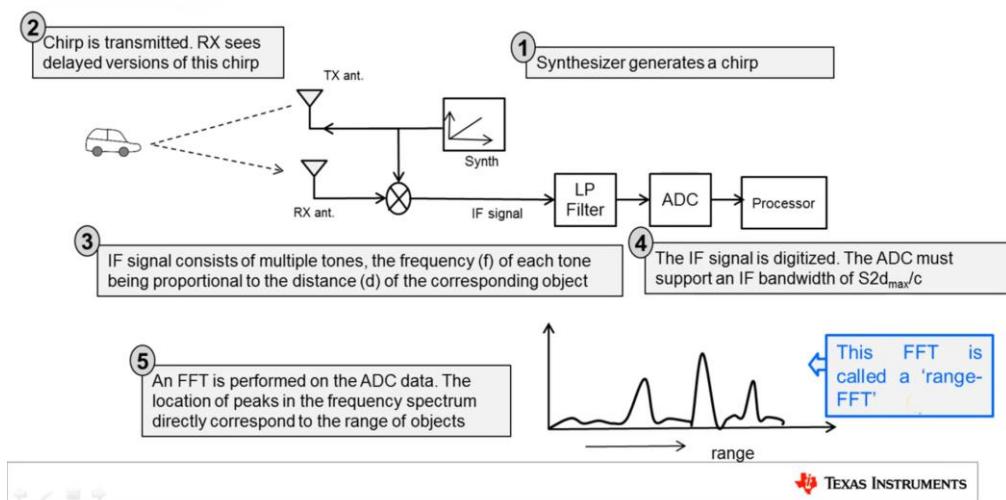
$$S = \frac{B}{T_{chirp}}$$

2. Range Extraction

When the transmitted signal hits a target, it reflects and returns to the receiver with a time delay (τ), where $\tau = \frac{2R}{c}$. Because the transmitter is constantly sweeping its frequency, the frequency of the transmitted signal will have changed by the time the echo returns.

By mixing the transmitted and received signals, the system produces a **beat** frequency (f_b), which represents the frequency difference between the two. The target range (R) is directly proportional to this beat frequency:

$$R = \frac{c \cdot f_b}{2S}$$



Reference to Texas Instruments series for designing FMCW Radar

3. Velocity Extraction

To measure velocity, the radar transmits a sequence of chirps (a coherent processing interval). While the range is determined by the frequency shift within a single chirp (fast-time), the relative velocity (v) of a target causes a minute, constant phase shift from one chirp to the next (slow-time). This is known as the **Doppler Effect**.

The Doppler frequency shift (f_D) is calculated based on the carrier frequency (f_c) and the target's radial velocity:

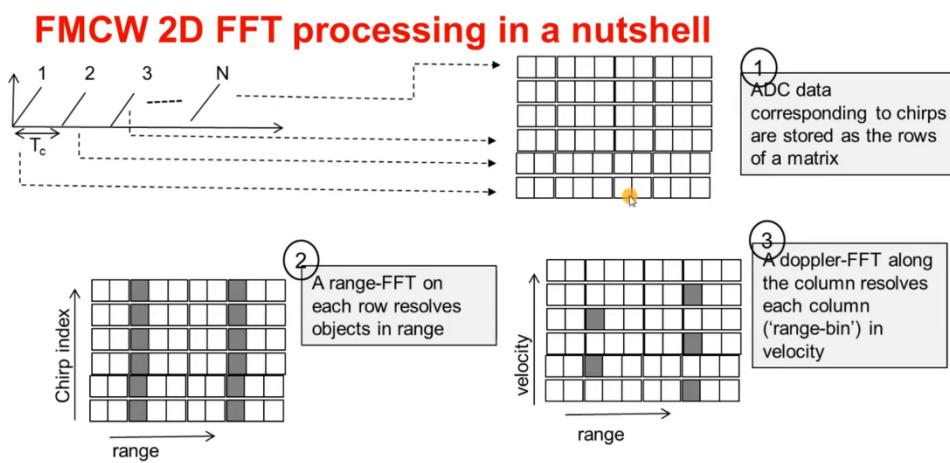
$$f_D = \frac{2 \cdot v \cdot f_c}{c}$$

By performing a Fourier Transform across the sequence of chirps, the system extracts this phase progression to determine the target's speed and direction (approaching or receding).

Fast-Time and Slow-Time Processing (Range–Velocity Estimation)

In FMCW radar processing, the received signal is naturally organized into a two-dimensional structure defined by fast-time and slow-time axes. The fast-time axis represents samples taken within a single chirp and captures the time delay caused by target range. Applying a Fourier Transform along this dimension converts the beat frequency into a range profile.

The slow-time axis corresponds to successive chirps transmitted over the coherent processing interval, where each target introduces a consistent phase progression due to its radial velocity. By applying a second Fourier Transform across the slow-time dimension, the Doppler frequency is extracted. Combining both transforms results in a Range–Velocity Map, which provides a two-dimensional representation of target distance and speed simultaneously.



System Specifications

The performance of an FMCW radar is dictated by its waveform parameters, which define the system's reach, precision, and processing requirements. Below are the key specifications implemented in the project:

Waveform and Frequency Parameters

Parameter	Symbol	Value	Significance
Carrier Frequency	f_c	76.5 GHz	Determines the wavelength ($\lambda \approx 3.9$ mm) and the sensitivity to Doppler shifts.
Bandwidth	B	1 GHz	Directly defines the Range Resolution ; higher bandwidth allows for closer target separation.
Chirp Duration	T_{chirp}	8 μ s	Controls the frequency slope and the time available for signal integration per pulse.
Sweep Slope	S	125 MHz/ μ s	The rate of frequency change; used to map beat frequencies to physical distances.
Pulse Repetition Interval	PRI	8 μ s	The time between the start of successive chirps, determining the Maximum Unambiguous Velocity .

Sampling and Processing Parameters

Parameter	Symbol	Value	Significance
Sampling Period	T_s	0.5 ns	Defines the time resolution of the Analog-to-Digital Converter (ADC).
Sampling Frequency	f_s	2 GHz	Must satisfy the Nyquist criterion to prevent aliasing of the high-frequency beat signals.
Samples per Chirp	N_{sample}	16,000	The number of data points collected during the fast-time interval for range processing.
Number of Chirps	N_p	512	The "Slow-time" dimension defines the Velocity Resolution and integration gain.

Operational Constraints

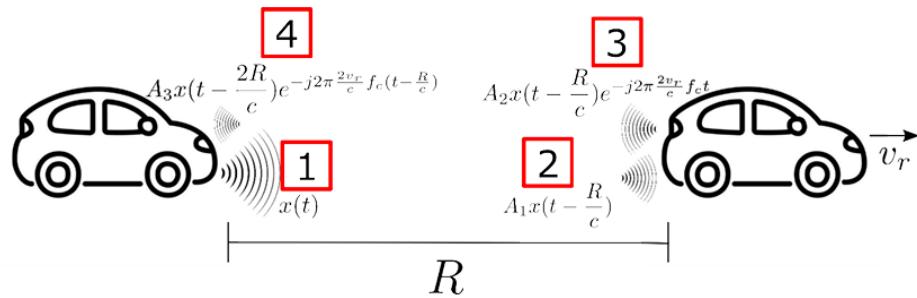
Parameter	Symbol	Value	Significance
Maximum Range	R_{max}	250 m	The design limit for the radar's reach based on expected automotive requirements.
Max Round Trip Time	RTT_{max}	1.67 μ s	The maximum delay expected for a signal to return from R_{max} .
Gating Window	n_{start}	3,333 samples	Calculated from RTT_{max} to ensure the FFT only processes the overlap region where Tx and Rx signals coexist.

Signal Processing Chain

The signal processing chain converts raw electromagnetic reflections into actionable target data. It is grossly critical for maintaining signal integrity and ensuring very accurate extraction of spatial as well as kinematic information from the signals.

1. Transmission (TX) and Receiving (RX)

The signal processing chain converts raw electromagnetic reflections into actionable target data. It is grossly critical for maintaining signal integrity and ensuring very accurate extraction of spatial as well as kinematic information from the signals.



Where The Transmitted Signal is:

$$x(t) = \sum_{m=1}^{m=512} P(t - m \cdot PRI)$$

$$P(t) = 1 \cdot e^{j \cdot 2\pi(f_{min}t + s \cdot t^2 / 2)} [\mathbf{u}(t) - \mathbf{u}(t - T_{chirp})]$$

Then The Received Signal is:

$$y(t) = \sum_{k=1}^L A_k \cdot x\left(t - \frac{2R_k}{c}\right) \cdot e^{j \cdot 2\pi f_{Doppler} \cdot t}, \quad L \text{ is the number of objects}$$

2. Mixing (De-chirping)

The mixing stage forms the heart of the FMCW processing chain. This is performed through multiplying the transmitted signal by the complex conjugate of the received signal. Now we can say that the system has actually done "de-chirping." This practically takes high-frequency radar signals down to baseband and creates a Beat Frequency (f_b). This profoundly lowers hardware requirements down the line for digital processing, while still capturing the phase information that is needed for Doppler estimation.

$$z_{mix}(t) \cong x(t) \cdot y^*(t - R_{TT}) \cong \bar{A} e^{j \cdot 2\pi S R_{TT} \cdot t} \cdot e^{j \cdot 2\pi f_D \cdot (t - R_{TT}/2)}$$

3. Gating and Windowing

Before frequency analysis, the mixed signal is Gated, at which point it cuts off the period of initial transience (Round Trip Time) where TX and RX ramps have not yet overlapped. Gating allows the FFT to operate only on valid, stationary beat frequencies, thus preventing spectral leakage and "smearing" due to processing on non-overlapping segments.

4. Fast-Fourier Transform (FFT)

Adopting a 2D FFT framework, the system disentangles the range from the velocity:

- **Range FFT (Fast-Time):** At play across the samples of a single chirp, converting the time-domain beat signal into a frequency domain wherein each spectral peak is indicative of a shot at his/her target range.
- **Doppler FFT (Slow-Time):** Using across the sequence of chirps. It sees exceeding small changes in phase due to target motion in between pulses. The resultant Range-Doppler Map gives a very localized energy peak for every unique object in the radar's sight.

5. Detection (CFAR)

Indeed, it is filled with thermal noise and clutter in the radar environment; thus, the static or fixed threshold cannot be effective. In the Detection block, a Constant False Alarm Rate (CFAR) algorithm is used. In this way, it keeps a constant probability of detection because the system will calculate a dynamic local noise floor using "training cells" around a target. This step is essential in separating true targets from random noise spikes or interference.

Mapping Each Code Segment to Its Corresponding Processing Step

This section bridges the theoretical processing chain with the specific software implementation found in `main_radar_project.m` and `radar_config.m`.

1. Scenario Configuration (`radar_config.m`)

This segment initializes the ground truth by defining target range, velocity, amplitude, and noise levels (SNR) to simulate diverse operational environments and test system performance limits.

```
%% %%%%%%%%%%%%%% Radar Scenarios Configuration
switch scenario_id
    case 1 % Normal
        R_targets = [100, 220]; v_targets = [30, 110]; A_targets = [1, 0.8];
        SNR_target_db = 15;
    case 2 % Low SNR
        R_targets = [100, 220]; v_targets = [30, 110]; A_targets = [1, 1];
        SNR_target_db = 5;
    case 3 % Stationary
        R_targets = [100, 220]; v_targets = [0, 0]; A_targets = [1, 0.7];
        SNR_target_db = 15;
    case 4 % Receding/Approaching Mix
        R_targets = [100, 220]; v_targets = [-50, 110]; A_targets = [1, 1];
        SNR_target_db = 15;
    case 5 % Close Range
        R_targets = [120, 122]; v_targets = [30, 110]; A_targets = [1, 0.9];
        SNR_target_db = 15;
    case 6 % Close Velocity
        R_targets = [100, 220]; v_targets = [30, 32]; A_targets = [1, 1];
        SNR_target_db = 15;
    case 7 % High Velocity Difference
        R_targets = [100, 220]; v_targets = [-100, 120]; A_targets = [1, 1];
        SNR_target_db = 15;
    case 8 % Unambiguous Range
        R_targets = [10, 240]; v_targets = [30, 90]; A_targets = [1, 0.5];
        SNR_target_db = 15;
    case 9 % Multiple Objects
        R_targets = [10, 105, 75, 210]; v_targets = [12, -70, 0, 97]; A_targets = [1,
        0.4, 0.8, 0.6];
        SNR_target_db = 15;
end
```

2. Scenarios Choice

Implementation: Main Script Header

This section acts as the System Controller, where the user selects a specific test case (`scenario_id`) and toggles environmental noise (`use_noise`) to direct the simulation flow and verification logic.

3. Parameters

Implementation: Global Variable Definitions

This block defines the **Physical and Operational Constraints** of the radar hardware, calculating essential constants like the frequency slope ($S = \frac{bw}{T_{chirp}}$) and sampling requirements to determine the system's resolution and maximum sensing boundaries.

4. Transmitted Signal

Implementation: Signal Synthesis and Pulse Train Generation

This section generates the **Linear Frequency Modulated (LFM)** waveform, the foundation of FMCW radar. The code synthesizes a single complex exponential chirp (`x_chirp`) where the frequency increases linearly over time via a quadratic phase term. This single pulse is then replicated into a 2D matrix (`x_tx`) to represent a continuous stream of 512 chirps.

The accompanying plots (Figure 1) visualize the **Instantaneous Frequency** ($f_{inst} = f_{min} + S \cdot t$), confirming a perfect sawtooth ramp from -0.5 GHz to 0.5 GHz. This linearity is critical, as any deviation would cause spectral spreading and degrade the accuracy of the range estimation.

5. Received Signal

➤ **Implementation: Target Modeling and Environmental Effects**

This block simulates the physics of radar reflection by applying spatial and kinematic transformations to the transmitted signal. For each target defined in the configuration, the code calculates the round-trip time delay ($\tau = \frac{2R}{c}$) and the Doppler frequency shift ($f_D = 2v \cdot \frac{f_c}{c}$).

The code implements two distinct physical effects:

- **Range Modeling:** The signal is shifted in "Fast-Time" (samples) by $N_{\text{delay_k}}$, representing the time-of-flight.
 - **Velocity Modeling:** A phase shift is applied across "Slow-Time" (chirp indices) using the Doppler term, where the phase rotates from chirp to chirp based on the target's radial speed.

Finally, the script uses the **Superposition Principle** to sum all individual target reflections and injects **Additive White Gaussian Noise (AWGN)** based on the specified SNR. This creates a realistic, noisy raw data matrix (y_{rx}) ready for processing.

➤ Implementation: Time-Domain Visualization

This section verifies the simulated environment by visualizing the two primary dimensions of radar data. **Figure 2** (in the test scenarios section) highlights the time-of-flight delay (τ) between the transmitted and received waveforms. **Figure 3** distinguishes between **Fast-Time** (intra-chirp delay for range) and **Slow-Time** (inter-chirp phase rotation for velocity), confirming that the target physics were correctly modeled before frequency-domain processing.

```
% % % Received Signal Plots
% % % =====
figure(2);
subplot(2,1,1);
plot(real(x_tx(:,1)), 'b');
xlabel('Time (\mu s)');
ylabel('Magnitude');
title('Transmitted Signal');
grid on;

subplot(2,1,2);
plot(real(y_rx(:,1)), 'r');
xlabel('Time (\mu s)');
ylabel('Magnitude');
title('Received Signal (Delayed)');
grid on;

% Verify Doppler Plots
figure(3);
subplot(2,1,1);
plot(real(y_rx(:,1)));
xlabel('Time (\mu s)');
title('Fast-time: Range Effect (Single Chirp)');
grid on;

n_test = round(2 * R_targets(1) / c / T_sample) + 50;
subplot(2,1,2);
plot(real(y_rx(n_test,:)));
xlabel('Chirp Index');
title('Slow-time: Velocity / Doppler Effect (Across Chirps)');
grid on;
```

6. Signal Processing

Implementation: Mixing (Dechirping) and Gating

This stage implements the **Dechirping** process by multiplying the transmitted signal with the complex conjugate of the received signal. This operation extracts the **Beat Frequency (f_b)**, a baseband signal where distance is linearly mapped to frequency.

To ensure signal integrity, the code performs **Gating (Windowing)** by discarding the initial samples defined by `n_start`. This removes the transient "idle period" where the transmitted ramp and the delayed echo do not overlap, preventing spectral leakage and ensuring that the subsequent FFT only processes valid, stationary beat tones.

7. FFT in the Fast-time Axis → Range

➤ Implementation: Range Estimation and Spectral Analysis

This block performs the **Range FFT** by transforming the gated beat signal from the time domain into the frequency domain. By applying the FFT across the "Fast-time" dimension (rows), the system identifies the specific beat frequencies (f_b) present in the signal.

The code then maps these frequencies to physical units using the radar range equation $R = \frac{c \cdot f_b}{2S}$. To optimize computational efficiency, the FFT size is rounded to the next power of two. The resulting **Range Profile** (Figure 5) displays distinct magnitude peaks, where each peak's position on the x -axis directly corresponds to the radial distance of a detected target.

```
% % % FFT in the Fast-time Axis --> Range
% % % =====
tic;
Nfft_r = 2^nextpow2(size(z_windowed,1));           % for number of points to be a power of 2
z_r = fft(z_windowed, Nfft_r, 1);                  % FFT across rows
z_r = z_r(1:Nfft_r/2, :);
time = time + toc;

f_beat = (0:Nfft_r/2-1).' * (f_sample / Nfft_r);    % Beat Frequency
R_axis = (c * f_beat) / (2 * S);                      % Range Axis

figure(5);
range_data = abs(z_r(:,1));                          % first chirp only for range detection
plot(R_axis, range_data);
hold on;
xlabel('Range (m)');
ylabel('Magnitude');
title('Range Peaks');
xlim([0 R_max]);
grid on;
```

➤ Implementation: Target Localization and Labeling

This segment automates target identification by isolating significant local maxima within the range spectrum. The `findpeaks` function filters out low-level noise and sidelobes using a relative height threshold (25% of the maximum peak). By mapping the indices of these peaks (`locs_r`) back to the `R_axis`, the code provides precise numerical distance estimates, which are then annotated directly on the plot for immediate verification of radar accuracy.

```
% % % Automated Detection of peaks
% % % =====
[pks, locs_r] = findpeaks(range_data, 'MinPeakHeight', max(range_data)*0.25,
'MinPeakDistance', 2);
plot(R_axis(locs_r), pks, 'ro');

for i = 1:length(pks)
    text(R_axis(locs_r(i)), pks(i)*1.05, sprintf('%1f m', R_axis(locs_r(i))), ...
        'HorizontalAlignment', 'center');
end
```

➤ Implementation: Cell-Averaging Constant False Alarm Rate Detection

This block implements a robust detection algorithm designed to distinguish true targets from non-stationary background noise. Using a sliding window approach, the code evaluates a **Cell Under Test (CUT)** by comparing its power to a local noise floor calculated from surrounding **Training Cells**.

The implementation utilizes **Greatest-Of (GO) Logic**, which selects the maximum average power between the left and right noise windows to set a conservative threshold. **Guard Cells** immediately adjacent to the CUT are excluded from the noise calculation to prevent target energy from "smearing" into the threshold and causing missed detections. The resulting adaptive threshold (Figure 6) ensures a consistent detection probability even as noise levels fluctuate across the range axis.

```
% % % Range CA-CFAR
% % % =====
r_train = 15;
r_guard = 3;
r_offset = 2.2;

sig_r = abs(z_r(:, 1));
N_r = length(sig_r);
r_threshold = zeros(N_r, 1);
r_detect = zeros(N_r, 1);

for i = (r_train + r_guard + 1) : (N_r - (r_train + r_guard))
    % Split noise cells into Left and Right windows
    noise_left = sig_r(i-r_train-r_guard : i-r_guard-1);
    noise_right = sig_r(i+r_guard+1 : i+r_train+r_guard);

    % Greatest-Of Logic: Picks the noisier side to set a safer threshold
    r_threshold(i) = max(mean(noise_left), mean(noise_right)) * r_offset;

    if sig_r(i) > r_threshold(i)
        r_detect(i) = sig_r(i);
    end
end

figure(6);
plot(R_axis, sig_r, 'Color', 'w'); hold on;
plot(R_axis, r_threshold, 'r--');
stem(R_axis, r_detect, 'g', 'Marker', 'none');
xlabel('Range (m)');
xlim([0, R_max]);
ylabel('Magnitude');
title('CA-CFAR for Range');
legend('Signal', 'Threshold', 'Detections'); grid on;
```

8. FFT in the Slow-time Axis → Velocity

➤ Implementation: Doppler Processing and Velocity Mapping

This segment performs the **Doppler FFT** to extract the velocity of all detected targets. By processing the data across the "Slow-time" dimension (columns), the system measures the phase shift from one chirp to the next, which is directly proportional to the target's radial velocity.

The code utilizes `fftshift` to center the zero-velocity component, allowing for the detection of both approaching (positive Doppler) and receding (negative Doppler) objects. The Doppler frequencies are converted to physical units using the relationship $v = \frac{c \cdot f_D}{2f_c}$. The resulting **Velocity Profile** (Figure 7) collapses the range data to show the distribution of speeds within the environment, identifying the kinetic state of every object in the radar's field of view.

```
% % % FFT in the Slow-time Axis --> Velocity
% % % =====
tic;
Nfft_v = 512; % Number of chirps
z_v = fftshift(fft(z_r, Nfft_v, 2), 2);
time = time + toc;

fD_axis = (-Nfft_v/2 : Nfft_v/2-1) / (Nfft_v * PRI);
v_axis = -(c * fD_axis) / (2 * f_c); % -ve means that the target is receding

% % % Velocity Profile
% % % =====
figure(7);
vel_data = max(abs(z_v), [], 1);
plot(v_axis, vel_data);
xlabel('Velocity (m/s)');
ylabel('Magnitude');
title('Velocity Peaks');
xlim([-150 150]);
grid on;
hold on;
```

➤ Implementation: Velocity Localization

This code identifies valid motion components by locating local maxima in the Doppler spectrum. By applying a 25% magnitude threshold, it isolates actual target velocities from noise and marks them on the velocity profile for precise kinetic tracking.

```
% % % Automated Detection of peaks
% % % =====
[pks_v, locs_v] = findpeaks(vel_data, 'MinPeakHeight', max(vel_data)*0.25);
plot(v_axis(locs_v), pks_v, 'ro');
```

➤ Implementation: Kinematic Classification

This section classifies targets by interpreting the sign of the measured velocity v . It categorizes motion into three states to provide immediate situational awareness:

- **Receding ($v > 0.5$):** Target is moving away.
- **Approaching ($v < -0.5$):** Target is moving closer.
- **Stationary ($|v| \leq 0.5$):** Target is at rest relative to the radar.

These classifications are then annotated on the plot to translate spectral data into logical tracking information.

```
for i = 1:length(pks_v)
    v_val = v_axis(locs_v(i));

    if v_val > 0.5
        status = '(Receding)';
    elseif v_val < -0.5
        status = '(Approaching)';
    else
        status = '(Stationary)';
    end

    label = [num2str(v_val, '%.1f'), ' m/s', status];
    text(v_val, pks_v(i)*1.05, label, 'HorizontalAlignment', 'center', ...
        'Color', 'r', 'FontWeight', 'bold');
end
```

➤ **Implementation:** Doppler Domain Adaptive Detection

This segment applies the **CA-CFAR** algorithm to the velocity dimension to filter out the noise floor and spectral leakage. Similar to the range detection, it uses a sliding window of **Training Cells** to calculate a local average, which is then scaled by an offset to create a dynamic threshold.

By utilizing **Guard Cells**, the system prevents the target's own energy from inflating the noise estimate, ensuring that even small, fast-moving targets are reliably detected. The resulting output (Figure 8) effectively isolates true Doppler peaks from the surrounding interference.

```
% % % Velocity CA-CFAR
% % % =====
v_train = 15;
v_guard = 2;
v_offset = 1.3;

sig_v = max(abs(z_v), [], 1);
N_v = length(sig_v);
v_threshold = zeros(1, N_v);
v_detect = zeros(1, N_v);

for i = (v_train + v_guard + 1) : (N_v - (v_train + v_guard))
    % Extract training cells
    v_noise = [sig_v(i-v_train-v_guard : i-v_guard-1), sig_v(i+v_guard+1 : i+v_train+v_guard)];
    v_threshold(i) = mean(v_noise) * v_offset; % Compute threshold
    if sig_v(i) > v_threshold(i)
        v_detect(i) = sig_v(i);
    end
end

figure(8);
plot(v_axis, sig_v, 'Color', [0.5 0.5 0.5]);
hold on;
plot(v_axis, v_threshold, 'r--');
stem(v_axis, v_detect, 'g', 'Marker', 'none');
xlabel('Velocity (m/s)');
xlim([-150 150]);
ylabel('Magnitude');
title('CA-CFAR for Velocity');
legend('Signal', 'Threshold', 'Detections');
grid on;
```

9. Range-Doppler Map (2D FFT)

Implementation: Spatial-Temporal Data Visualization

This heatmap displays the final **2D FFT** output, mapping targets simultaneously across range and velocity. By using a logarithmic decibel (**dB**) scale, the plot reveals both high-intensity reflections and subtle targets that would be invisible in a linear scale.

- **Y-Axis:** Distance from radar (Range).
- **X-Axis:** Speed and direction (Velocity).
- **Color:** Signal strength (RCS).

```
% % % Range Velocity Map, in dB
% % % =====
figure(9);
imagesc(v_axis, R_axis, 20*log10(abs(z_v)));
xlabel('Velocity (m/s)');
ylabel('Range (m)');
title('Range-Velocity Map (dB Scale)');
colorbar;
axis xy;
```

10. Performance Evaluation & Reporting

Implementation: Statistical Comparison and Ground Truth Verification

The final stage of the pipeline quantifies the radar's accuracy by comparing the extracted spectral data against the initial simulation parameters. This step transitions from signal processing to actionable data reporting.

- **Range Validation:** Matches the identified FFT peaks to the ground truth distances (`R_targets`), calculating the absolute error relative to the theoretical range resolution.
- **Velocity Classification:** Evaluates the Doppler shift to assign a kinematic status (**Approaching**, **Receding**, or **Stationary**) based on the sign and magnitude of the detected velocity.
- **Temporal Analysis:** Measures the total execution time (`time`) for the mixing, FFT, and CFAR stages to ensure the algorithm meets the latency requirements for real-time automotive or industrial applications.

```
% % % Table
% % % =====

% Range Table
true_R = R_targets(:);
matched_R = arrayfun(@(x) R_axis(locs_r(find(abs(R_axis(locs_r) - x) == ...
    min(abs(R_axis(locs_r) - x)), 1))), true_R);

T1 = table((1:length(true_R))', matched_R, true_R, abs(matched_R - true_R), ...
    'VariableNames', {'ID', 'Detected Range (m)', 'True Range (m)', 'Error (m)'});

% Velocity Table
true_V = v_targets(:);
matched_V = arrayfun(@(x) v_axis(locs_v(find(abs(v_axis(locs_v) - x) == ...
    min(abs(v_axis(locs_v) - x)), 1))), true_V);

status = repmat({'Stationary'}, length(true_V), 1);
status(matched_V > 0.5) = {'Receding'};
status(matched_V < -0.5) = {'Approaching'};

T2 = table((1:length(true_V))', matched_V, true_V, abs(matched_V - true_V), status, ...
    'VariableNames', {'ID', 'Detected Vel. (m/s)', 'True Vel. (m/s)', 'Error (m/s)', ...
    'Status'});

% Console Display
fprintf("____ FMCW Radar Detection Results" + ...
    "____\n\n");
disp('Table 1: Range Performance'); disp(T1);
disp('Table 2: Velocity Performance'); disp(T2);
fprintf('FFT Processing Time: %.4f ms\n\n', time * 1e3);

fprintf('Program ended...\n');
```

Test Scenarios

Scenario-Based Validation Strategy Testing the FMCW radar system under different scenarios is important to scientifically ascertain the robustness and reliability of the system beyond ideal situations. Multi-scenario testing evaluates the physical limit and algorithmic limit of the system by putting the signal processing pipeline in different target configurations. This strategy aims to establish detection limits at the extreme ranges and velocities while testing the object's capability to differentiate between closely spaced targets.

Moreover, these tests address the dynamic sensitivity of CA-CFAR under high noise conditions and verify the integrity of phase shift logic in classifying directional movement correctly. With the systematic variation of these parameters, the simulation engages in radar detection logic validation as being accurate and stable under high-stress configurations.

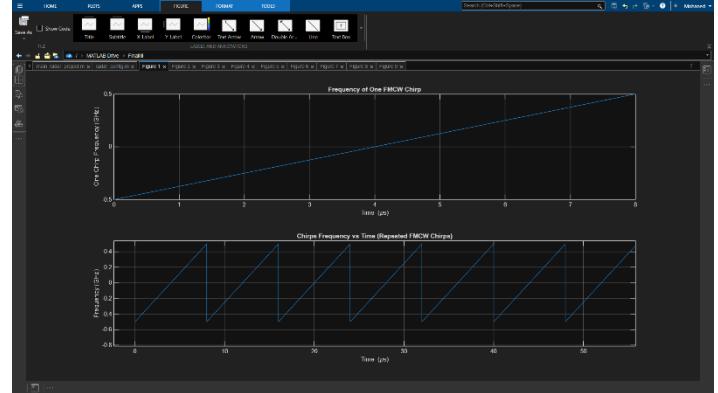
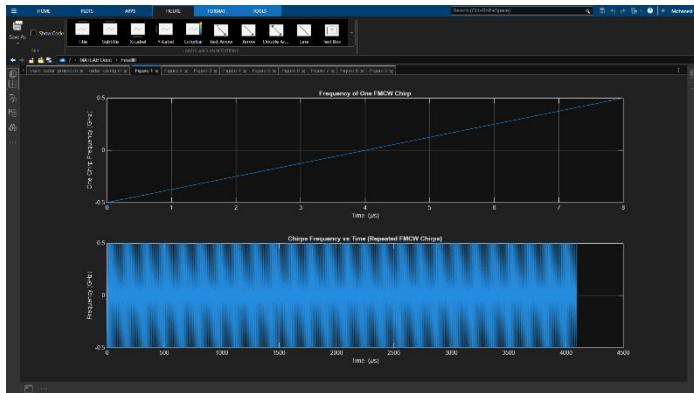
Test Outputs & Plots: Scenario 1 (Normal / Default Case)

Following the execution of the script for a standard target environment, the system generates nine figures illustrating the transformation from raw time-domain signals to a processed detection map.

Phase 1: Signal Generation & Reception

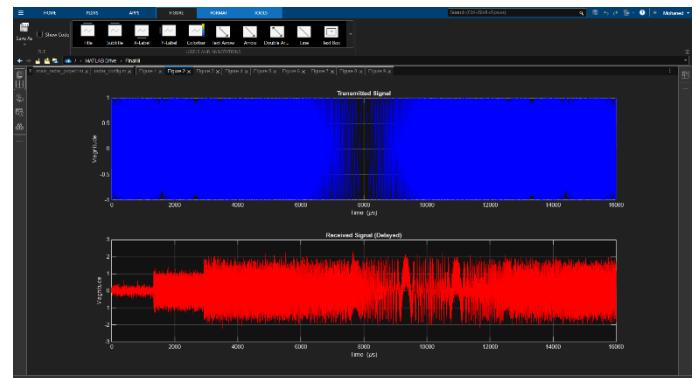
- Figure 1: Frequency of FMCW Chirps

Visualizes the generated linear frequency ramp of each FMCW chirp, transitioning from f_{min} to f_{max} over the duration of the coherent processing interval, which forms a sawtooth waveform when repeated over multiple chirps.



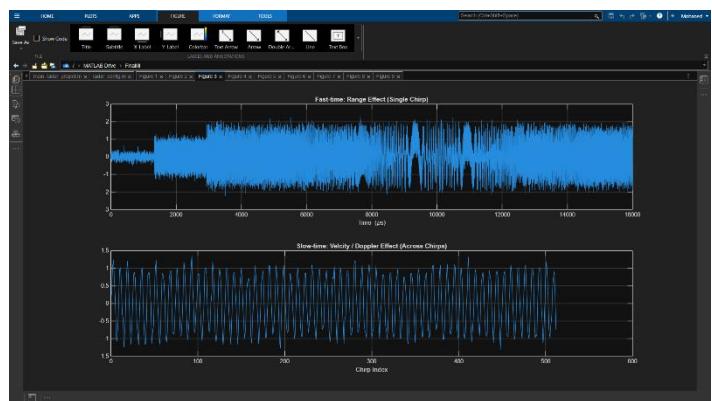
- Figure 2: Transmitted vs. Received Signal

A time-domain comparison showing the magnitude of the sent pulse against the delayed, attenuated echo returning from the environment.



- Figure 3: Fast-time and Slow-time Effects

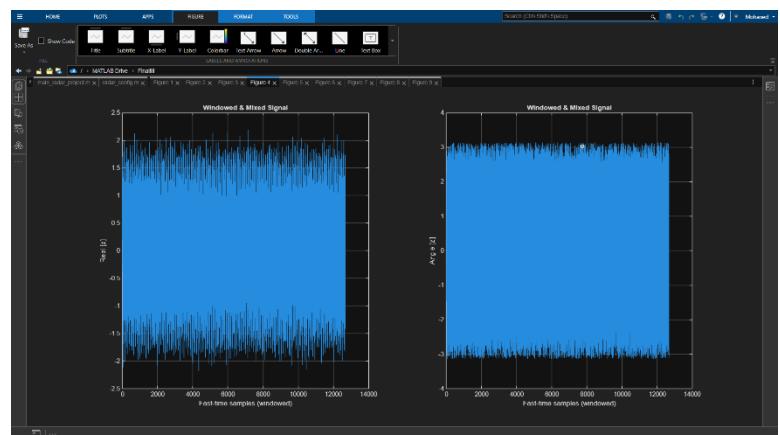
A dual plot verifying the range-induced delay within a single chirp and the velocity-induced phase rotation across multiple consecutive pulses.



Phase 2: Range Processing

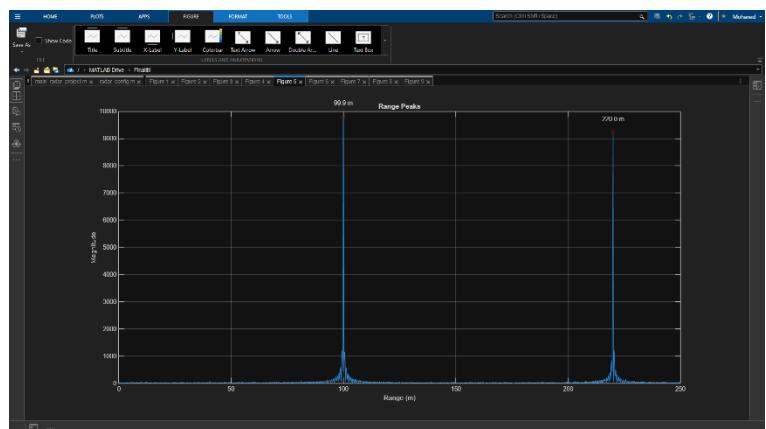
- Figure 4: Windowed & Mixed Signal

Displays the "beat signal" after dechirping and removing idle periods, resulting in a low-frequency tone ready for spectral analysis.



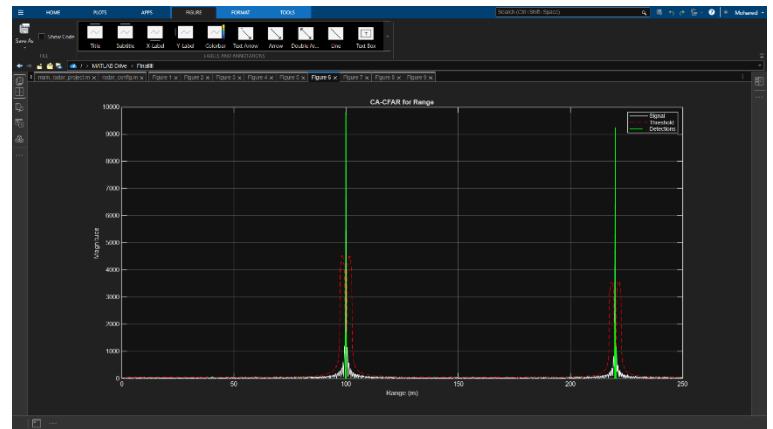
- Figure 5: Range Peaks

The output of the Fast-Time FFT, showing distinct magnitude spikes at the precise distances of the simulated targets.



- Figure 6: CA-CFAR for Range

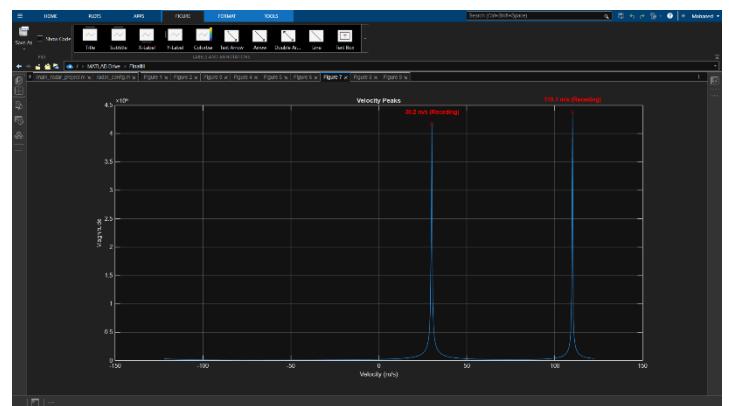
Shows the adaptive threshold (red dashed line) tracking the noise floor to successfully isolate target peaks from background interference.



Phase 3: Doppler Processing & Final Mapping

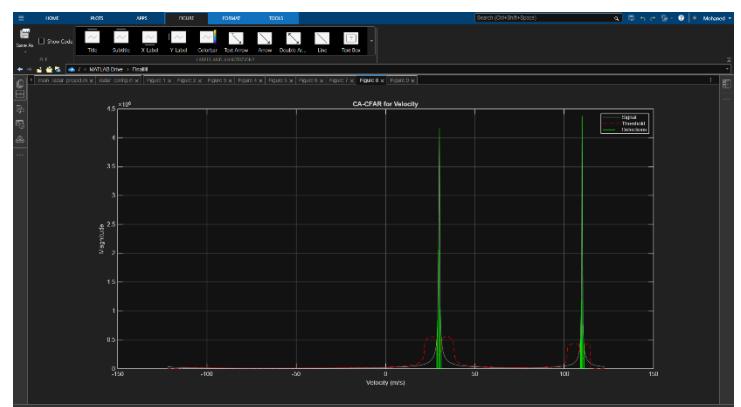
- Figure 7: Velocity Peaks

The output of the Slow-Time FFT, revealing the radial speeds of all detected objects in the field of view.



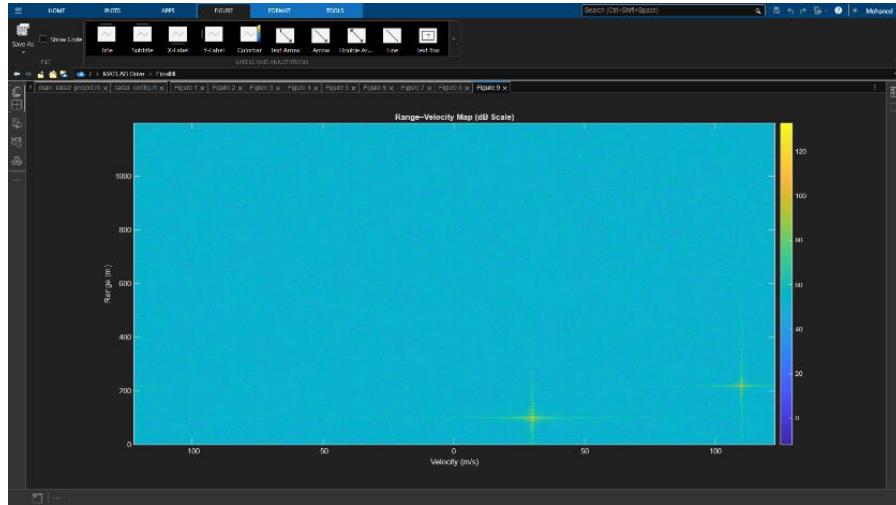
- Figure 8: CA-CFAR for Velocity

Demonstrates the automated detection of motion, applying a dynamic threshold to confirm target velocities while suppressing Doppler sidelobes.



- Figure 9: Range–Velocity Map (dB Scale)

The final 2D heatmap provides a complete spatial and kinematic overview, plotting target distance against speed in a single high-resolution frame.



Phase 4: Numerical Performance Summary

The following tables summarize the accuracy of the radar system by comparing detected values against the ground truth for Scenario 1.

Command Window				
Program started.....				
Tested Scenario is 1.				
Noise is ENABLED for this run.				
<u>FMCW Radar Detection Results</u>				
Table 1: Range Performance				
ID	Detected Range (m)	True Range (m)	Error (m)	
—	—	—	—	
1	99.902	100	0.097656	
2	220.02	220	0.019531	
Table 2: Velocity Performance				
ID	Detected Vel. (m/s)	True Vel. (m/s)	Error (m/s)	Status
—	—	—	—	—
1	30.159	30	0.15855	{'Receding'}
2	110.1	110	0.10263	{'Receding'}
FFT Processing Time: 211.2156 ms				
Program ended...				

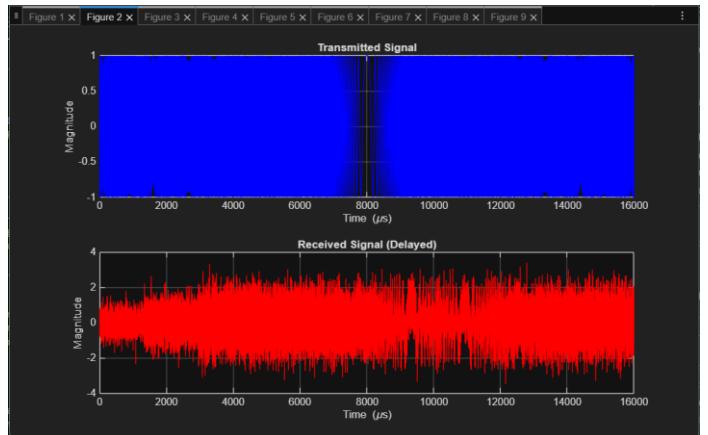
Now, other scenarios are going to be tested, and the figures provided are only the important figures that expose the differences between different scenarios.

Test Outputs & Plots: Scenario 2

This scenario evaluates the robustness of the FMCW radar system under degraded signal conditions by reducing the signal-to-noise ratio to 5 dB while keeping the same target ranges and velocities as the normal case.

Phase 1: Signal Generation & Reception

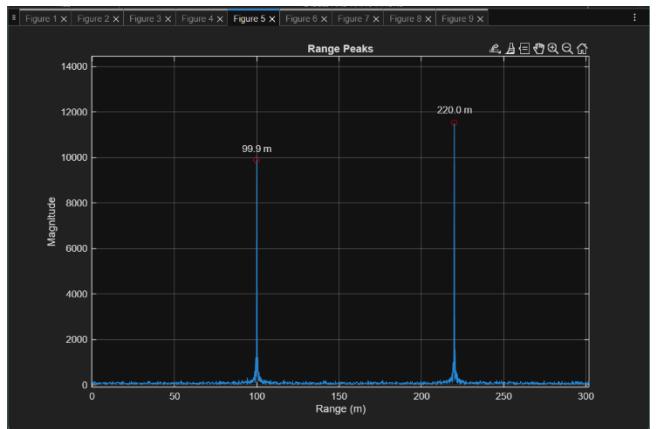
- The transmitted signal is the same as the normal case (scenario 1).
- The received signal got different delay rather than other scenarios.
- Figure 10: Transmitted vs. Received Signal



Phase 2: Range Processing

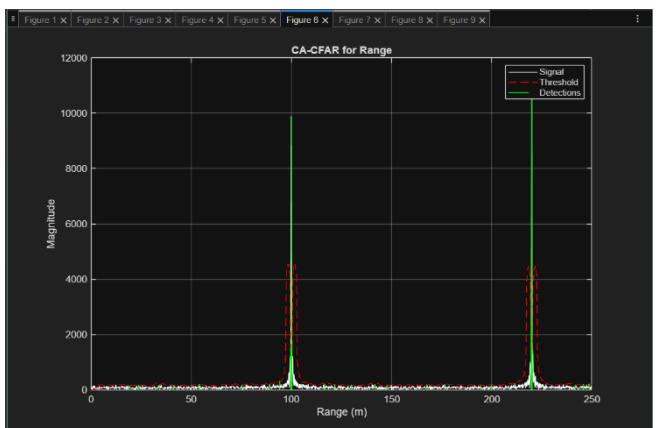
- Figure 11: Range Peaks

The output of the Fast-Time FFT, showing distinct magnitude spikes at the precise distances of the simulated targets.



- Figure 12: CA-CFAR for Range

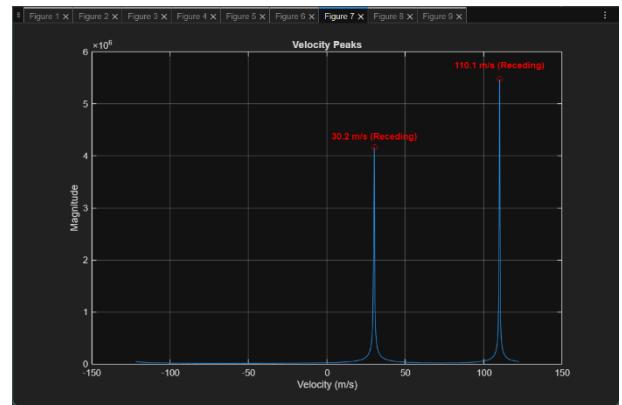
Shows the adaptive threshold (red dashed line) tracking the noise floor to successfully isolate target peaks from background interference.



Phase 3: Doppler Processing & Final Mapping

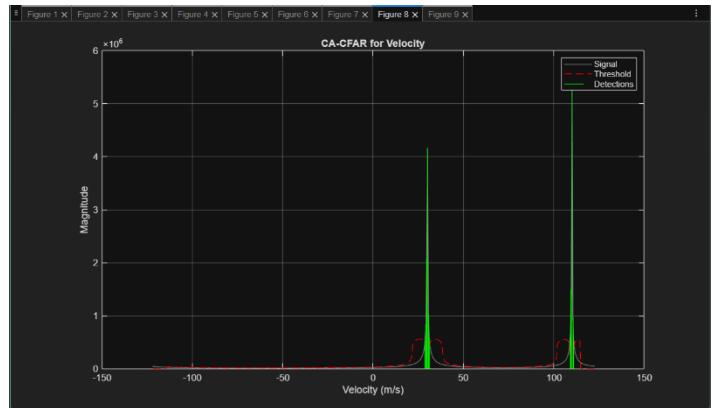
- Figure 13: Velocity Peaks

The output of the Slow-Time FFT, revealing the radial speeds of all detected objects in the field of view.



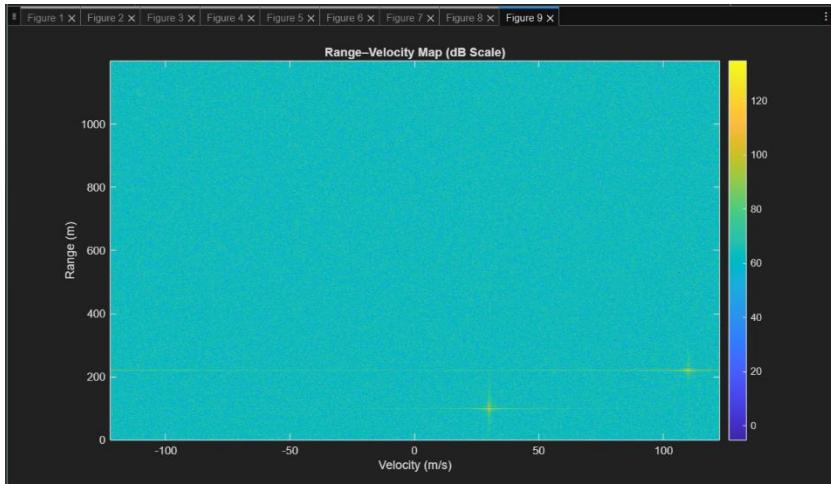
- Figure 14: CA-CFAR for Velocity

Demonstrates the automated detection of motion, applying a dynamic threshold to confirm target velocities while suppressing Doppler sidelobes.



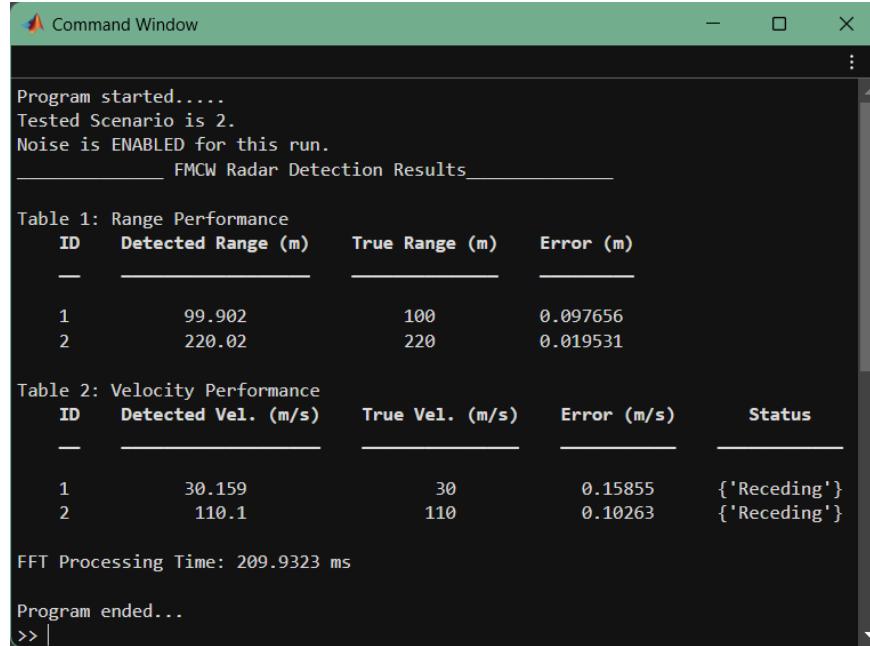
- Figure 15: Range–Velocity Map (dB Scale)

The final 2D heatmap provides a complete spatial and kinematic overview, plotting target distance against speed in a single high-resolution frame.



Phase 4: Numerical Performance Summary

The following tables summarize the accuracy of the radar system by comparing detected values against the ground truth for Scenario 2.



Program started.....
Tested Scenario is 2.
Noise is ENABLED for this run.
FMCW Radar Detection Results

ID	Detected Range (m)	True Range (m)	Error (m)
1	99.902	100	0.097656
2	220.02	220	0.019531

ID	Detected Vel. (m/s)	True Vel. (m/s)	Error (m/s)	Status
1	30.159	30	0.15855	{'Receding'}
2	110.1	110	0.10263	{'Receding'}

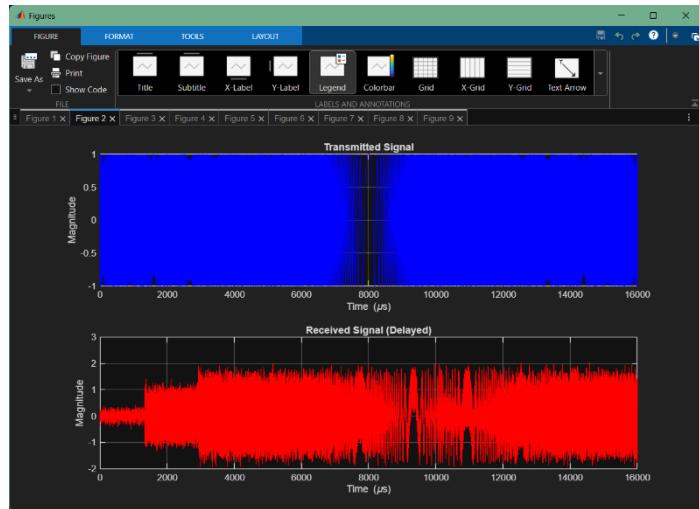
FFT Processing Time: 209.9323 ms
Program ended...
>> |

Test Outputs & Plots: Scenario 3

This scenario tests the radar's ability to distinguish stationary objects by setting all target velocities to zero.

Phase 1: Signal Generation & Reception

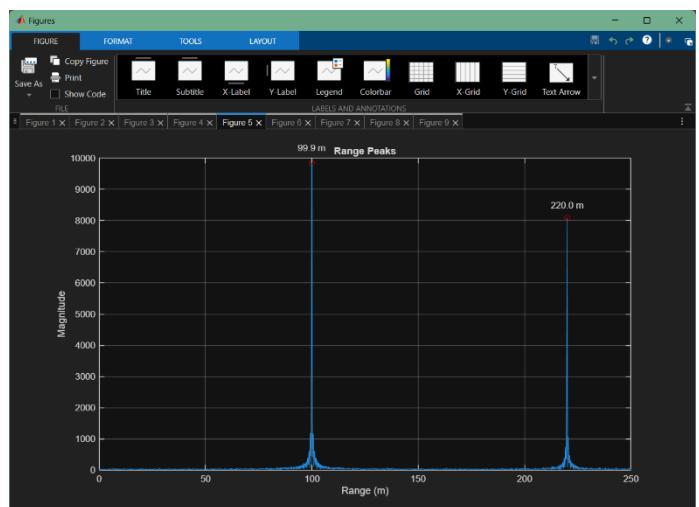
- The transmitted signal is the same as the normal case (scenario 1).
- The received signal got different delay rather than other scenarios.
- Figure 16: Transmitted vs. Received Signal



Phase 2: Range Processing

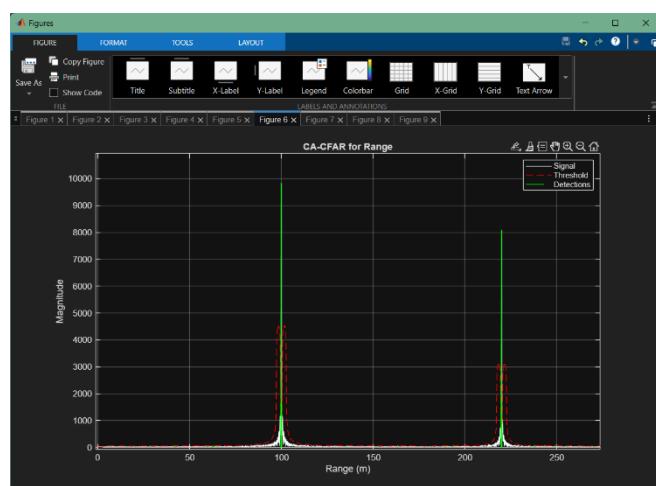
- Figure 17: Range Peaks

The output of the Fast-Time FFT, showing distinct magnitude spikes at the precise distances of the simulated targets.



- Figure 18: CA-CFAR for Range

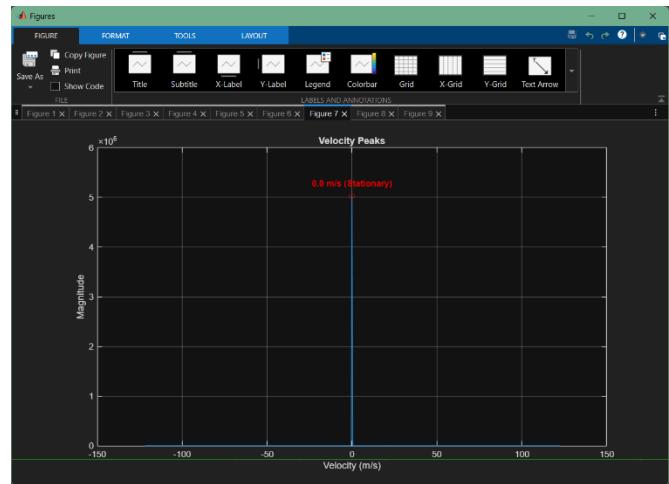
Shows the adaptive threshold (red dashed line) tracking the noise floor to successfully isolate target peaks from background interference.



Phase 3: Doppler Processing & Final Mapping

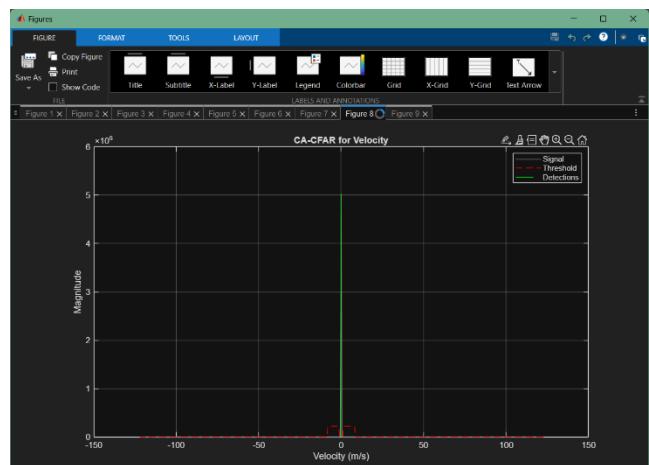
- Figure 19: Velocity Peaks

The output of the Slow-Time FFT, revealing the radial speeds of all detected objects in the field of view.



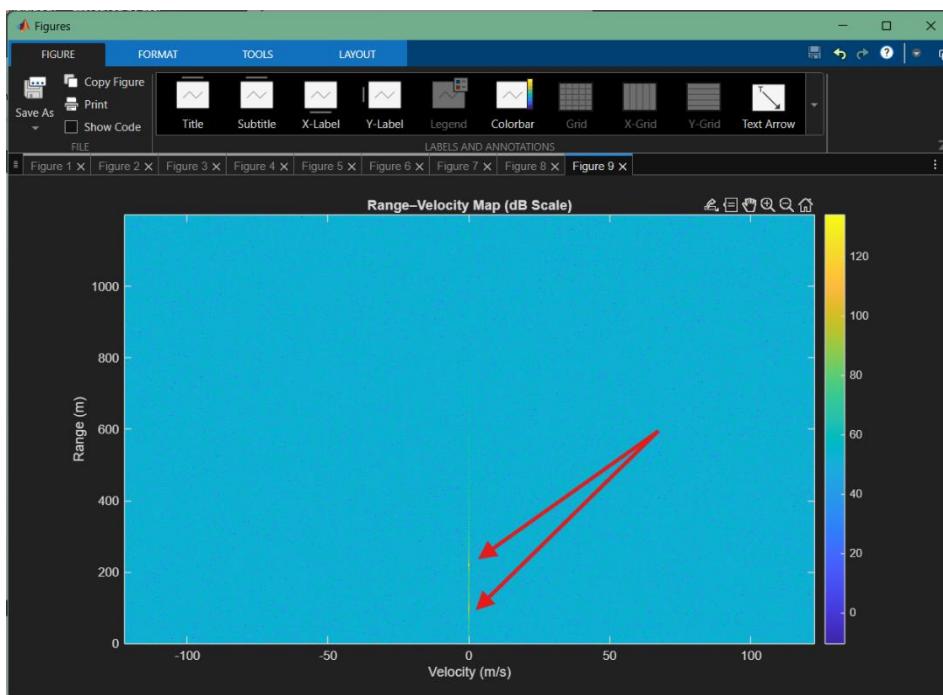
- Figure 20: CA-CFAR for Velocity

Demonstrates the automated detection of motion, applying a dynamic threshold to confirm target velocities while suppressing Doppler sidelobes.



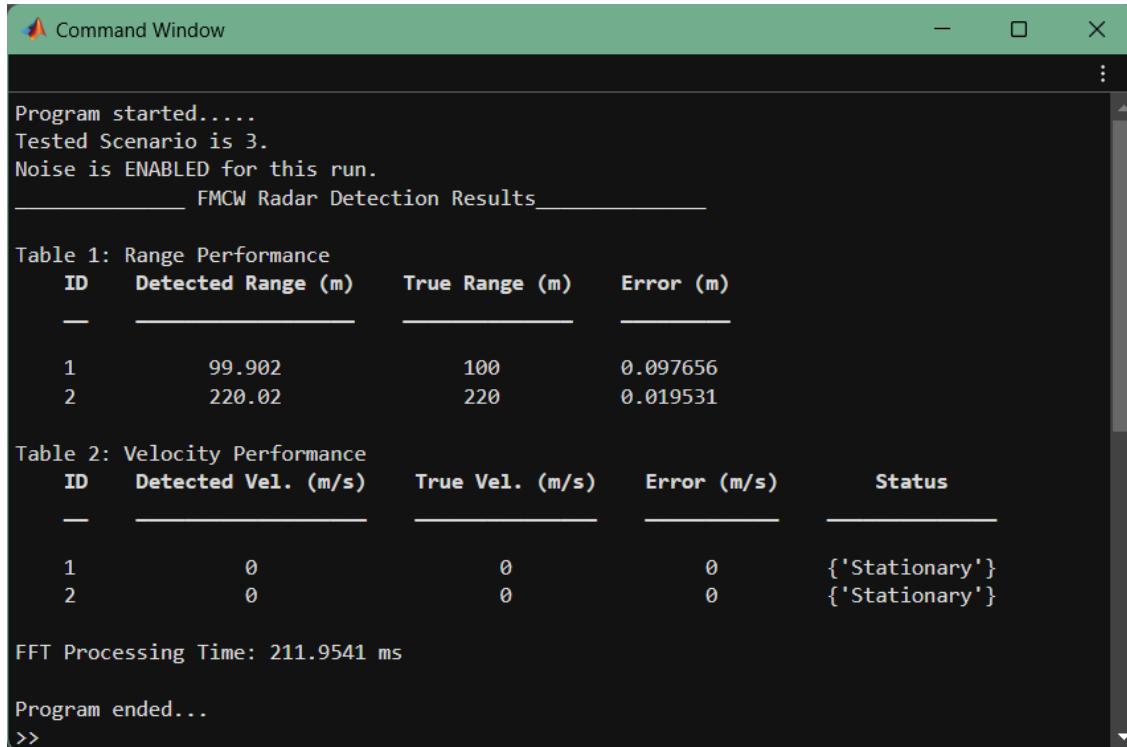
- Figure 21: Range–Velocity Map (dB Scale)

The final 2D heatmap provides a complete spatial and kinematic overview, plotting target distance against speed in a single high-resolution frame.



Phase 4: Numerical Performance Summary

The following tables summarize the accuracy of the radar system by comparing detected values against the ground truth for Scenario 3.



Program started.....
Tested Scenario is 3.
Noise is ENABLED for this run.
FMCW Radar Detection Results

ID	Detected Range (m)	True Range (m)	Error (m)
1	99.902	100	0.097656
2	220.02	220	0.019531

ID	Detected Vel. (m/s)	True Vel. (m/s)	Error (m/s)	Status
1	0	0	0	{'Stationary'}
2	0	0	0	{'Stationary'}

FFT Processing Time: 211.9541 ms

Program ended..

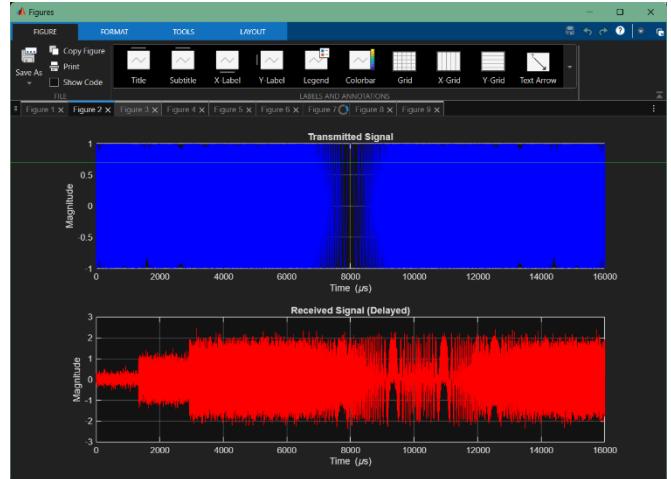
>>

Test Outputs & Plots: Scenario 4

This scenario evaluates Doppler polarity handling by simulating one approaching and one receding target.

Phase 1: Signal Generation & Reception

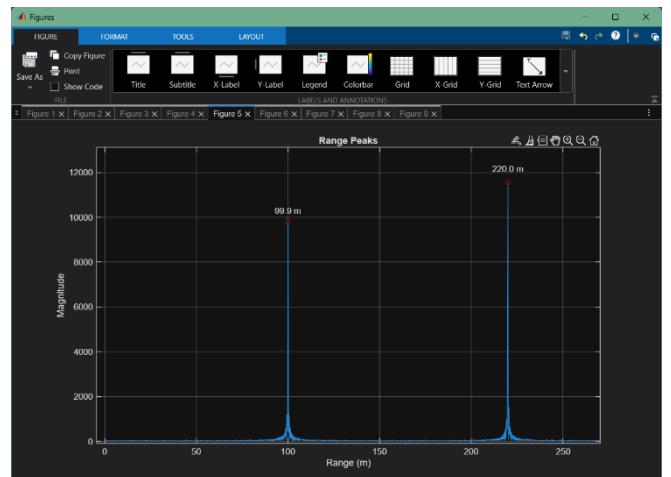
- The transmitted signal is the same as the normal case (scenario 1).
- The received signal got different delay rather than other scenarios.
- Figure 22: Transmitted vs. Received Signal



Phase 2: Range Processing

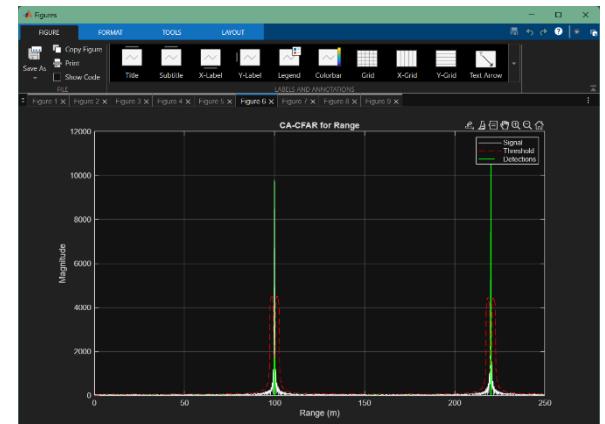
- Figure 23: Range Peaks

The output of the Fast-Time FFT, showing distinct magnitude spikes at the precise distances of the simulated targets.



- Figure 24: CA-CFAR for Range

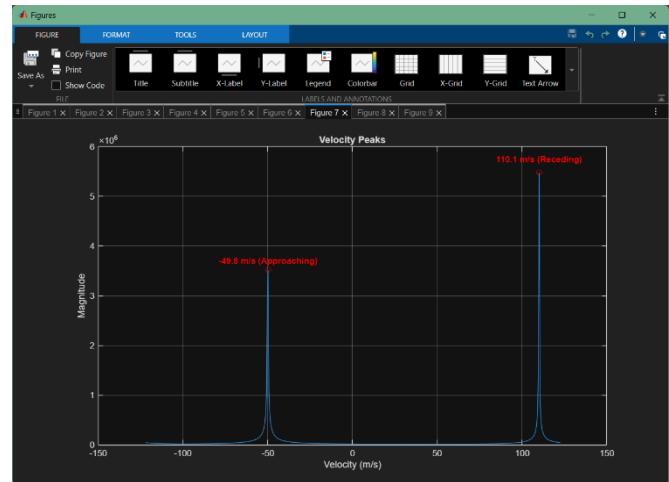
Shows the adaptive threshold (red dashed line) tracking the noise floor to successfully isolate target peaks from background interference.



Phase 3: Doppler Processing & Final Mapping

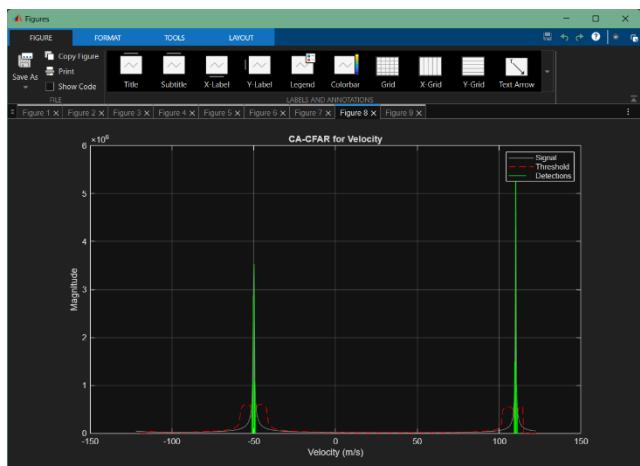
- Figure 25: Velocity Peaks

The output of the Slow-Time FFT, revealing the radial speeds of all detected objects in the field of view.



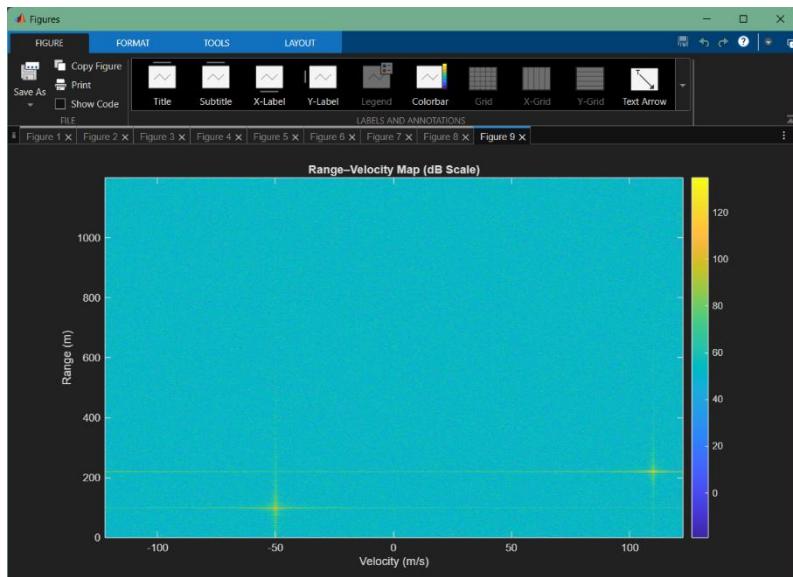
- Figure 26: CA-CFAR for Velocity

Demonstrates the automated detection of motion, applying a dynamic threshold to confirm target velocities while suppressing Doppler sidelobes.



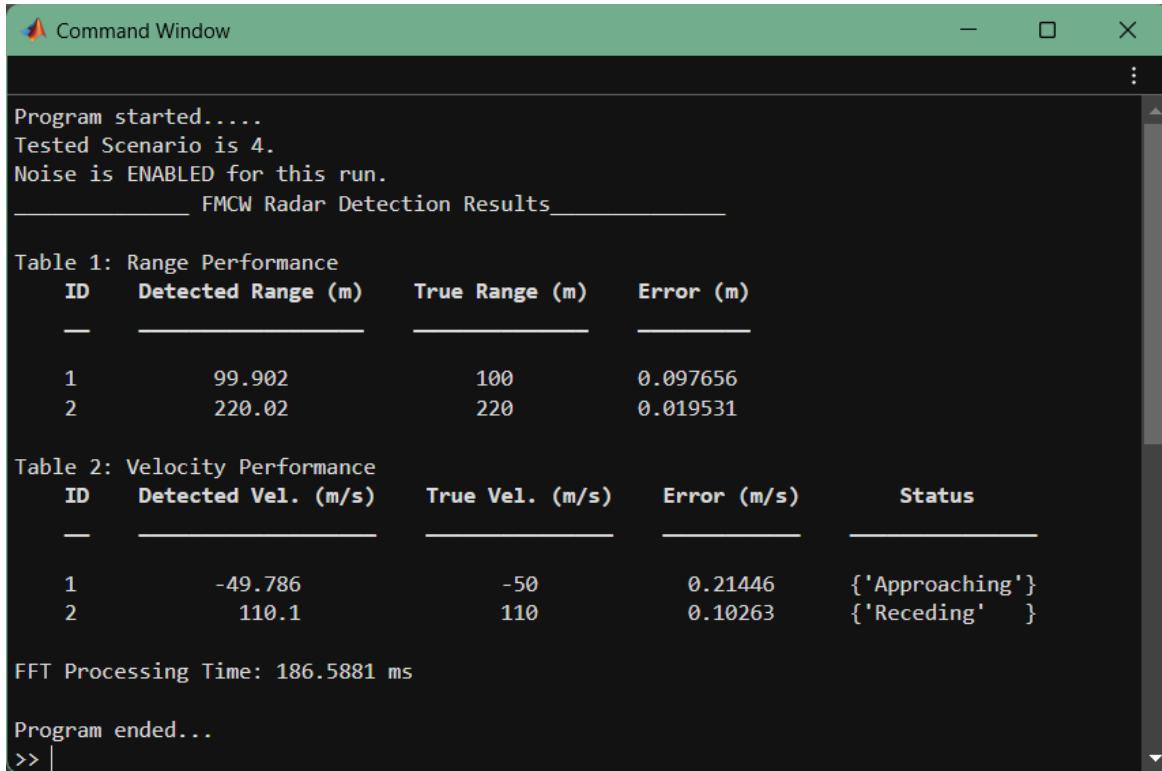
- Figure 27: Range–Velocity Map (dB Scale)

The final 2D heatmap provides a complete spatial and kinematic overview, plotting target distance against speed in a single high-resolution frame.



Phase 4: Numerical Performance Summary

The following tables summarize the accuracy of the radar system by comparing detected values against the ground truth for Scenario 4.



Program started.....
Tested Scenario is 4.
Noise is ENABLED for this run.

_____ FMCW Radar Detection Results _____

Table 1: Range Performance

ID	Detected Range (m)	True Range (m)	Error (m)
1	99.902	100	0.097656
2	220.02	220	0.019531

Table 2: Velocity Performance

ID	Detected Vel. (m/s)	True Vel. (m/s)	Error (m/s)	Status
1	-49.786	-50	0.21446	{'Approaching'}
2	110.1	110	0.10263	{'Receding'}

FFT Processing Time: 186.5881 ms

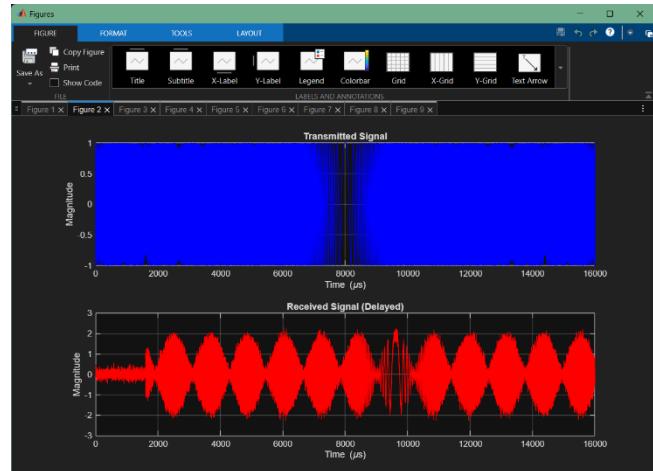
Program ended...
|>>|

Test Outputs & Plots: Scenario 5

This scenario stresses range resolution by placing two targets only 2 m apart.

Phase 1: Signal Generation & Reception

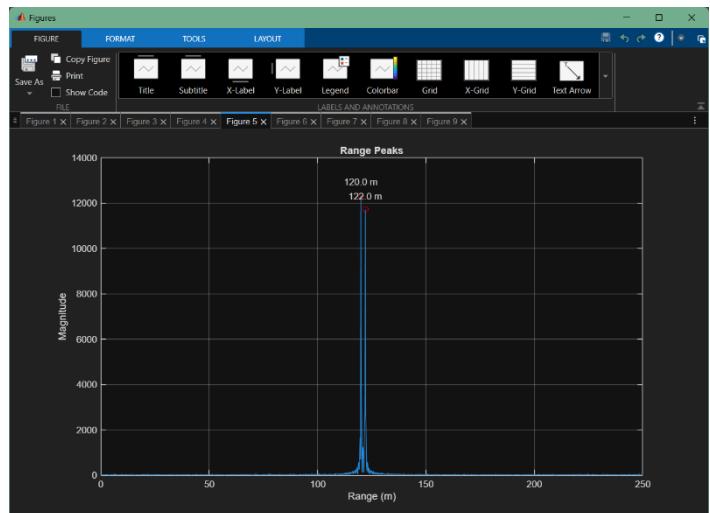
- The transmitted signal is the same as the normal case (scenario 1).
- The received signal got different delay rather than other scenarios.
- Figure 28: Transmitted vs. Received Signal



Phase 2: Range Processing

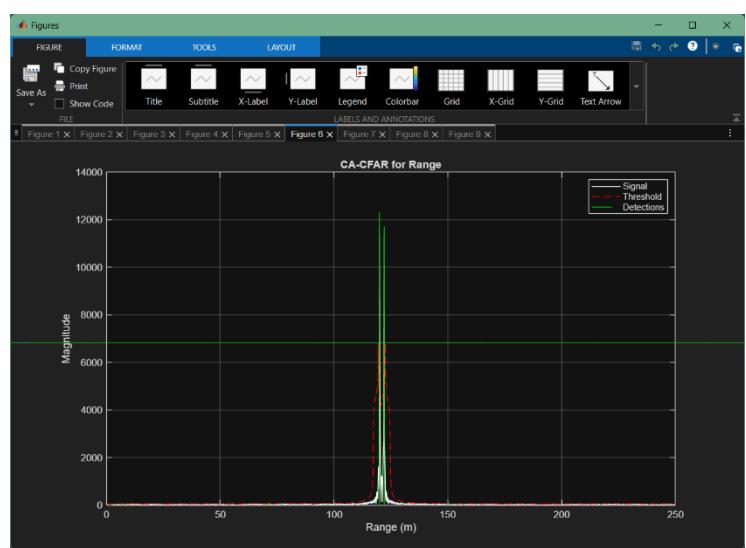
- Figure 29: Range Peaks

The output of the Fast-Time FFT, showing distinct magnitude spikes at the precise distances of the simulated targets.



- Figure 30: CA-CFAR for Range

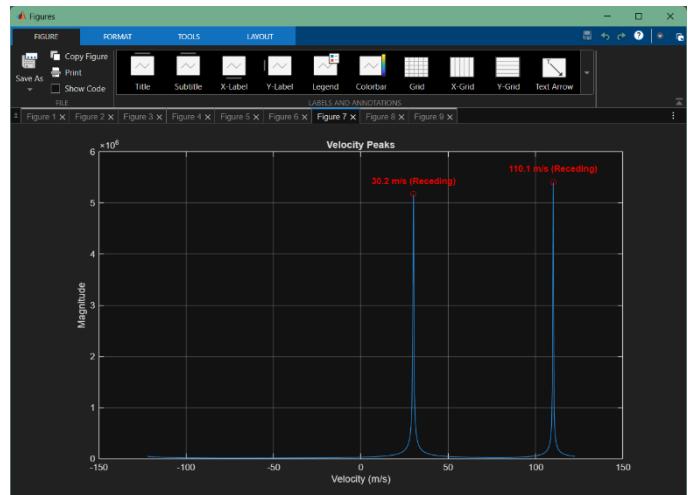
Shows the adaptive threshold (red dashed line) tracking the noise floor to successfully isolate target peaks from background interference.



Phase 3: Doppler Processing & Final Mapping

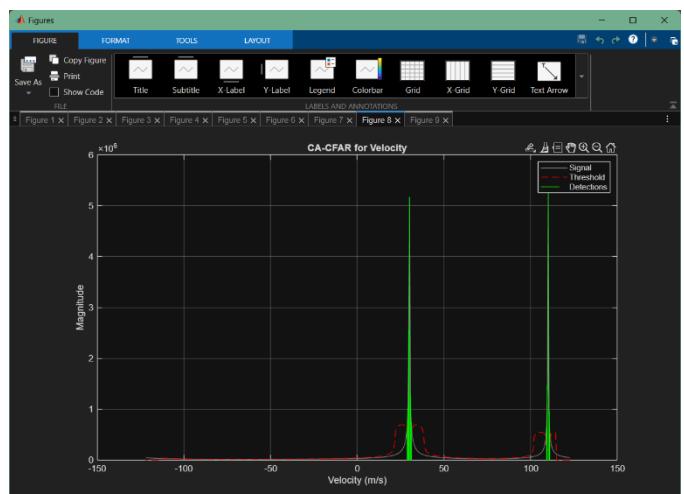
- Figure 31: Velocity Peaks

The output of the Slow-Time FFT, revealing the radial speeds of all detected objects in the field of view.



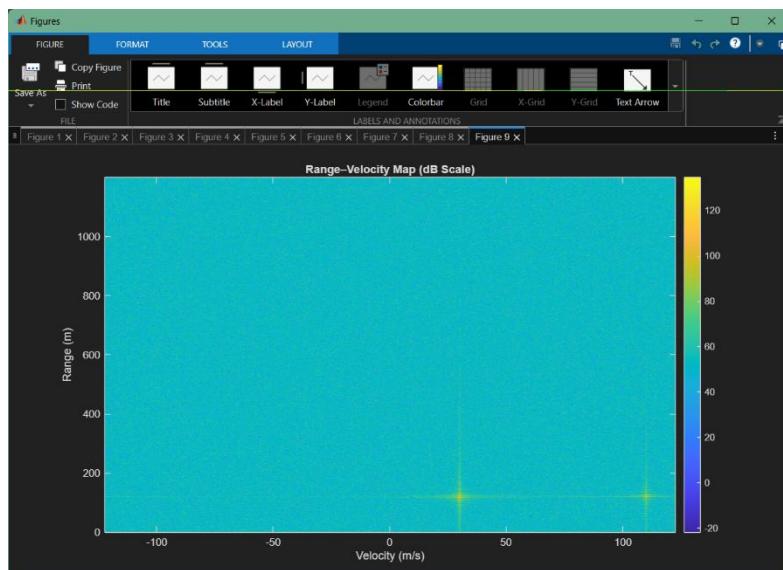
- Figure 32: CA-CFAR for Velocity

Demonstrates the automated detection of motion, applying a dynamic threshold to confirm target velocities while suppressing Doppler sidelobes.



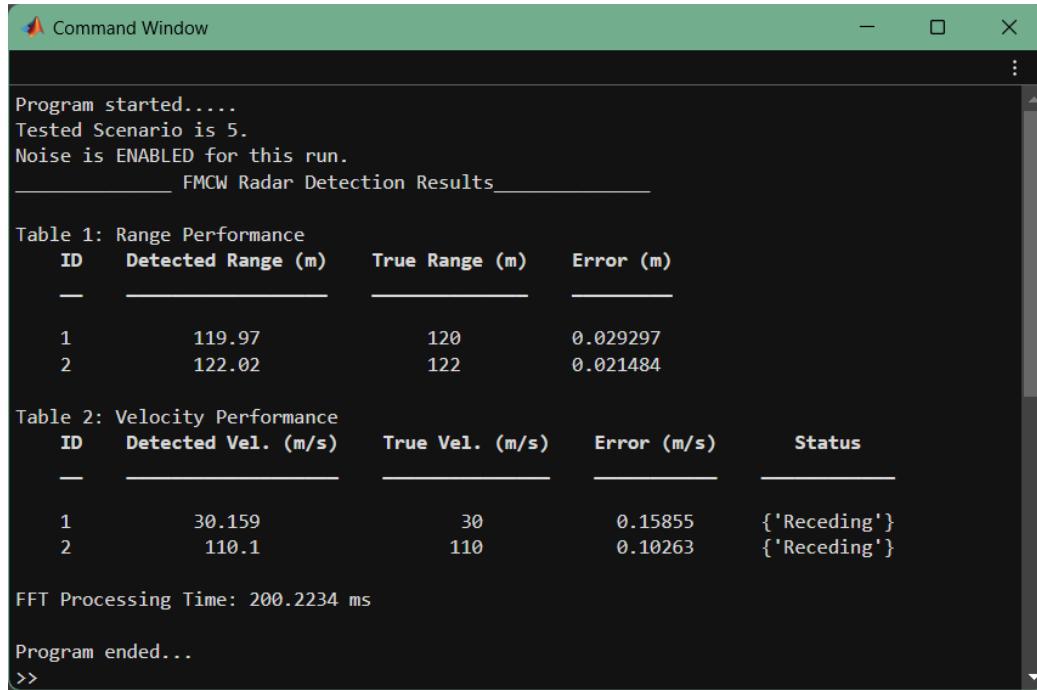
- Figure 33: Range–Velocity Map (dB Scale)

The final 2D heatmap provides a complete spatial and kinematic overview, plotting target distance against speed in a single high-resolution frame.



Phase 4: Numerical Performance Summary

The following tables summarize the accuracy of the radar system by comparing detected values against the ground truth for Scenario 5.



Program started.....
Tested Scenario is 5.
Noise is ENABLED for this run.
_____ FMCW Radar Detection Results _____

Table 1: Range Performance

ID	Detected Range (m)	True Range (m)	Error (m)
1	119.97	120	0.029297
2	122.02	122	0.021484

Table 2: Velocity Performance

ID	Detected Vel. (m/s)	True Vel. (m/s)	Error (m/s)	Status
1	30.159	30	0.15855	{'Receding'}
2	110.1	110	0.10263	{'Receding'}

FFT Processing Time: 200.2234 ms

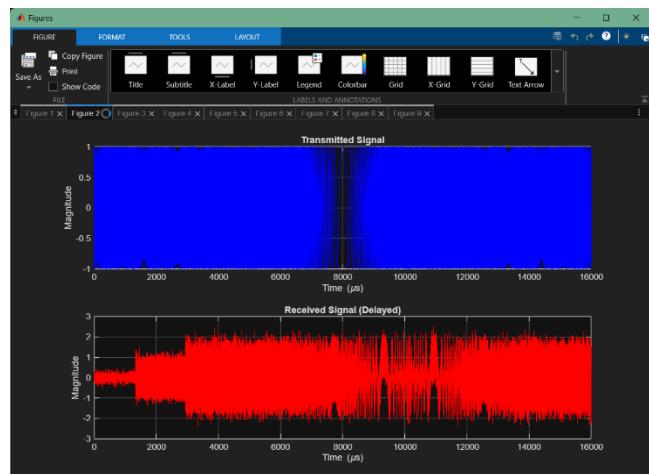
Program ended...
>>

Test Outputs & Plots: Scenario 6

This scenario examines Doppler resolution by introducing two targets with a small velocity difference.

Phase 1: Signal Generation & Reception

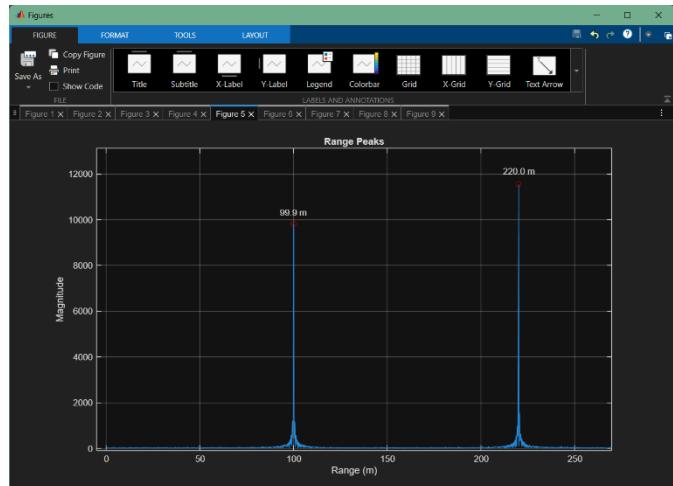
- The transmitted signal is the same as the normal case (scenario 1).
- The received signal got different delay rather than other scenarios.
- Figure 34: Transmitted vs. Received Signal



Phase 2: Range Processing

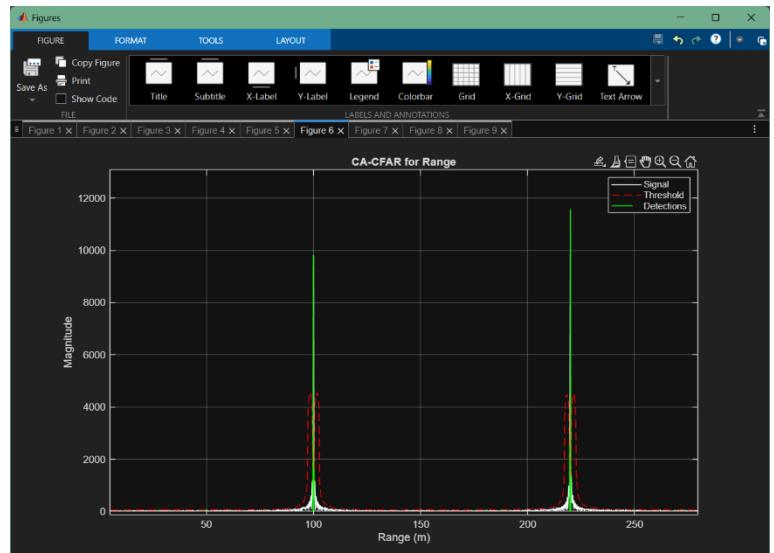
- Figure 35: Range Peaks

The output of the Fast-Time FFT, showing distinct magnitude spikes at the precise distances of the simulated targets.



- Figure 36: CA-CFAR for Range

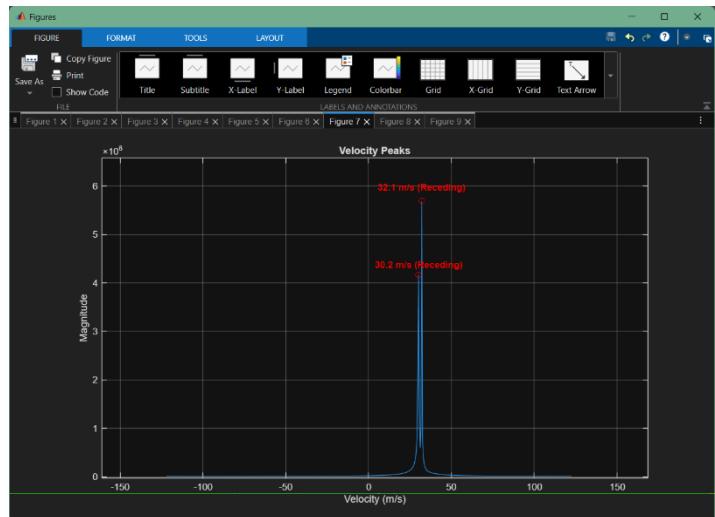
Shows the adaptive threshold (red dashed line) tracking the noise floor to successfully isolate target peaks from background interference.



Phase 3: Doppler Processing & Final Mapping

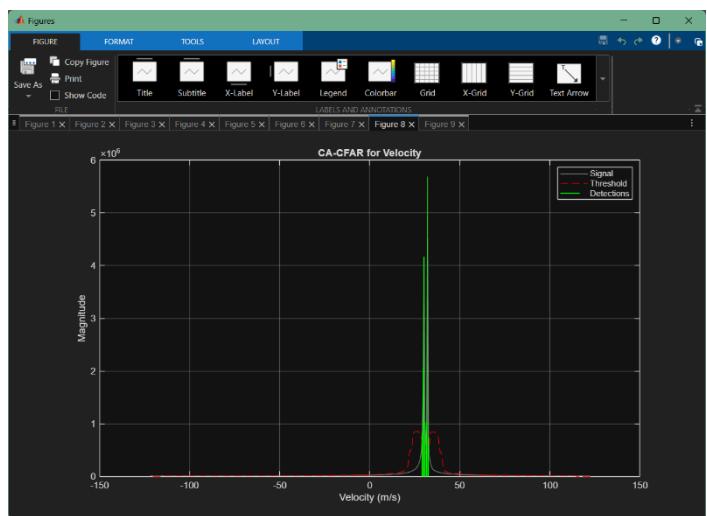
- Figure 37: Velocity Peaks

The output of the Slow-Time FFT, revealing the radial speeds of all detected objects in the field of view.



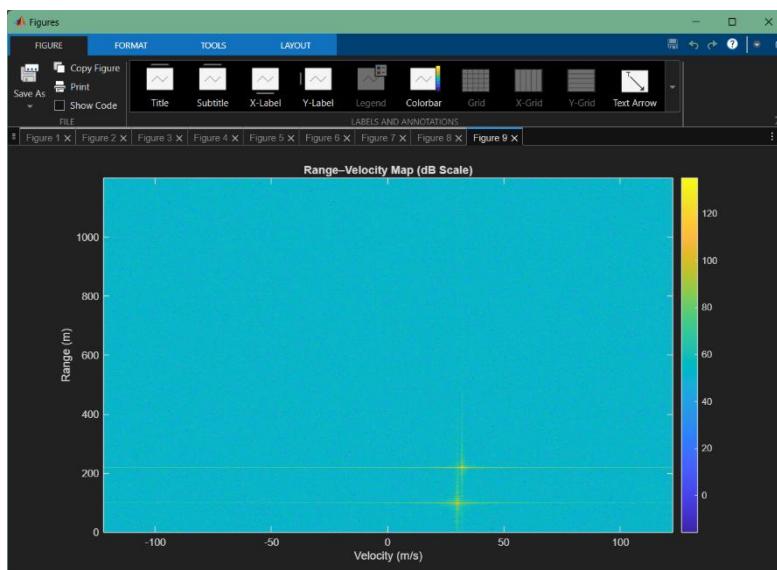
- Figure 38: CA-CFAR for Velocity

Demonstrates the automated detection of motion, applying a dynamic threshold to confirm target velocities while suppressing Doppler sidelobes.



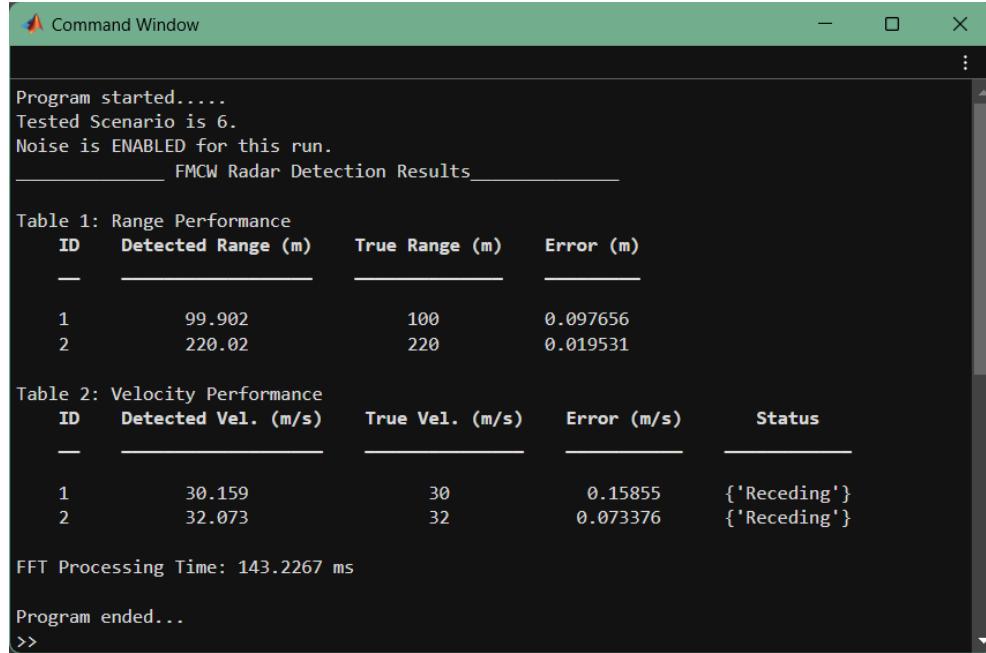
- Figure 39: Range–Velocity Map (dB Scale)

The final 2D heatmap provides a complete spatial and kinematic overview, plotting target distance against speed in a single high-resolution frame.



Phase 4: Numerical Performance Summary

The following tables summarize the accuracy of the radar system by comparing detected values against the ground truth for Scenario 6.



Program started.....
Tested Scenario is 6.
Noise is ENABLED for this run.
FMCW Radar Detection Results

ID	Detected Range (m)	True Range (m)	Error (m)
1	99.902	100	0.097656
2	220.02	220	0.019531

ID	Detected Vel. (m/s)	True Vel. (m/s)	Error (m/s)	Status
1	30.159	30	0.15855	{'Receding'}
2	32.073	32	0.073376	{'Receding'}

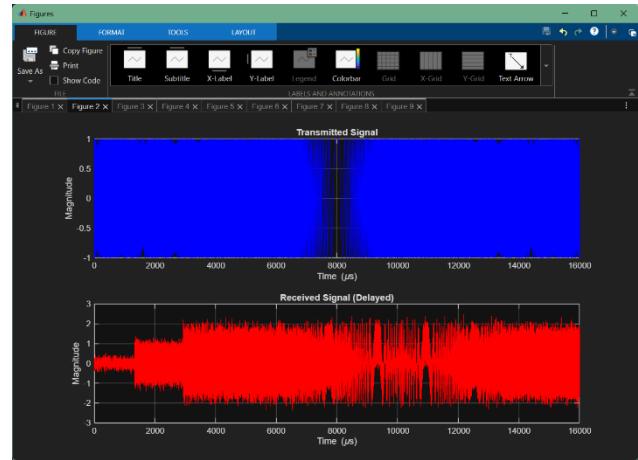
FFT Processing Time: 143.2267 ms
Program ended...
>>

Test Outputs & Plots: Scenario 7

This scenario tests the system's ability to handle large positive and negative target velocities within the unambiguous Doppler region.

Phase 1: Signal Generation & Reception

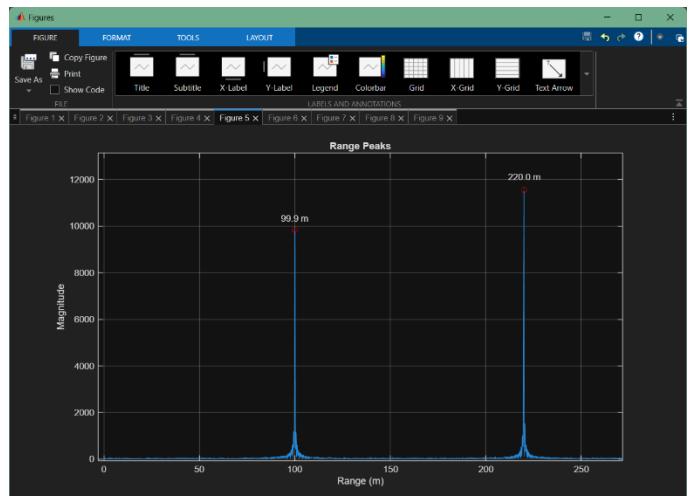
- The transmitted signal is the same as the normal case (scenario 1).
- The received signal got different delay rather than other scenarios.
- Figure 40: Transmitted vs. Received Signal



Phase 2: Range Processing

- Figure 41: Range Peaks

The output of the Fast-Time FFT, showing distinct magnitude spikes at the precise distances of the simulated targets.



- Figure 42: CA-CFAR for Range

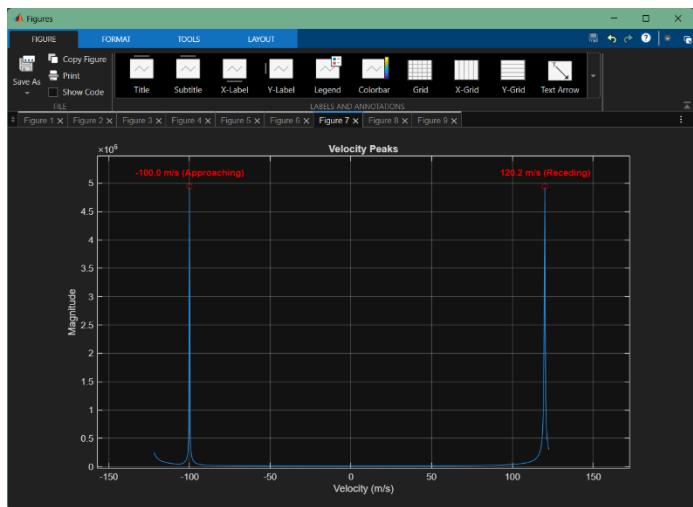
Shows the adaptive threshold (red dashed line) tracking the noise floor to successfully isolate target peaks from background interference.



Phase 3: Doppler Processing & Final Mapping

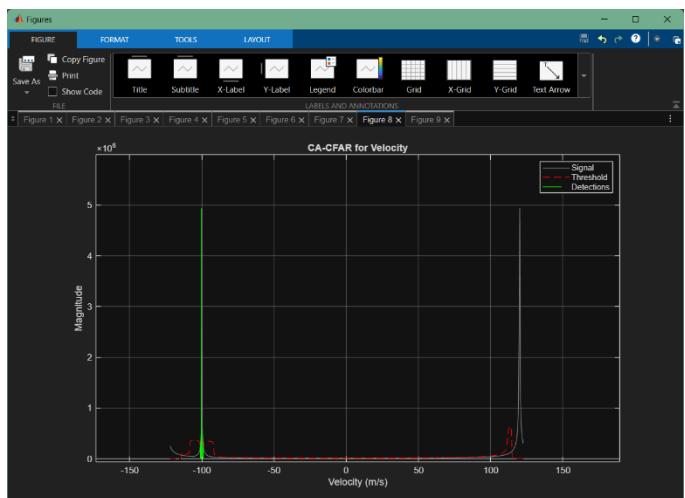
- Figure 43: Velocity Peaks

The output of the Slow-Time FFT, revealing the radial speeds of all detected objects in the field of view.



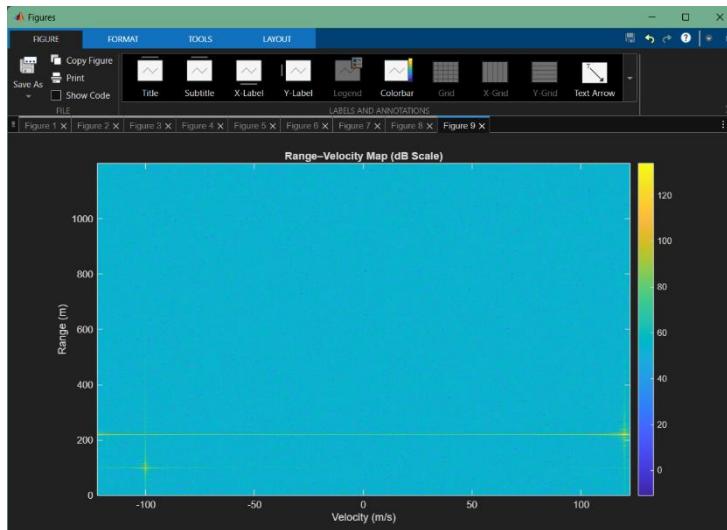
- Figure 44: CA-CFAR for Velocity

Demonstrates the automated detection of motion, applying a dynamic threshold to confirm target velocities while suppressing Doppler sidelobes.



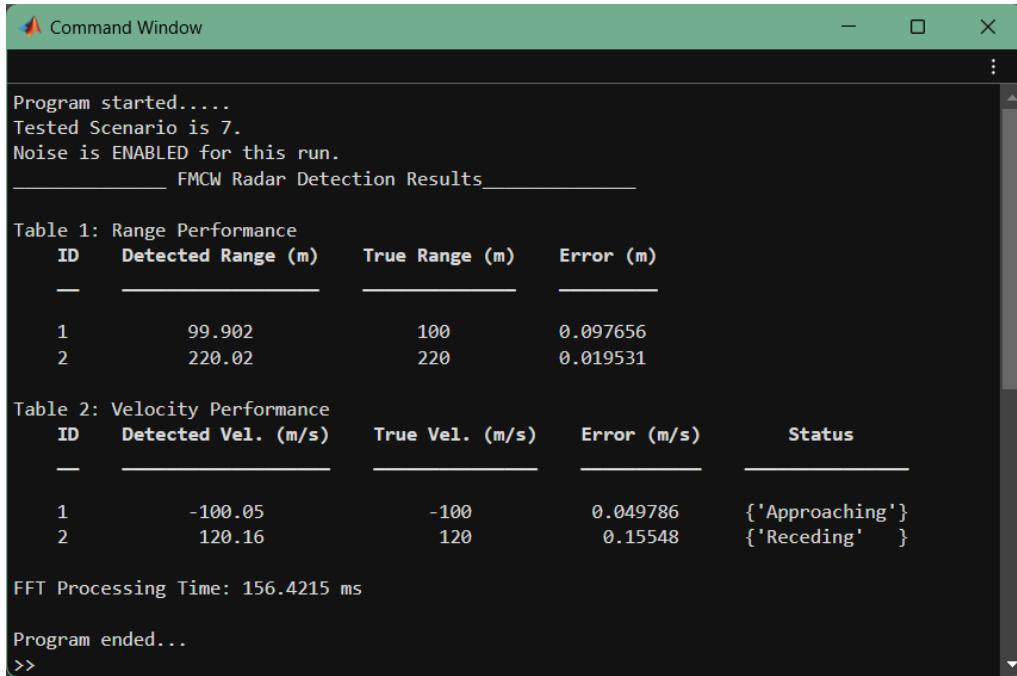
- Figure 45: Range–Velocity Map (dB Scale)

The final 2D heatmap provides a complete spatial and kinematic overview, plotting target distance against speed in a single high-resolution frame.



Phase 4: Numerical Performance Summary

The following tables summarize the accuracy of the radar system by comparing detected values against the ground truth for Scenario 7.



Program started.....
Tested Scenario is 7.
Noise is ENABLED for this run.
FMCW Radar Detection Results

ID	Detected Range (m)	True Range (m)	Error (m)
1	99.902	100	0.097656
2	220.02	220	0.019531

ID	Detected Vel. (m/s)	True Vel. (m/s)	Error (m/s)	Status
1	-100.05	-100	0.049786	{'Approaching'}
2	120.16	120	0.15548	{'Receding'}

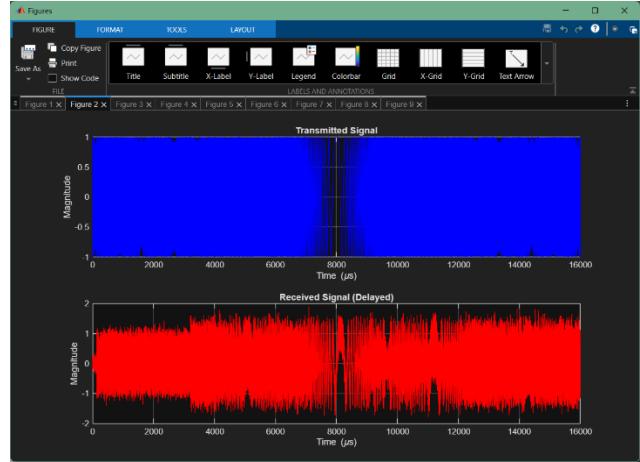
FFT Processing Time: 156.4215 ms
Program ended...
>>

Test Outputs & Plots: Scenario 8

This scenario evaluates performance near the radar's maximum unambiguous range.

Phase 1: Signal Generation & Reception

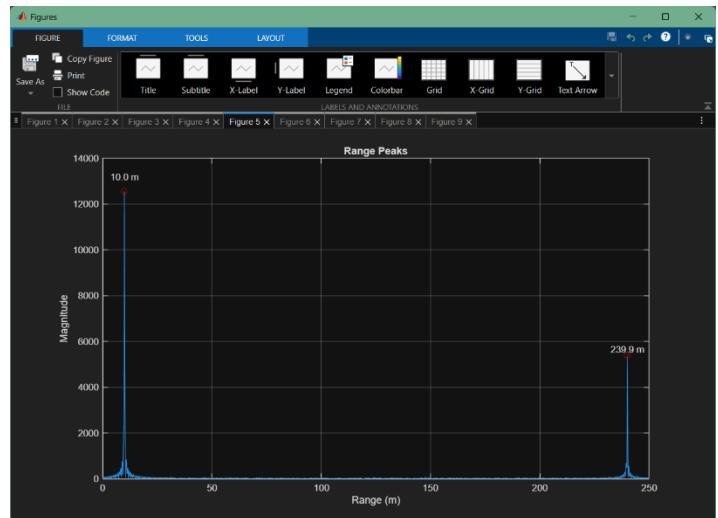
- The transmitted signal is the same as the normal case (scenario 1).
- The received signal got different delay rather than other scenarios.
- Figure 46: Transmitted vs. Received Signal



Phase 2: Range Processing

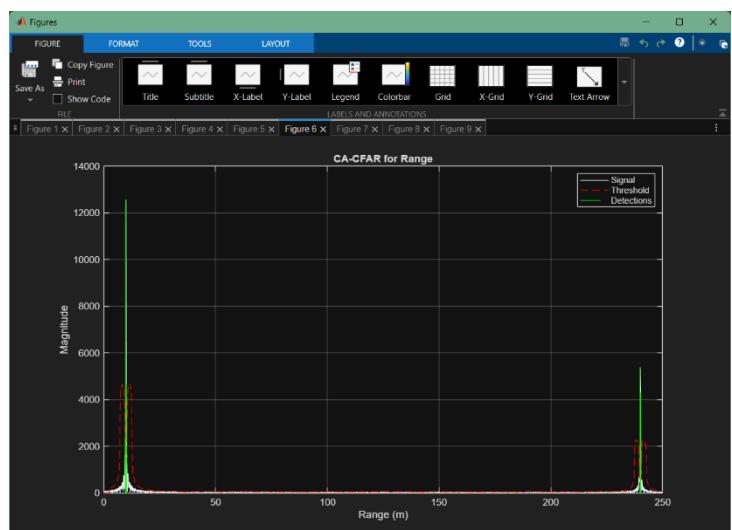
- Figure 47: Range Peaks

The output of the Fast-Time FFT, showing distinct magnitude spikes at the precise distances of the simulated targets.



- Figure 48: CA-CFAR for Range

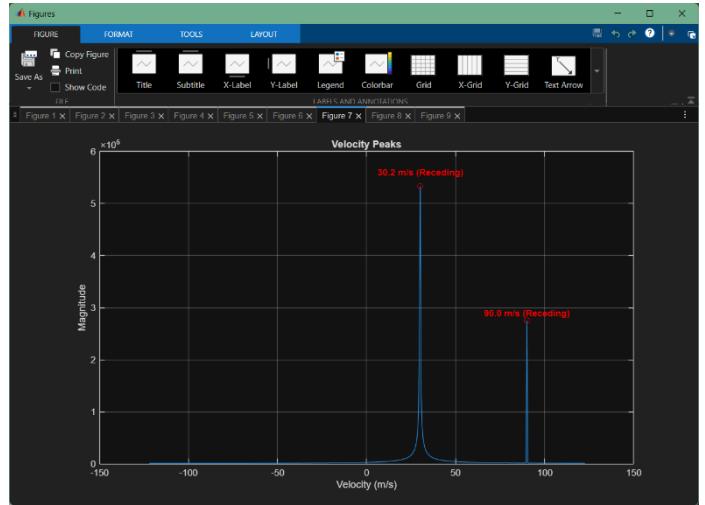
Shows the adaptive threshold (red dashed line) tracking the noise floor to successfully isolate target peaks from background interference.



Phase 3: Doppler Processing & Final Mapping

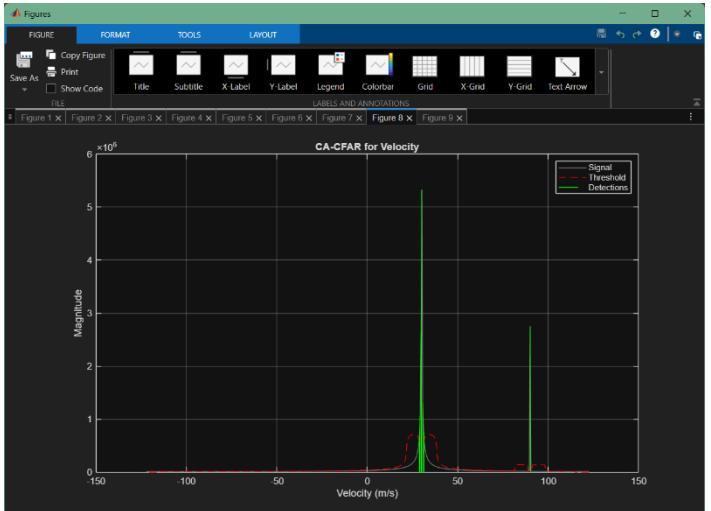
- Figure 49: Velocity Peaks

The output of the Slow-Time FFT, revealing the radial speeds of all detected objects in the field of view.



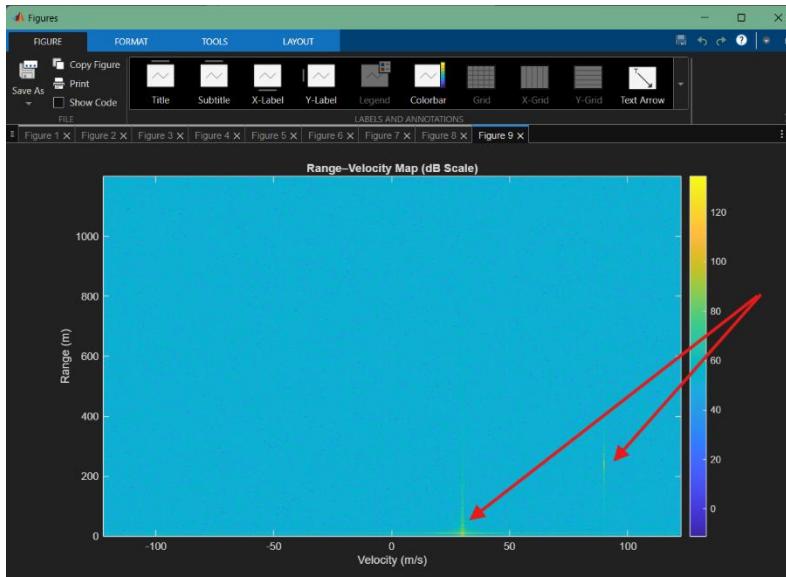
- Figure 50: CA-CFAR for Velocity

Demonstrates the automated detection of motion, applying a dynamic threshold to confirm target velocities while suppressing Doppler sidelobes.



- Figure 51: Range–Velocity Map (dB Scale)

The final 2D heatmap provides a complete spatial and kinematic overview, plotting target distance against speed in a single high-resolution frame.



Phase 4: Numerical Performance Summary

The following tables summarize the accuracy of the radar system by comparing detected values against the ground truth for Scenario 8.

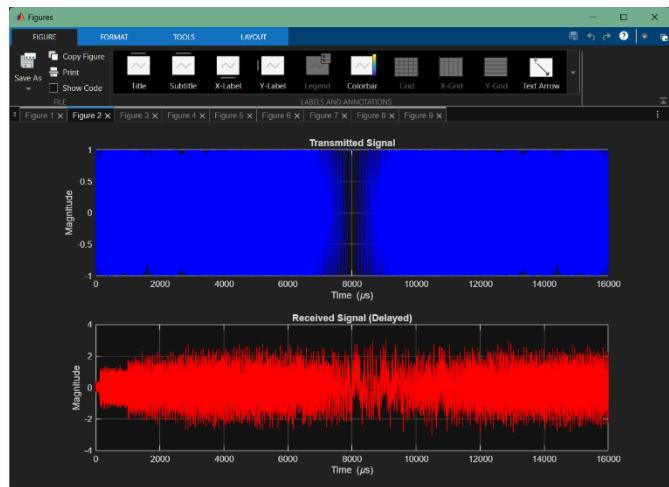
```
Command Window
Program started.....
Tested Scenario is 8.
Noise is ENABLED for this run.
_____
FMCW Radar Detection Results
_____
Table 1: Range Performance
ID    Detected Range (m)    True Range (m)    Error (m)
_____
1        9.9609            10            0.039063
2       239.94            240            0.058594
_____
Table 2: Velocity Performance
ID    Detected Vel. (m/s)    True Vel. (m/s)    Error (m/s)    Status
_____
1        30.159             30            0.15855      {'Receding'}
2       89.997              90            0.0030637   {'Receding'}
_____
FFT Processing Time: 177.4227 ms
_____
Program ended...
>> |
```

Test Outputs & Plots: Scenario 9

This scenario evaluates system performance in a dense multi-target environment with varying ranges, velocities, and amplitudes.

Phase 1: Signal Generation & Reception

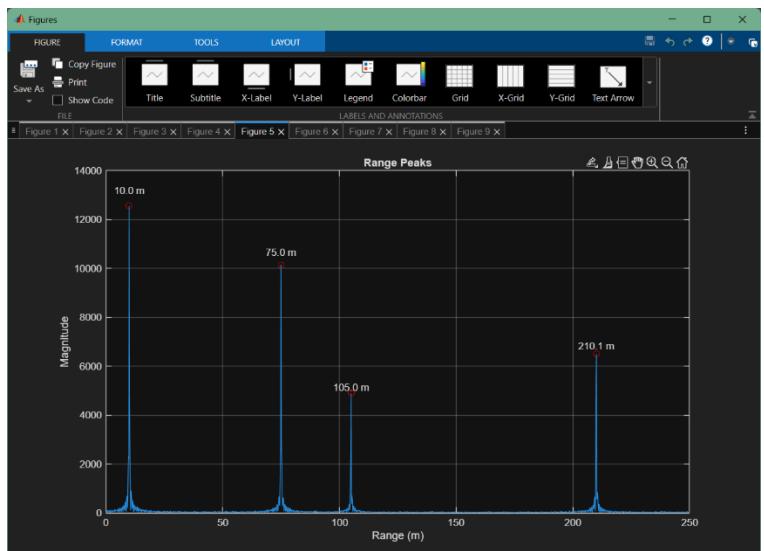
- The transmitted signal is the same as the normal case (scenario 1).
- The received signal got different delay rather than other scenarios.
- Figure 52: Transmitted vs. Received Signal



Phase 2: Range Processing

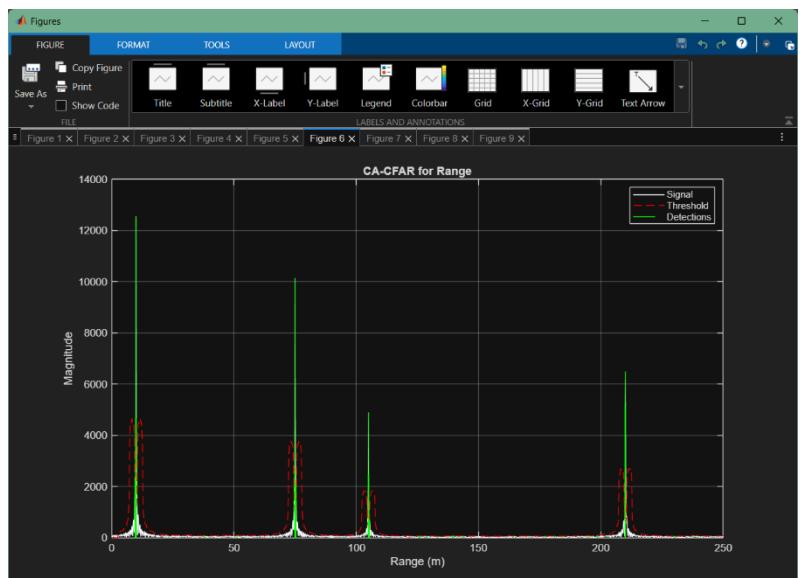
- Figure 53: Range Peaks

The output of the Fast-Time FFT, showing distinct magnitude spikes at the precise distances of the simulated targets.



- Figure 54: CA-CFAR for Range

Shows the adaptive threshold (red dashed line) tracking the noise floor to successfully isolate target peaks from background interference.



Phase 3: Doppler Processing & Final Mapping

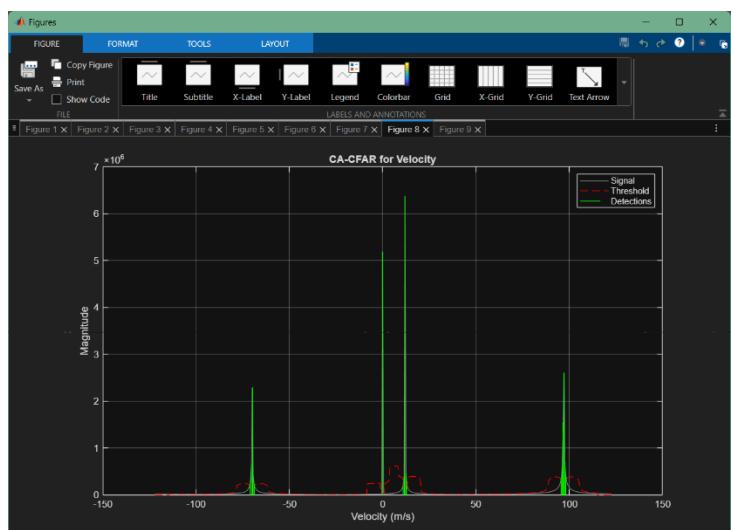
- Figure 55: Velocity Peaks

The output of the Slow-Time FFT, revealing the radial speeds of all detected objects in the field of view.



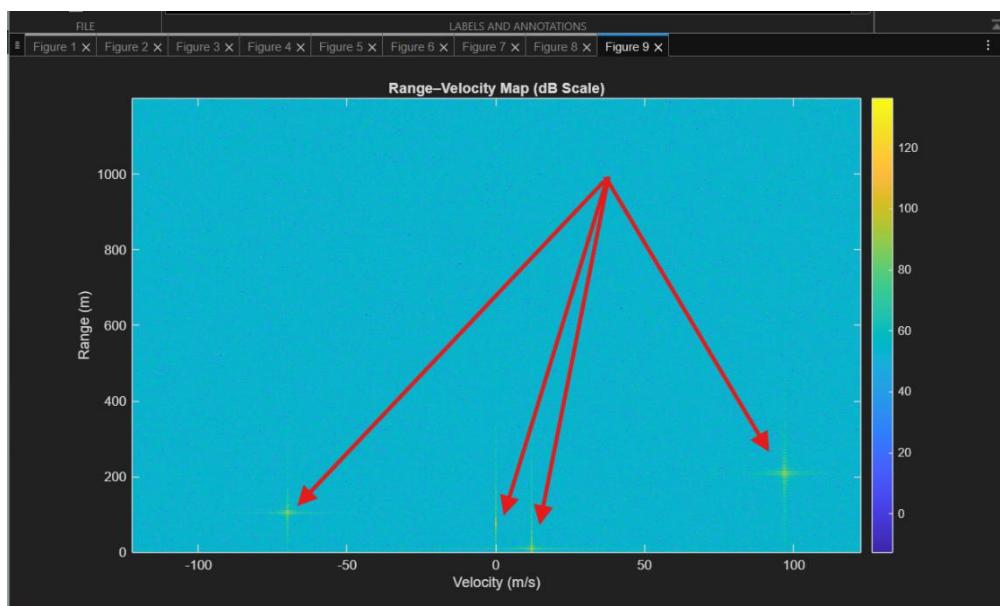
- Figure 56: CA-CFAR for Velocity

Demonstrates the automated detection of motion, applying a dynamic threshold to confirm target velocities while suppressing Doppler sidelobes.



- Figure 57: Range–Velocity Map (dB Scale)

The final 2D heatmap provides a complete spatial and kinematic overview, plotting target distance against speed in a single high-resolution frame.



Phase 4: Numerical Performance Summary

The following tables summarize the accuracy of the radar system by comparing detected values against the ground truth for Scenario 9.

The image shows a MATLAB Command Window with the following content:

Table 1: Range Performance

ID	Detected Range (m)	True Range (m)	Error (m)
1	9.9609	10	0.039063
2	105.03	105	0.029297
3	75	75	1.4211e-14
4	210.06	210	0.058594

Table 2: Velocity Performance

ID	Detected Vel. (m/s)	True Vel. (m/s)	Error (m/s)	Status
1	11.968	12	0.032322	{'Receding'}
2	-69.891	-70	0.10876	{'Approaching'}
3	0	0	0	{'Stationary'}
4	97.178	97	0.17754	{'Receding'}

FFT Processing Time: 178.5571 ms

Program ended...

>>

Test Outputs & Plots: Scenario 10

This scenario compares the robustness of the FMCW radar system under degraded signal conditions by reducing the signal-to-noise ratio to 10 dB while keeping the same target ranges and velocities as the normal case. This scenario is the same as scenarios 1 and 2 but with different SNR, so all figures are almost the same as scenarios 1 and 2. The comparison is better to be only for the range-velocity plot.

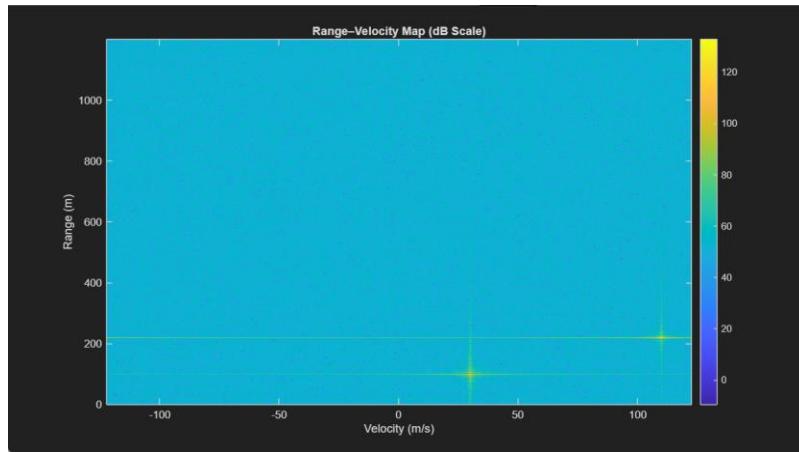


Figure 9. Range-Velocity Map with SNR = 15dB (Scenario 1)

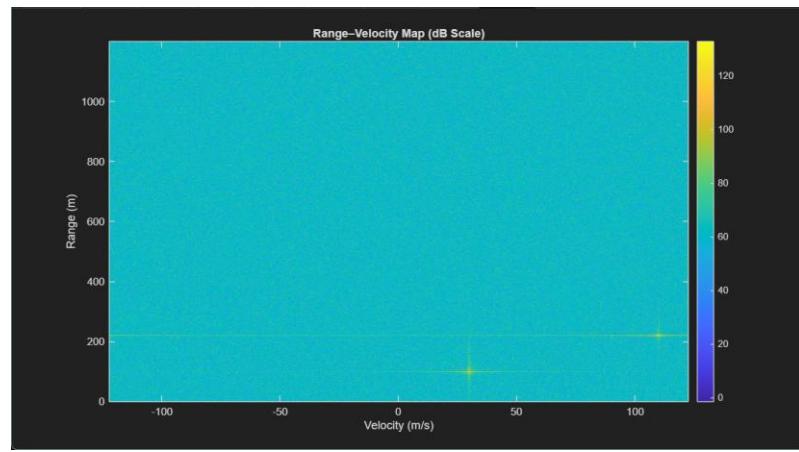


Figure 15. Range-Velocity Map with SNR = 5dB (Scenario 2)

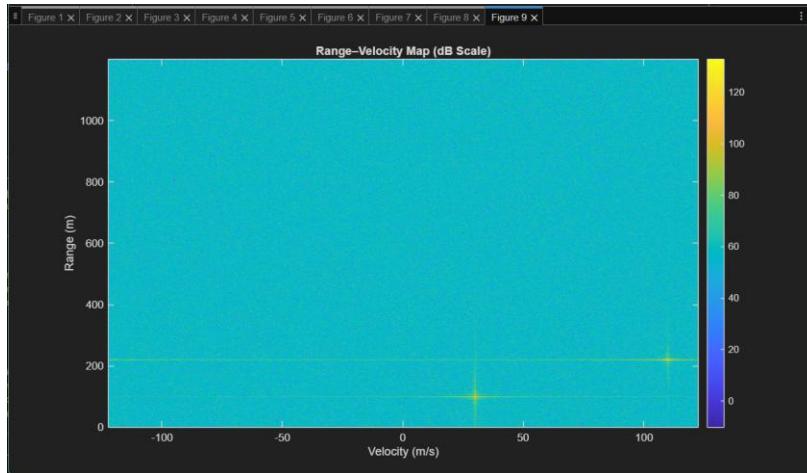


Figure 58. Range-Velocity Map with SNR = 10dB (Scenario 10)

Design Limitations

Implementing an FMCW radar system effectively uncovers a few more physical and mathematical restrictions than its ideal and entirely simulated condition. The first and most important restriction would be a **Range-Doppler Coupled system** in which the frequency measured at the receiver is a summation of the range of the target and its velocity.

The beat frequency formula is given as $f_b = f_R + f_D$: with $f_R = \frac{S \cdot 2R}{c}$ as the range frequency and $f_D = \frac{2v}{\lambda}$ as the Doppler shift. In high-speed scenarios, a target's perceived range can be altered by the Doppler component. For example, when the target's movement reaches about 100 km/h, the deviation may be sufficient to require a dual-slope chirp (triangular modulation) for decoupling these parameters.

Another equally important limitation is that of maximum unambiguous ranges and Velocities. The maximum reach determined by the sampling frequency f_s is given by the relationship:

- **Maximum unambiguous range** $R_{max} = \frac{f_s \cdot c}{2S}$. Given the simulation parameters $f_s = 2$ GHz and $S = 1.25 \times 10^{14}$ Hz/s:

$$R_{max} = \frac{2 \times 10^9 \cdot 3 \times 10^8}{2 \cdot 1.25 \times 10^{14}} = 2400 \text{ m}$$

Beyond this point, the beat frequency of a target will exceed the Nyquist limit and it will be perceived through aliasing. The same is true for maximum observable speed, which is subject to Pulse Repetition Interval (PRI) regulations:

- **Maximum unambiguous Velocity** $v_{max} = \frac{\lambda}{4 \cdot PRI}$. With $\lambda \approx 3.92$ mm (at 76.5 GHz) and $PRI = 8 \mu\text{s}$:

$$v_{max} = \frac{0.00392}{4 \cdot 8 \times 10^{-6}} \approx 122.5 \text{ m/s}$$

This simulation had $PRI = 8 \mu\text{s}$; hence, if a target moves faster than that corresponding to v_{max} , the phase shift across chirps exceeds $\pm\pi$, leading to an incorrect velocity reading.

Spectral Leakage and Resolution limitations are also pertinent to the system. Range resolution fundamentally links to bandwidth BW by $\Delta R = \frac{c}{2 \cdot BW}$. Even at zero noise levels, targets separated by less than 15 cm (for $BW = 1$ GHz) cannot be distinguished as separate peaks. The shadowing effects introduced by use of windowing functions to reduce sidelobes--necessary since a very strong target could mask a nearby weak one--lead to further broadening of the main spectral peak, thus practically degrading resolution.

The last assumption made in the simulation is a **Linear Frequency Ramp**. Real hardware would often involve non-linearities in the slope S of the Voltage Controlled Oscillator (VCO). If the slope deviates by just a small percentage, the beat frequency can be nothing more than a pure sine wave, which spreads the energy through multiple range bins and results in reduced signal-to-noise ratio (SNR) and an increased likelihood of false alarms.

Conclusion

This project involved designing and implementing a full FMCW radar signal processing pipeline, which has resulted in converting raw time-domain signals into action-specific spatial and kinematic data. The entire setup performed fairly accurately by keeping range and velocity errors limited within theoretical boundaries while incorporating the CA-CFAR algorithm to provide strong target isolation from the noise floor.

The results show a massively efficient architecture that can even process data within a millisecond. This becomes highly viable for real-time applications such as automotive safety or industrial monitoring. In fact, the simulation verifies that an effective mapping into a Range-Doppler grid, and automated direction classification indeed confirms that the chosen radar configuration would make for a very good trade-off between resolution and depth of detection.

References

- “Radar Tutorial” site, contains all basics of designing a radar, FMCW Radars as well:
<https://www.radartutorial.eu/02.basics/Frequency%20Modulated%20Continuous%20Wave%20Radar.en.html>
- The Scientist and Engineer's Guide to Digital Signal Processing, **Chapter 31: The Complex Fourier Transform:** <https://www.dspsguide.com/ch31.htm>
- Texas Instrument FMCW Radar Design Series: <https://www.ti.com/video/series/mmwave-training-series.html>
- Math Work documentations (MatLab Site):
https://www.mathworks.com/help/radar/ug/automotive-adaptive-cruise-control-using-fmcw-technology.html?s_eid=PSM_15028

Appendix

The FMCW-Radar main Code and the radar configuration file (for different scenarios) can be accessed directly from the [GitHub repository](#):

https://github.com/mhmd-sameer309/FMWC_Radar.git