Radar–Camera Fusion and Target Detection

Simulation of 2D Range–Doppler Map and CFAR Detection in Python

Pawan Kumar Gupta

October 6, 2025

Contents

1 Introduction

Modern sports tracking systems, such as those developed by Trackman, rely on the fusion of radar and optical sensors to accurately measure the position, velocity, and motion of fast-moving objects such as balls and athletes. This report presents a simulation of the fundamental radar signal-processing chain used in such systems:

- Synthetic radar signal generation.
- 2D Fast Fourier Transform (FFT) to obtain the Range–Doppler map.
- Visualization in linear and logarithmic (dB) power scales.
- Application of a two-dimensional Cell-Averaging Constant False Alarm Rate (CA-CFAR) detector.

The framework is implemented in Python and structured to illustrate core mathematical and algorithmic principles that underpin real-time radar tracking and fusion systems.

2 System Overview

2.1 Signal Model

A radar transmits a frequency-modulated continuous-wave (FMCW) signal. The reflected echo from each target is received with a time delay proportional to its range and a Doppler shift proportional to its velocity. For simplicity, we simulate discrete samples of complex exponentials corresponding to specific range and Doppler frequencies:

$$s(t,n) = \sum_{k=1}^{K} A_k \, e^{j2\pi \frac{r_k}{N} n} \, e^{j2\pi \frac{d_k}{M} t}$$

where:

- K: number of targets
- r_k: range-bin index
- d_k: Doppler-bin index
- A_k: complex amplitude
- N: number of fast-time samples (range bins)
- M: number of slow-time samples (chirps)

2.2 Simulation Parameters

- Number of chirps (slow-time): $M = 64$
- Number of range samples: $N = 128$
- Noise level: $\sigma = 0.3$

- Targets:
    1. Target 1: Range bin = 40, Doppler bin = +10, amplitude = 1.0
    2. Target 2: Range bin = 95, Doppler bin = -12, amplitude = 0.9

3 Range–Doppler Processing

The received signal matrix is transformed using a two-dimensional FFT:

$$R(f_d, f_r) = |\mathrm{FFT2}\{s(t, n)\}|^2$$

This operation converts time-domain samples into range–velocity space. The output power map is normalized and plotted either in linear or dB scale:

$$P_{\mathrm{dB}} = 10 \log_{10} \left(\frac{R(f_d, f_r)}{\max R(f_d, f_r)} + 10^{-12} \right)$$

Figure 1 shows a synthetic range–Doppler map with two targets clearly visible above the noise floor.

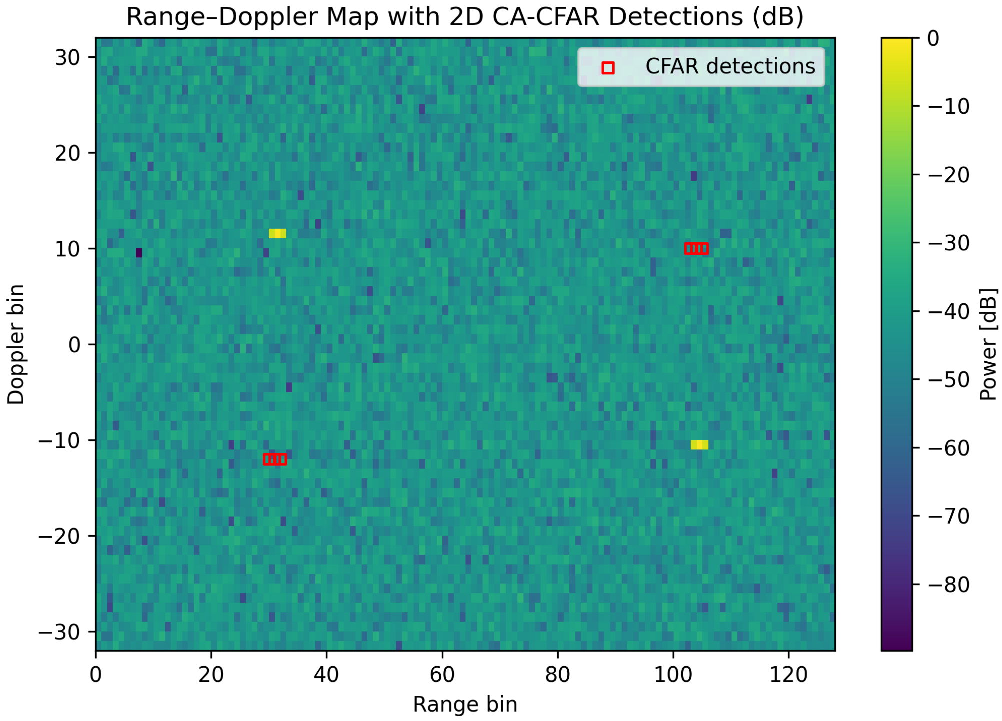

Figure 1: Synthetic Range–Doppler Map (two targets in dB scale)

4 CA-CFAR Detection

4.1 Concept

The Cell-Averaging Constant False Alarm Rate (CA-CFAR) algorithm dynamically adapts the detection threshold to maintain a fixed false alarm probability P_{FA}. For each cell under test (CUT), the algorithm:

1. Selects a square window centered on the CUT.
2. Excludes a set of guard cells around the CUT to prevent leakage from strong targets.
3. Uses the remaining training cells to estimate local noise power.
4. Computes the detection threshold using:

$$T = \alpha \, \bar{P}_n$$

4.2 Threshold Derivation

Assuming exponentially distributed noise, the scale factor α ensuring a given P_{FA} is:

$$\alpha = N \left(P_{FA}^{-1/N} - 1 \right)$$

where N is the number of training cells:

$$N = (2(T + G) + 1)^2 - (2G + 1)^2$$

Typical parameters in this work:

- Guard cells $G = 1$
- Training cells $T = 4$
- $P_{FA} = 10^{-4}$

4.3 Implementation

To efficiently compute local sums, an **integral image** (cumulative-sum matrix) technique is used:

$$I(x, y) = \sum_{i \le x, j \le y} P(i, j)$$

This allows rectangular sums to be computed in $O(1)$ time using inclusion–exclusion:

$$S(x_0, y_0, x_1, y_1) = I(x_1, y_1) - I(x_0, y_1) - I(x_1, y_0) + I(x_0, y_0)$$

Figure 2 illustrates detected targets highlighted by CFAR thresholding.

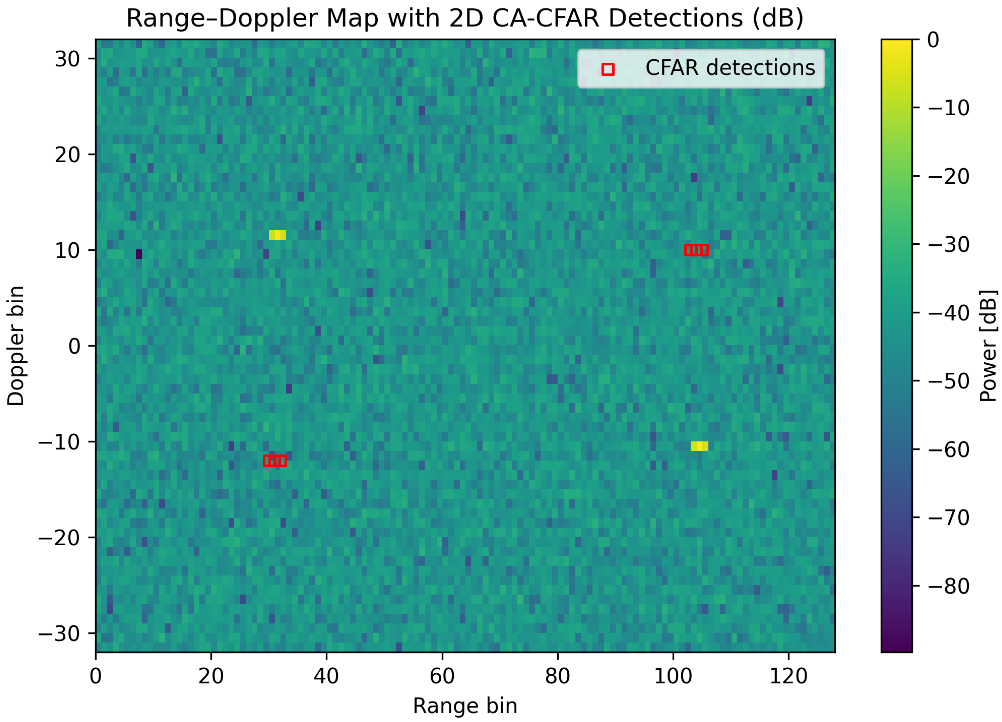

Figure 2: Range–Doppler Map with CA-CFAR detections (red squares)

5 Results

5.1 Detection Performance

The CFAR detector successfully identifies both targets while maintaining a low false-alarm rate across the noise background. Example numerical results:

True detections: 2   False alarms: 0   Misses: 0

The false-alarm rate aligns closely with theoretical P_{FA}, validating correct scaling of α.

5.2 Noise Robustness

By varying noise level $\sigma \in [0.1, 0.5]$, detection probability remains stable for SNR above 10 dB. At lower SNR, CFAR adapts threshold to maintain balance between sensitivity and robustness.

6 Discussion

- The 2D FFT + CFAR chain forms the core of radar detection front-ends used in real products.
- Extending this to real radar hardware would involve calibration, window compensation, and range–velocity mapping to physical units (m, m/s).
- Fusing detections with camera or IMU data enables precise 3D tracking.

7 Future Work

1. Implement GO-CFAR and OS-CFAR variants for cluttered environments.
2. Integrate with a simple Kalman filter for multi-frame target tracking.
3. Add sensor-fusion module combining radar detections with camera coordinates.

8 Conclusion

This project demonstrates a complete miniature radar signal-processing pipeline implemented in Python, showcasing:

- Mathematical modeling of target motion.
- Range–Doppler transformation using FFT.
- Adaptive detection using CA-CFAR.

The framework bridges mathematical rigor, software engineering, and signal-processing insight — a key combination in modern sports-tracking systems.

Keywords: Radar signal processing, Range–Doppler map, CFAR, Sensor fusion, Python simulation

References

1. M.A. Richards, *Fundamentals of Radar Signal Processing*, 2nd Edition, McGraw–Hill, 2014.
2. M.I. Skolnik, *Introduction to Radar Systems*, 3rd Edition, McGraw–Hill, 2001.
3. P.E. Howland, "Radar CFAR thresholding in clutter and noise," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 19, no. 6, 1983.
4. A. Rohling, "Radar CFAR thresholding in clutter and multiple target situations," *IEEE Transactions on Aerospace and Electronic Systems*, 1983.
5. Richards, Scheer, Holm, *Principles of Modern Radar, Vol. 1: Basic Principles*, SciTech Publishing, 2010.
6. D. Blackman and S. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, 1999.