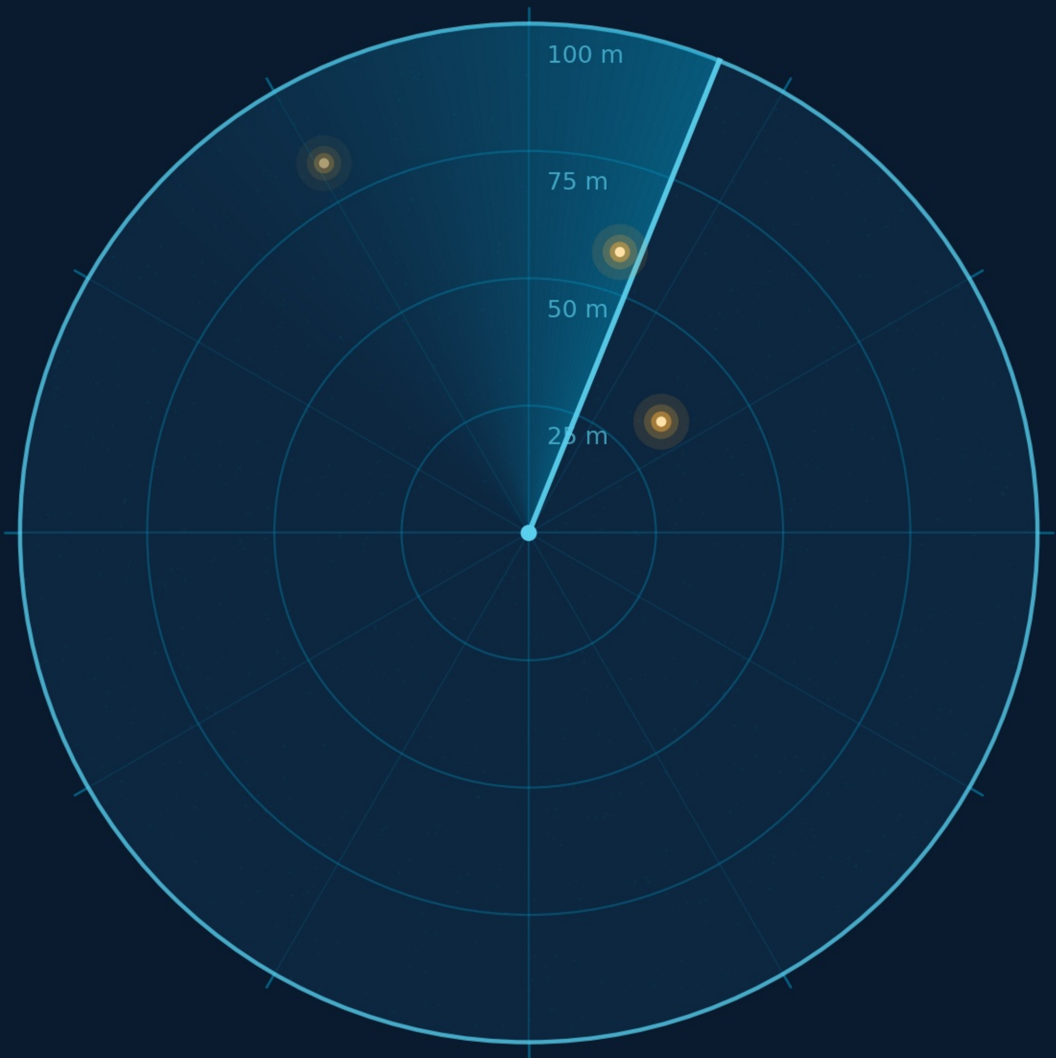
Airborne Drone Detection Using RADAR

DSP Pipeline – Subgroup 2

A. Marjanovic & A. Yaldız

Delft University of Technology



Airborne Drone Detection Using RADAR

DSP Pipeline – Subgroup 2

by

A. Marjanovic & A. Yaldız

Supervisors:	prof.dr. O. Yarovyi
	dr. N. Kruse
Faculty:	Faculty of Electrical Engineering, Mathematics and Computer Science, Delft

Abstract

This thesis investigates drone detection using radar in the K-band, around 24 GHz, where detection is challenging because drones have a low radar cross section (RCS), move at relatively low radial speeds, and must be distinguished from clutter. For range-Doppler detection several Constant False Alarm Rate (CFAR) detectors have been implemented, namely Cell-Averaging CFAR (CA-CFAR), Ordered-Statistic CFAR (OS-CFAR), Greatest-Of CFAR (GO-CFAR), Smallest-Of CFAR (SO-CFAR), and Clutter-Map CFAR (CM-CFAR). For direction of arrival (DoA) estimation, MVDR and MUSIC are used to estimate the azimuth angle. Together, these stages output the range, radial velocity and azimuth of the detected drone. The trade-off between computational cost and detection performance is then studied by building a benchmark and testing the algorithms against it. At close range, from 0 to 25 m, the CFAR algorithm with the highest F_2 score was CA-CFAR, while beyond 25 m CM-CFAR was generally more robust, and among the spatial CFAR algorithms SO-CFAR remained competitive at longer ranges. For the direction of arrival a new range-Doppler guided processing pipeline is introduced, which runs between 80 and 120 times faster than the original full range-angle pipeline. This speed-up comes at the cost of detection performance. The most robust angle estimator was MUSIC combined with argmax selection, although it requires the number of sources to be known beforehand. MVDR with argmax selection achieved slightly lower results but does not carry this requirement. By contrast, in the original full range-angle pipeline, MUSIC combined with the 2D GO-CFAR consistently achieved the highest $F_{0.5}$ scores. Finally, the selected configuration was evaluated on an additional validation flight. Although the detected range trajectory broadly followed the GPS-derived trajectory, the large number of false detections showed that the selected configuration only partially transferred to this flight.

Preface and Acknowledgments

Defending against the growing threat of drones in military warfare, or mitigating the impact of wind turbine farms on bird and bat populations are both applications of modern RADAR technologies. Thanks to Dopplium, a Delft based SME who provided us with multiple radar systems and drones, we were able to write this BSc Thesis about airborne drone detection using RADAR. This project was split into 3 sub-projects exploring RCS modeling of drones, signal processing methods for drone detection and tracking of drones. We chose the second sub-project due to our interest in combining electromagnetic-physics and signal processing. This project would not have been possible without the efforts of dr. F. Fioranelli, dr. N. Kruse, dr. I. Roldan and prof. O. Yarovyi. We thank you sincerely for this opportunity and we hope this thesis reaches the expectations you all set out for us.

-Andrej Marjanovic and Abdullah Yaldız

Contents

1
Introduction

In this chapter, the radars and drone, used in this project, given by Dopplium will be studied. Then the basics governing RADAR physics and signal processing will be explored. Lastly, the goal of the project, a formal definition of the problem and the structure of the thesis will be described.

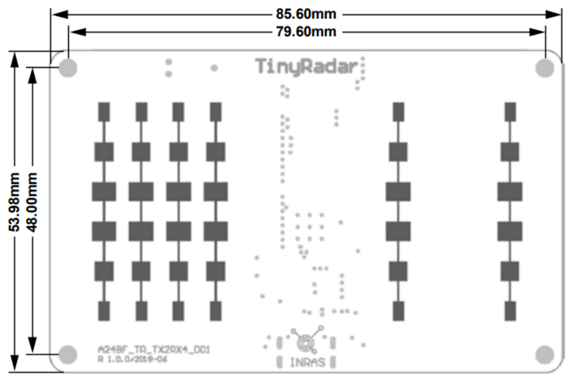
1.1. Provided Equipment

During the measurements, two Frequency Modulated Continuous Wave (FMCW) radar systems and one drone were utilized:

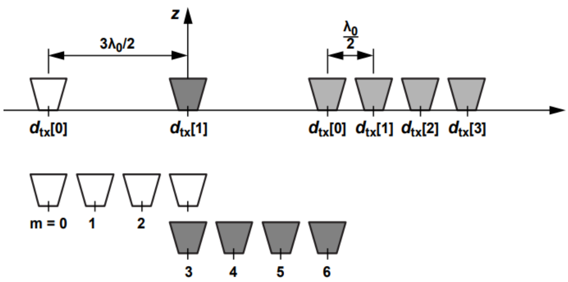
- A modified version of the EV-TINYRAD24G radar by Analog Devices.
- A IWR6843ISK radar by Texas Instruments.
- A DJI Air 3S drone by DJI

1.1.1. EV-TINYRAD24G

The EV-TINYRAD24G operates at a carrier frequency of 24 GHz, it contains 2 transmitter (TX) and 4 receiver (RX) antennas. It can operate in Multiple Input Multiple Output (MIMO) mode. A mechanical drawing and the virtual array are seen in the figure below.



(a) Mechanical Drawing of EV-TINYRAD24G



(b) Antenna Arrangement with Virtual Antenna Positions of EV-TINYRAD24G

Figure 1.1: EV-TINYRAD24G figures from User Guide

Although each element appears as a string of six patches, only one radiates. The other 5 are used for managing the mutual coupling and preventing interference. In MIMO operation, a virtual Uniform Linear Array (ULA) consisting of seven distinct virtual antenna elements is formed. These elements are each spaced half a wavelength apart, which is approximately 6.25 mm, while two virtual antenna channels overlap.

1.1.2. IWR6843ISK

The IWR6843ISK operates at a carrier frequency of 60 GHz, it contains 3 TX and 4 RX antennas. Thus it is also capable of MIMO operation. A picture of the radar and the corresponding virtual array formed by the antennas can be seen in the figure below

(a) Picture of IWR6843ISK

(b) Antenna Arrangement with Virtual Antenna Positions of IWR6843ISK

Figure 1.2: IWR6843ISK figures from User Guide

Similar to the former system, the IWR6843ISK uses two patches for coupling and reducing interference and one patch to radiate. The RX antennas are placed half a wavelength apart (2.5mm), whereas the transmitter patches are a wavelength (5mm) apart, with the TX2 antenna being placed half a wavelength above TX1 and TX3.

A virtual antenna array consisting of an eight-element ULA and a four-element ULA is obtained, with the four-element array separated vertically by half a wavelength above the eight-element array. This allows for angle estimation in both azimuth and elevation planes at the same time.

1.1.3. Drones

In a windless environment, the Air 3S can travel at speeds up to 21m/s, with proper wind conditions this can go up to 27m/s. It has dimensions of 266.11×325.47×106.00 mm (L×W×H) without the propellers.

Figure 1.3: Picture of the Air 3S drone used

1.2. FMCW RADAR Physics

Since both radar systems used are based on FMCW operation, it is logical to review the physics governing these systems. For the sake of simplicity, imagine a radar system with perfect components, only one TX, one RX patch, one target, in an environment with no clutter.

1.2.1. The Sawtooth Chirp and Range Estimation

The radar emits an electromagnetic signal through the TX antenna [1]. This signal is frequency modulated, it starts at a certain carrier frequency f_c and linearly moves up to frequency $f_c + B$ with slope S, and finally drops back to f_c. This process repeats for how long the radar observes. One such ramp is called a sawtooth chirp.

This signal travels in space as an electromagnetic (EM) wave, interacts with an object, reflects and is absorbed by the RX antenna. By mixing the received signal with the transmitted signal, a new signal is constructed which is passed through a low-pass filter. Using trigonometric identities, it is clear that:

$$cos(a) \cdot cos(b) = \frac{cos(a-b) + cos(a+b)}{2} \tag{1.1}$$

This new signal contains two components, one whose frequency is the difference of the two input signals and one whose frequency is the sum of the two input signals. The latter component is then removed by the low-pass filter.

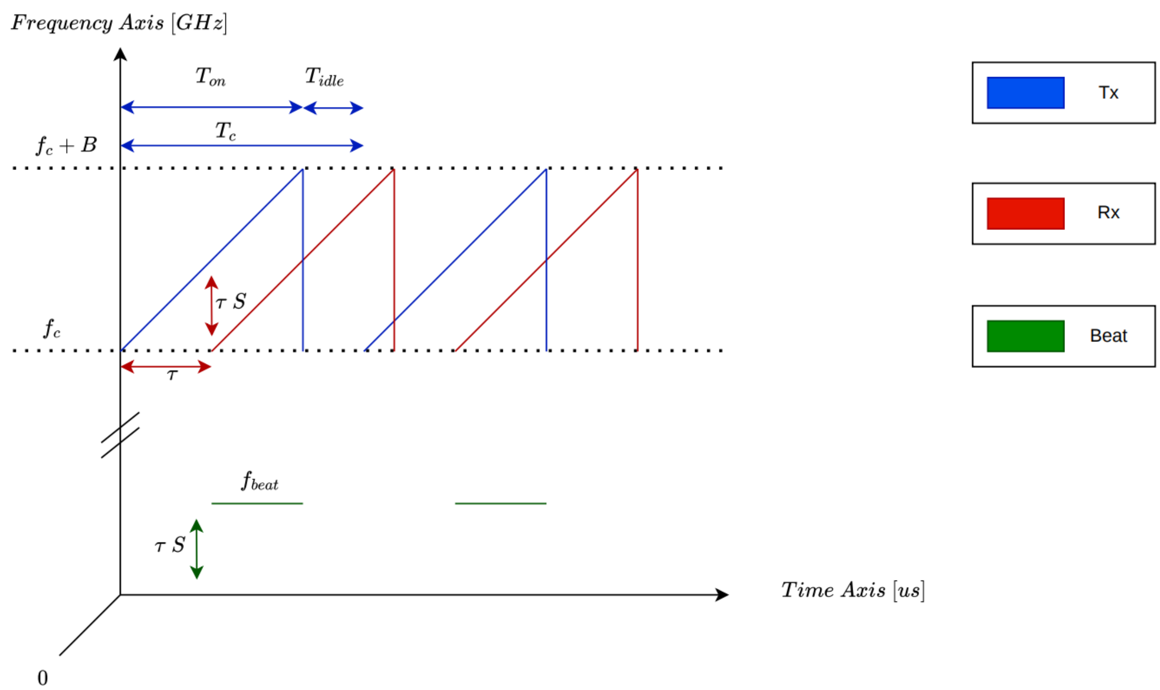


Figure 1.4: Frequency-Time Graph of TX RX and Beat signals

The remaining signal, referred to as the beat signal, contains the frequency $f_b = \tau S$, where τ is the time it took for the TX signal to be sent out, hit a target and return back to be absorbed by the RX antenna. EM waves travel at the speed of light c in vacuum. In practice the waves propagate through air, which reduces the phase velocity slightly. This difference is negligible and ignored throughout. It follows that $\tau = \frac{2R}{c}$. Thus, we can extract the range of a target by using Equation (1.2):

$$R = \frac{c f_b}{2S} \tag{1.2}$$

This whole process is visualized in the figure above, the slope of the chirp is a parameter that is known, since it is set by the operator, but it can be obtained from:

$$S = \frac{B}{T_{on}} \tag{1.3}$$

where B is the bandwidth of the signal, T_{on} is the segment of one chirp T_c without the idle time T_{idle}.

Since f_b is obtained by the FFT of the beat signal, the range resolution ΔR is set by the bandwidth, in other terms:

$$\Delta R = \frac{c}{2B} \tag{1.4}$$

This FMCW range-estimation method is able to estimate the range of multiple targets at different ranges, but suffers from returns due to clutter in the environment that will bring unwanted returns. This is where velocity extraction through coherent integration comes in, allowing us to detect the range and velocity of a target.

1.2.2. Velocity from Slow-Time Phase

Imagine a target moving towards the radar with constant radial velocity v at range R. Using the method above, if two chirps were transmitted and processed, the target would be found in the same range bin for targets with a low radial velocity, even though the target was at ranges R and $R - vT_c$ respectively. This is due to the contrast between the low velocity of the target and the coarse range resolution.

The phase difference of the two returning chirps would not be the same, however. This difference is:

$$\Delta\phi = \frac{4\pi v T_c}{\lambda} \tag{1.5}$$

After N_c chirps, the n-th chirp has a phase $n\Delta\phi$ relative to the first. Looking across chirps, this is a sinusoid with frequency $f_d = \Delta\phi/(2\pi T_c) = 2v/\lambda$. Applying an FFT across chirps at the same index recovers f_d, from which the radial velocity follows:

$$v = \frac{\lambda f_d}{2} \tag{1.6}$$

The FFT resolution is the inverse of the total observation time which is $N_c T_c$:

$$\Delta f_d = \frac{1}{N_c T_c} \tag{1.7}$$

Using the equation for velocity above, the formula for velocity resolution is obtained:

$$\Delta v = \frac{\lambda}{2 N_c T_c} \tag{1.8}$$

If the N_c chirps are gathered into a 2D matrix, each row a beat signal and each column a time sample, an FFT along both axes produces a 2D spectrum showing range and radial velocity of every target. The number of chirps gathered in this data-matrix is referred to as the Coherent Processing Interval (CPI). Static returns collapse onto the zero-velocity bin. This spectrum is referred to as the Range-Doppler (RD) map.

The figure below shows an example RD map. A target at 50 m moving at +5 m/s appears as a distinct peak, while the strong response at zero velocity in the near-range bins is stationary clutter such as buildings or blowing leaves.

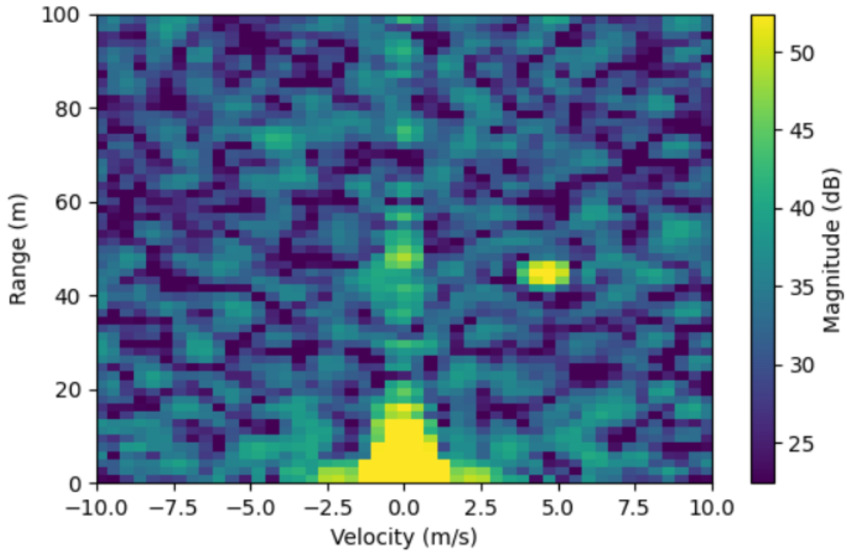


Figure 1.5: Example Range-Doppler map. A moving target appears at 50 m range and +5 m/s radial velocity; the bright column at zero velocity is stationary clutter.

1.3. Radar Software

Both radar systems are operated using Dopplium software. The software provides an interface for configuring radar parameters, initiating measurements, and storing the recorded data. During operation, the radar transmits data with packets of 50 chirps. This is referred to as an acquisition frame. Through processing techniques, the number of chirps in a CPI / frame can be adjusted.

1.4. Problem definition

Unlike detecting large aircraft, or missiles that travel at extremely high speeds, detection of airborne drones poses a distinct challenge, precisely due to it being the opposite of the situation above. Drones fly at relatively low speeds and have a much smaller Radar Cross Section (RCS). New RADAR systems that want to faithfully detect drones need to be able to work around these new physical limitations, yet still be able to filter out noise and clutter from the environment.

1.5. Use of Generative AI

Generative AI models such as Claude and ChatGPT were used during the development and writing of this thesis.

It was an objective to not let AI models guide the design-choices of this thesis, during development these LLM's were only used to accelerate tasks such as creating plotting functions or cleaning up code in order to improve readability. Furthermore, during the writing of the thesis, AI models were only used for verification of the material written, and grammar checks. This thesis does not contain any sections that were only written (and re-written by a human) by AI.

The cover of this thesis was generated using Claude.

1.6. Thesis Structure

Chapter 3 reviews existing signal processing methods for (FMCW) radar. Chapter 4 details the pipeline developed in this work. Chapter 5 presents and discusses the benchmarking framework and its results. Chapter 6 concludes and outlines directions for future work. The Appendices will consist of annotations of the test-flights for testing and benchmarking purposes and our complete testing results.

2

Programme of Requirements

2.1. Requirements for the Entire System

The system is a radar detection and tracking pipeline for aerial targets. It measures the environment, suppresses clutter, detects drones, and extracts the range, radial velocity, and azimuth angle of each detection. The tracking subsystem maintains one track per target.

Functional requirements:

1.1 The radar system shall continuously measure its environment.

1.2 The radar system shall suppress clutter from the received signal.

1.3 The radar system shall detect drones within its field of view.

1.4 The radar system shall estimate the range of each detected target.

1.5 The radar system shall estimate the radial velocity of each detected target.

1.6 The radar system shall estimate the azimuth angle of each detected target.

1.7 The radar system shall form a track for each detected target.

1.8 The radar system shall terminate a track when its target is no longer detected.

2.2. Requirements for the Signal Processing Subsystem

This subsystem is the signal processing chain. It operates on the radar ADC recordings: it suppresses clutter, detects targets on the range-Doppler map, and estimates the range, radial velocity, and azimuth angle of each detection.

Functional requirements:

1.1 The system shall produce a binary detection map from the range-Doppler map using CFAR detection.

1.2 The system shall provide a CA-CFAR detector.

1.3 The system shall provide an OS-CFAR detector.

1.4 The system shall provide a GO-CFAR detector.

1.5 The system shall provide an SO-CFAR detector.

1.6 The system shall provide a CM-CFAR detector.

1.7 The system shall subtract the mean over the chirp dimension before range-Doppler processing to suppress stationary clutter.

1.8 The system shall allow range-Doppler processing with mean subtraction disabled, so clutter-rejection methods can be compared.

1.9 The system shall generate a range-angle map using MVDR direction-of-arrival estimation from range-FFT data.

1.10 The system shall generate a range-angle map using MUSIC direction-of-arrival estimation from range-FFT data.

1.11 The system shall combine range-Doppler detections with direction-of-arrival estimates to output detections containing range, radial velocity, and angle.

Performance and accuracy requirements:

2.1 The system shall estimate drone range within one range resolution bin, equal to 2.15 m at the current radar parameters.

2.2 The system shall estimate drone radial velocity within a fixed velocity tolerance window of 0.5 m/s.

2.3 The system shall estimate the DoA of the drone within a fixed angle tolerance of 10 degrees.

Computational time requirements: An acquisition frame is 50 chirps, and each range-Doppler map is computed from one acquisition frame. Timings are stated per 1000 frames (50000 chirps), which spans 11.2 s of recording.

4.1 CA-CFAR shall process 1000 frames in under 1 s.

4.2 GO-CFAR shall process 1000 frames in under 4 s.

4.3 SO-CFAR shall process 1000 frames in under 4 s.

4.4 OS-CFAR shall process 1000 frames in under 100.0 s.

4.5 CM-CFAR shall process 1000 frames in under 2 s.

4.6 Mean subtraction shall process 1000 frames in under 0.5 s.

4.7 Range-time processing shall process 1000 frames in under 2 s.

4.8 Range-Doppler processing shall process 1000 frames in under 0.5 s.

4.9 Covariance estimation shall process 1000 frames in under 2 s.

4.10 MVDR estimation shall process 1000 frames in under 5 s.

4.11 MUSIC estimation shall process 1000 frames in under 11 s.

4.12 The pipeline shall process 1000 frames in at most 11.2 s, their full duration.

4.13 The detection-and-DoA pipeline shall process 1000 frames in under 20 s.

3

State of the Art

This section of the thesis explores different signal processing techniques used in the field, it will review the advantages and disadvantages of the different techniques and provide results of the implementations.

3.1. Clutter Rejection

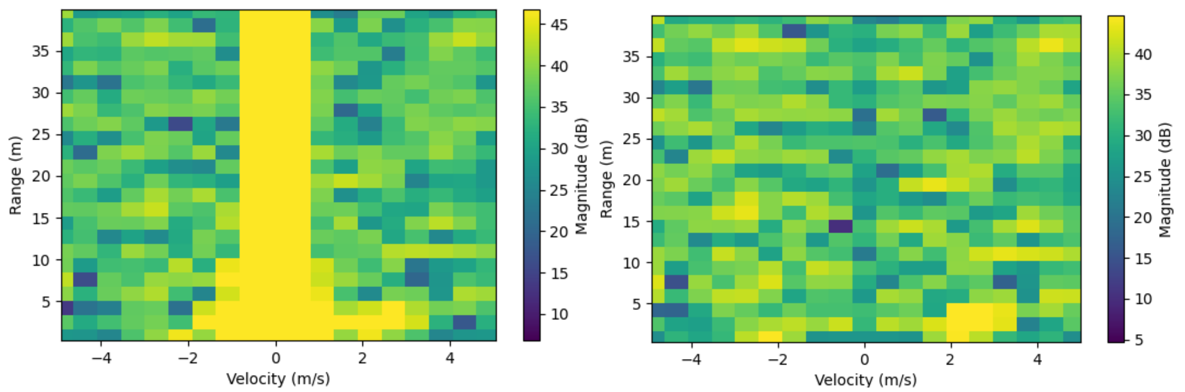
In every environment, there is the presence of reflectors that are not the target, such as buildings, walls and trees. The unwanted returns produced by these reflectors are referred to as clutter. When sufficiently strong, clutter can mask the returns of the desired targets. Removing the returns of this clutter is thus very important.

3.1.1. Mean Subtraction

In our case, most of the clutter in the environment is effectively stationary, such as walls, buildings or trees. The returns of these reflectors are present throughout the entire recording and concentrate in the zero-velocity bin, since stationary reflectors produce a constant phase across chirps. Mean Subtraction (MS) removes these returns by subtracting the mean range profile, averaged over the chirps along the slow-time axis, from the range profile:

$$\tilde{X} = X - \bar{X} \tag{3.1}$$

where $\bar{X}$ is the mean computed along the slow-time axis. As stated in [2], MS works well for uniform stationary reflections but fails for dynamic non-stationary clutter. It may also suppress wanted returns from targets that are either moving slowly or momentarily stationary.



(a) Without mean subtraction.

(b) With mean subtraction.

Figure 3.1: RD map of Flight 3 (drone at 4.5 m, and 3 m/s), without (a) and with (b) mean subtraction.

As seen in Figures 3.1a and 3.1b, MS effectively removes the stationary clutter present in the zero-velocity bins. The drone, visible at the 4.5m and 3m/s bin, is preserved since it is a moving target. Allowing the user to switch MS on and off will satisfy requirements 1.7 and 1.8 mentioned in the PoR.

3.1.2. Go Decomposition

Clutter is usually static, for example: a wall may span many range bins and is present throughout all frames of the recording. The returns of static clutter in the range-Doppler-azimuth map is thus low-rank, as they are highly correlated in time. They can be easily represented by a few dominant vectors. On the other hand, a drone occupies only a few range-Doppler bins at a time, it's also likely to be moving, so its returns are less correlated in time, they are sparse.

GoDec [3] separates the two by decomposing the data matrix as

$$X = L + S, \tag{3.2}$$

where L is the low-rank clutter and S the sparse target.

Tested on the recorded flight data, the decomposition left too much residual noise in S for reliable detection. The technique was not adopted in the final pipeline .

3.2. Windowing

Ideally, a target at range R would produce a Dirac delta at the frequency $f_b = 2SR/c$. However, since the measurement taken is finite in time, this delta spike becomes a sinc function which distributes the range information of the target [4]. This effect motivates the need for sidelobe suppression. There are many techniques, but in this project, windowing (specifically Hamming windowing) will be applied due to its simplicity and effectiveness [5] [6]. This window is applied both along the fast time axis, before computing the range-FFT, as well as along the slow-time axis before computing the Doppler FFT. The main disadvantage of windowing is the widening of the main lobe, which would effectively lower range and velocity resolution.

3.3. Constant False Alarm Rate

As seen before, the RD map yields the range and velocity bin of a target. CFAR algorithms serve a single purpose: determine whether the power in a cell is sufficient to conclude, with statistical confidence, that it is not thermal noise. Clutter cells with enough reflected power will and should be detected.

With the exception of Clutter-Map (CM), all CFAR algorithms require two parameters, n_{guard} and n_{train}. A guard window of n_{guard} cells surrounds the Cell-under-test (CUT) and is excluded from calculations. A training window of n_{train} cells surrounds the guard window and is used to estimate the local noise power $\hat{P}$, which is scaled by a threshold factor α. Everything above this threshold is considered a detection. Noise is statistical by nature, some noise samples can form peaks that exceed this threshold even when no target is present. CFAR does not eliminate false alarms completely.

As seen in [7], the shape of these windows can vary. In Chapter 4, different values for n_{guard}, n_{train} and α for the rectangular window shape will be tested for different CFAR algorithms, measuring accuracy and computational complexity.

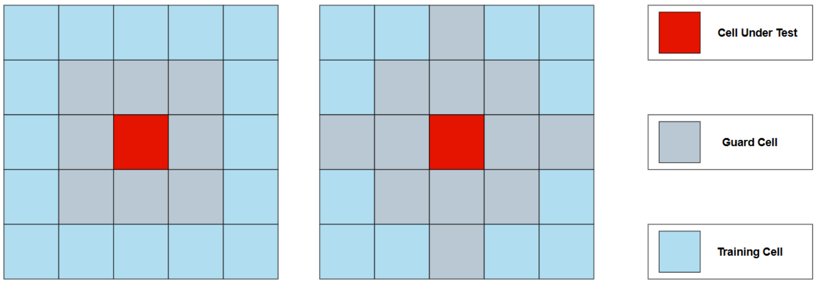


Figure 3.2: Rectangular and Circular window shape for 2D-CFAR

3.3.1. Cell Averaging CFAR

The Cell Averaging (CA) CFAR algorithm [8] takes the mean of the training cells, multiplied by a threshold factor α, as the detection threshold for the CUT. It is the simplest CFAR variant, performing well in homogeneous backgrounds and for point targets. Extended targets spanning multiple cells require adjusted window sizes.

This algorithm has performance issues at clutter edges, and in multi-target environments [9].

3.3.2. Ordered Statistic CFAR

The Ordered Statistic (OS) CFAR sorts the training cell magnitudes in ascending order and selects the k^{th} value. This value, scaled by the threshold factor α, sets the detection threshold. By picking a single ranked cell rather than an average, OS-CFAR is robust to a few interfering targets in the training window [10] [8]. For a given window size (N) and rank (k), the threshold factor α that achieves a design false alarm probability P_{fa} follows from [11]:

$$P_{fa} = k \binom{N}{k} \frac{(k-1)!(\alpha + N - k)!}{(\alpha + N)!} \tag{3.3}$$

The rank (k) is chosen separately, trading detection loss against robustness to interferers, with $k \approx 0.75N$ being a common choice.

3.3.3. Greatest/Smallest Of CFAR

The Greatest-Of (GO) CFAR builds on the CA-CFAR. Instead of averaging all training cells, the training window is split into sections, typically top, bottom, left and right. The mean of each section is computed separately, and the largest of these means, scaled by a threshold factor α, sets the threshold. Selecting the largest mean makes the threshold conservative, which suppresses false alarms at clutter edges [9]. The cost is reduced detection performance in multi-target environments: a strong target within one of the training sections can raise the selected background estimate, causing nearby weaker targets to be masked.

The Smallest-Of (SO) CFAR splits the training window in the same way, but selects the smallest section mean as the threshold. This resolves closely spaced targets and targets with weaker returns well, at the cost of a false alarm rate that exceeds even CA-CFAR at clutter edges.

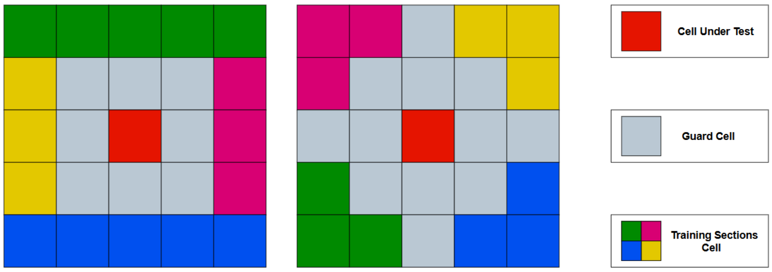


Figure 3.3: Training sections for Rectangular and Circular window shape for GO/SO CFAR

A comparison of the different results of the mentioned CFAR's can be seen in Figure 3.4

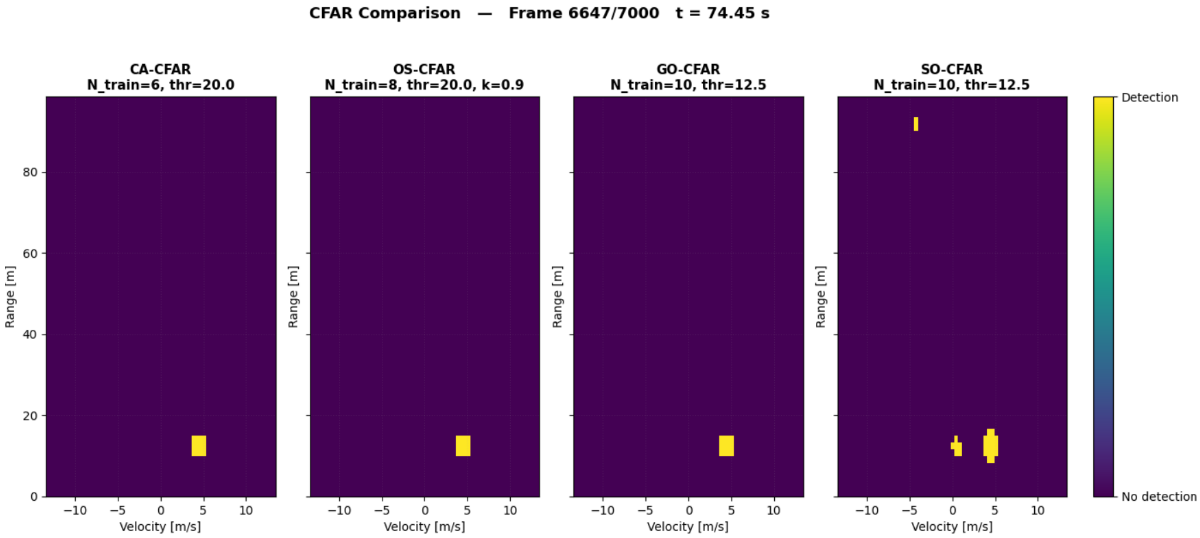


Figure 3.4: CA-, OS-, GO- and SO-CFAR applied to the same range-Doppler frame on a test flight, with the drone present at a range of 11 m and velocity of 4.5 m/s. Each method is shown at its tuned n_{train} and threshold factor α

3.3.4. Clutter Map CFAR

The Clutter Map (CM) CFAR estimates the clutter and noise power temporally instead of spatially [12]. To initialize the CM the range-Doppler power map is computed using a certain amount of blocks where the drone is not present. For each cell, the mean power over these blocks is used as the initial clutter estimate. Next, the clutter estimate of each cell is updated dynamically using Equation (3.4)

$$\hat{p}_n(k) = (1-\beta)\,\hat{p}_{n-1}(k) + \beta\, q_n(k), \tag{3.4}$$

Multiplying the clutter estimate ($\hat{p}_n(k)$) by the threshold factor α gives the detection threshold, as shown in Equation (3.5).

$$\alpha\,\hat{p}_{n-1}(k), \tag{3.5}$$

The algorithm has a higher memory consumption than the previously mentioned CFAR's, requires settling time and may not detect slow-moving/stationary targets that stay in the same bin too long, though it remains strong in nonhomogeneous environments, unlike CA-CFAR.

Implementing all the detection algorithms mentioned in this subsection will fulfill the functional requirements 1.2 till 1.7, including 1.7, that are mentioned in the PoR.

3.4. Direction of Arrival

Direction of Arrival (DoA) refers to the angle from which a signal of interest, in this case a drone, arrives. Three algorithms were implemented to estimate it: MUSIC, MVDR, and a simple angle FFT. Both MVDR and MUSIC require the spatial covariance matrix of the complex range-FFT outputs across the receive channels as well as the steering vectors, so these concepts are introduced first. Implementing the DoA algorithms that are mentioned in this subsection will satisfy requirements 1.9 and 1.10 mentioned in the PoR.

3.4.1. Covariance matrix

The covariance matrix quantifies the spatial relationship between the signals received by the different RX antenna channels. For a system with M antenna channels it is $M \times M$. An example with four RX channels is shown below.

$$\mathbf{R}_{xx} = \begin{bmatrix} R_{x_1,x_1} & R_{x_1,x_2} & R_{x_1,x_3} & R_{x_1,x_4} \\ R_{x_2,x_1} & R_{x_2,x_2} & R_{x_2,x_3} & R_{x_2,x_4} \\ R_{x_3,x_1} & R_{x_3,x_2} & R_{x_3,x_3} & R_{x_3,x_4} \\ R_{x_4,x_1} & R_{x_4,x_2} & R_{x_4,x_3} & R_{x_4,x_4} \end{bmatrix}$$

The diagonal entries give the power received by each antenna channel. The off-diagonal elements R_{x_i,x_j} quantify how similar the complex signal at antenna i is to the one at antenna j, with $i, j = 1, 2, 3, 4$ [13]. The covariance matrix is estimated using Equation 3.6 [14].

$$\mathbf{R}_{xx}[r] = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{x}_r[n]\mathbf{x}_r[n]^H \tag{3.6}$$

where N is the number of chirps used for the estimate and $\mathbf{x}_r[n]$ holds the M channel samples at range bin r for chirp n:

$$\mathbf{x}_r[n] = \begin{bmatrix} x_{r,1}[n] \\ x_{r,2}[n] \\ \vdots \\ x_{r,M}[n] \end{bmatrix}$$

This gives one covariance matrix per range bin per time-block: N chirps are used to measure how similar the RX channels are at each range bin.

3.4.2. Steering vector

The steering vector contains the phase shift for each receive element, corresponding to a signal from a given direction. This can be calculated for every direction if the spacing between the antennas as well as the total antenna array elements are known, as shown in Equation (3.7) [15].

$$\mathbf{v}_M(\psi) = \begin{bmatrix} e^{-j\frac{M-1}{2}\psi} & e^{-j\frac{M-3}{2}\psi} & \cdots & e^{j\frac{M-1}{2}\psi} \end{bmatrix} \tag{3.7}$$

Substituting the phase as given in Equation (3.8) [15], yields Equation (3.9).

$$\psi = \pi \sin(\theta), \quad \text{for } d = \frac{\lambda}{2} \tag{3.8}$$

$$\mathbf{v}_M(\theta) = \begin{bmatrix} e^{-j\frac{M-1}{2}\pi\sin(\theta)} & e^{-j\frac{M-3}{2}\pi\sin(\theta)} & \cdots & e^{j\frac{M-1}{2}\pi\sin(\theta)} \end{bmatrix} \tag{3.9}$$

Note here that if the signal comes from the broadside direction, i.e. $\theta = 0°$, then the steering vector is 1 for every array element.

3.4.3. Angle FFT

An FFT over the slow-time axis resolves the velocity of a target. In the same way, an FFT over the receiving elements can resolve its direction of arrival. A signal arriving at an angle θ reaches each element at different times. This is because there is a relation between the direction of travel of a wavefront, and the time each element receives the wavefront. This causes there to be a phase offset of

$$\Delta\phi_n = n\frac{2\pi d}{\lambda}\sin\theta, \tag{3.10}$$

between the receive antennas, where d is the spacing between elements. The FFT over the M antenna elements recovers this phase difference, and for half-wavelength spacing $d = \lambda/2$ the angle follows directly as

$$\theta = \arcsin\left(\frac{\Delta\phi}{\pi}\right). \tag{3.11}$$

Since the angular precision is limited, it was not used in the final pipeline.

3.4.4. MUltiple SIgnal Classification

MUltiple SIgnal Classification (MUSIC) [16] builds on the idea that the steering vectors of the true sources lie in a subspace orthogonal to the noise subspace. The covariance matrix $\mathbf{R}_{xx}$ contains contributions from both the sources and thermal noise; eigendecomposition separates these into a signal subspace and a noise subspace. $\mathbf{R}_{xx}$ is eigen-decomposed into eigenvalues λ_i and eigenvectors $\mathbf{e}_i$, one for each channel M

$$\mathbf{R}_{xx} = \sum_{i=1}^{M} \lambda_i \mathbf{e}_i \mathbf{e}_i^H, \qquad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M \tag{3.12}$$

If the amount of sources in the environment q and channels M are known and $q < M$, the eigenvalues can be split into two groups. The first largest q eigenvalues contain information about the signal and noise, whereas the remaining $M - q$ eigenvalues contain only information about the noise. The latter eigenvalues can be used to construct the noise subspace $\mathbf{E}_n$. MUSIC now exploits the relationship between the steering vectors and noise subspace into a detector by scanning candidate angles θ:

$$P_{\text{MUSIC}}(\theta) = \frac{1}{\mathbf{v}_M^H(\theta)\,\mathbf{E}_n\mathbf{E}_n^H\,\mathbf{v}_M(\theta)} \tag{3.13}$$

At a true source angle the steering vector $\mathbf{v}_M(\theta)$ is orthogonal to $\mathbf{E}_n$, the denominator goes to zero, and P_{MUSIC} peaks. The q largest peaks give the directions of arrival.

Unfortunately, MUSIC only performs well if the source signals are uncorrelated or weakly correlated, failing for highly correlated signals, as in multi-path propagation or against smart jammers [17], [18]. Coherent sources make the signal covariance rank-deficient, so the steering vectors no longer span the full signal subspace and leak into $\mathbf{E}_n$, breaking the orthogonality the method relies on.

Several methods modify the covariance matrix so that MUSIC also works for coherent targets. The forward and the forward/backward spatial smoothing methods are discussed here. Both need more antenna elements to resolve the same number of sources as the standard method.

Forward smoothing assumes a linear array of L elements, divided into overlapping sub-arrays of size p, giving $K = L - p + 1$ sub-arrays. The smoothed covariance is the mean of the sub-array covariances, as shown in Equation (3.14).

$$\overline{\mathbf{R}} = \frac{1}{K}\sum_{\ell=1}^{K} \mathbf{R}_\ell \tag{3.14}$$

where K is the number of sub-arrays. Each sub-array covariance is a slice of the full covariance from Equation (3.6), as shown in Equation (3.15).

$$\mathbf{R}_\ell^{(f)} = \mathbf{R}\big[\ell : \ell + p - 1,\ \ell : \ell + p - 1\big] \tag{3.15}$$

where ℓ is the sub-array index and p the sub-array size [18]. To resolve q targets, each sub-array must satisfy $p \geq q + 1$ and at least q sub-arrays are needed, so $L = K + p - 1 \geq 2q$ elements are required, against the $q + 1$ of the standard case.

Forward-backward smoothing extends this. In addition to the forward smoothed covariance, the backward smoothed covariance is computed, and the two are averaged into the forward-backward smoothed covariance. The advantage is that it resolves q coherent sources with only $\frac{3q}{2}$ elements, against the $2q$ required by forward smoothing alone [17].

3.4.5. MVDR

MVDR scans candidate angles under two constraints [15]. The distortionless constraint keeps the signal from the look direction at unity gain and undistorted. The minimum-variance constraint minimizes the total output power. As a direct consequence of these two constraints, the signal coming from a test direction θ is unchanged, while the energy from other directions, including interference, clutter, and noise, is suppressed as much as possible. In the practical implementation, first the steering vectors for a grid of candidate angles are calculated. Next, the MVDR power spectrum, given in Equation (3.16), is evaluated for each candidate angle using the corresponding steering vector.

$$P_{\mathrm{MVDR}}(\theta) = \frac{1}{\mathbf{v}_M^H(\theta)\,\mathbf{R}_{xx}^{-1}\,\mathbf{v}_M(\theta)}. \tag{3.16}$$

Peaks in $P_{\mathrm{MVDR}}(\theta)$ indicate likely directions of arrival. Figure 3.5 shows the MVDR pipeline for every frame.

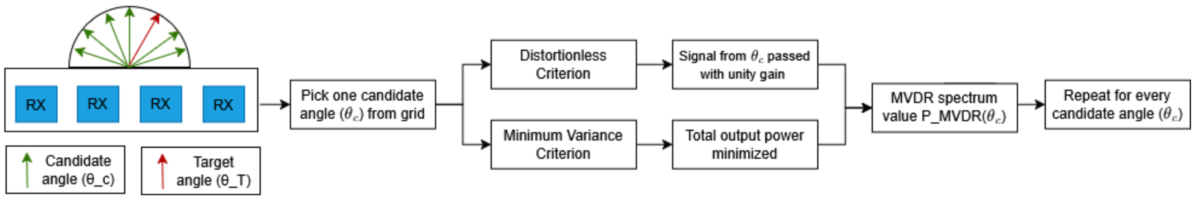


Figure 3.5: MVDR Pipeline

If θ_c corresponds to the true target direction θ_T, then the target signal will be passed with unity gain, while the energy from other directions will be minimized. $P_{\mathrm{MVDR}}(\theta)$ will thus have a peak if $\theta_c = \theta_T$. If the candidate angle θ_c does not correspond to the true target direction θ_T, then the true signal can be suppressed as a consequence of the aforementioned minimum variance criterion. The signal from θ_c will still be passed with unity gain, but this will be a weak signal. So $P_{\mathrm{MVDR}}(\theta_c)$ will be small in this case, unless there is strong clutter, in which case there will still be a peak in the power spectrum.

3.5. Density-Based Spatial Clustering of Applications with Noise

Density-based spatial clustering of applications with noise (DBSCAN) [19] is the most commonly used clustering algorithm in this field and one of the most reliable [20]. It is able to deal with noise, and doesn't require knowing the number of clusters in advance unlike k-means which is ideal for radar-applications where knowing the number of targets is what needs to be determined. It only detects clusters with at least minPts points that lie within a radius ε. Isolated CFAR cells will be labeled as noise. DBSCAN was implemented using the `scikit-learn` Python library.

4
Pipeline

This section describes the design iterations of the DSP pipeline. The first design is outlined and evaluated, and an improved pipeline is proposed.

4.1. Radar Data Cube

Parsing the raw radar data gives a 4-D data cube with axes: acquisition frame, chirp, ADC-sample and receive antenna channel. An acquisition frame refers to a hardware-defined unit containing 50 chirps. This should not be confused with the term frame, which is used refer to a group of consecutive chirps, possibly spanning multiple acquisition frames, that is used for coherent integration and covariance estimation. The ADC-samples contain the beat signal, since the radar does the mixing and the low pass filtering internally.

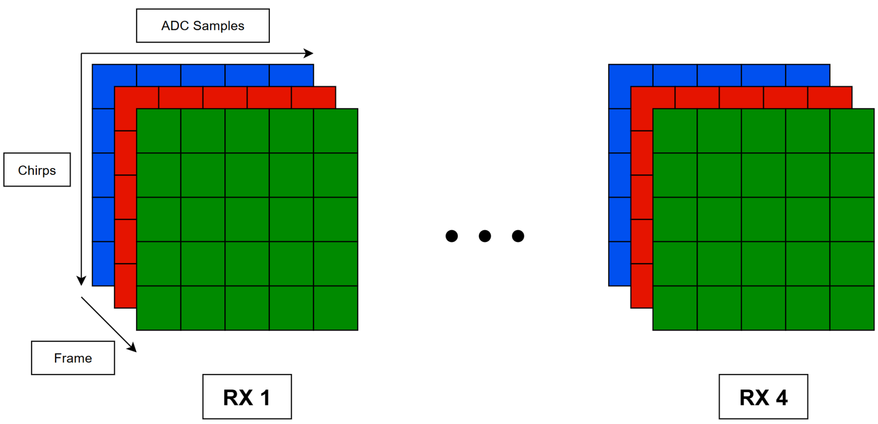


Figure 4.1: Radar data cube

4.2. Simulation

The pipeline is independent of how the data is obtained. It can work with real measurements or data generated from simulated flights. This simulation serves as a parallel input path, producing recordings with known ground truth that are later used to verify detection and angle estimation performance in Chapter 5.

In order to test the algorithms in a controlled environment, simulations of the test flights were created.

$$R[k] = \sqrt{x[k]^2 + y[k]^2 + z[k]^2}, \qquad \theta[k] = \mathrm{atan2}(y[k], x[k]), \qquad v_r[k] = \left.\frac{dR}{dt}\right|_{t=t_k}, \tag{4.1}$$

The amplitude of the synthetic beat signal is set proportional to the square root of the received power, ($A_r[k] = \sqrt{P_r[k]}$), where $P_r[k]$ is calculated using the radar equation, which is shown in Equation (4.2).

$$P_r[k] = \frac{P_t G_t G_r \lambda^2 \sigma[k]}{(4\pi)^3 R[k]^4 L}, \tag{4.2}$$

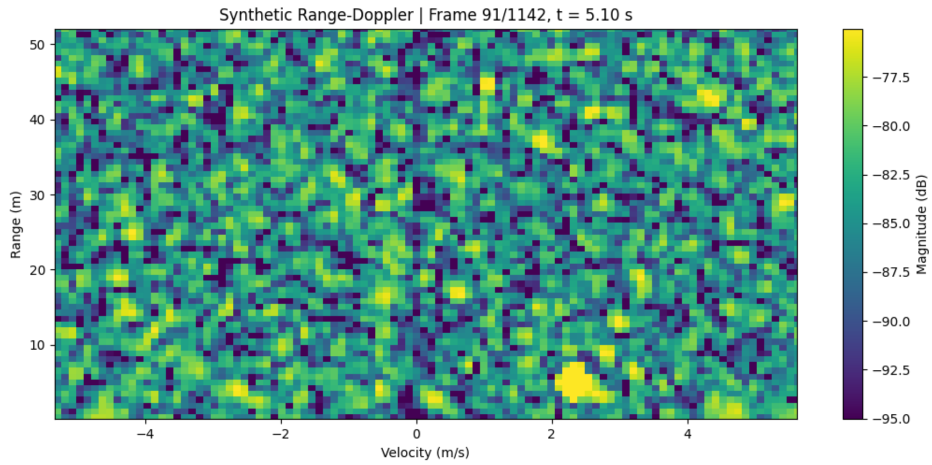
where

$$\begin{aligned} \text{Transmitted Power } P_t &= 6.3096 \times 10^{-3}\ \text{W}, \\ \text{Transmitter/Receiver Antenna Gain } G_t &= G_r = 18.197\ \text{dB}, \\ \text{Wavelength } \lambda &= 12.5\text{mm} \\ \text{Loss Factor } L &= 0\ \text{dB} \\ \text{Radar Cross Section } \sigma(t) &= \text{determined by SG1's model} \end{aligned}$$

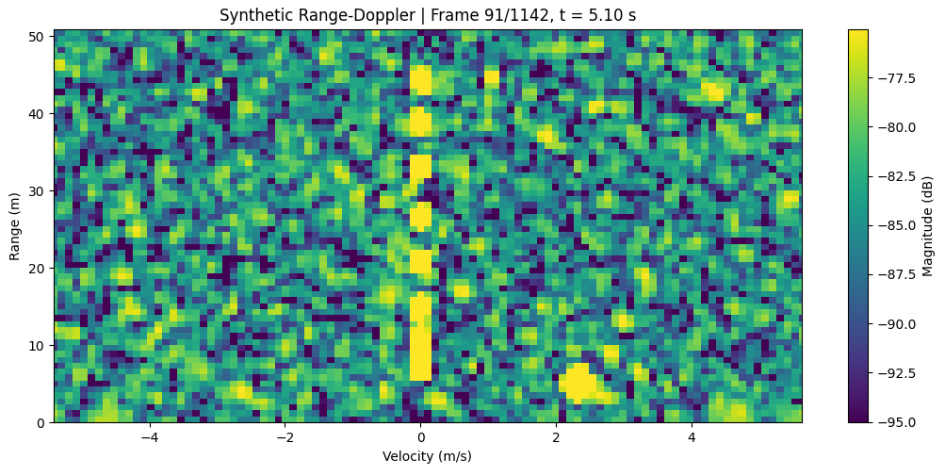
The RCS values were drawn from the Gamma distribution shown below, which was provided to us by subgroup 1.

$$\sigma(t) \sim \Gamma\left(k = 2, \theta = \frac{\bar{\sigma}}{2}\right), \qquad \bar{\sigma} = 0.2257\,\text{m}^2.$$

A new RCS value was drawn every 250 chirps. Consecutive RCS samples were statistically independent. The final simulator models the drone as a point reflector with white noise scaled to the target SNR, with static clutter placed at fixed ranges and angles. The target SNR is defined at a reference range of 30 m in the time-domain beat signal, for a target with mean RCS of $\bar{\sigma}$. The instantaneous SNR increases for shorter ranges, and decreases for longer ranges. The clutter phase is given a slight time variation, though this has negligible effect: because the returns stay nearly constant in phase across chirps, mean subtraction removes them almost completely, as seen in Figure 4.2.



(a) With mean subtraction.



(b) Without mean subtraction.

Figure 4.2: Synthetic range-Doppler map of Test Flight 6 at SNR = −15 dB with $N_d = 250$, with (a) and without (b) mean subtraction. The drone is at 5.3 m, and has a velocity of 2.35 m/s. Without mean subtraction the static return forms a bright ridge at zero velocity, which the mean subtraction removes.

An attempt was made to insert the synthetic drone signal into a background scene recorded during the real measurements. Due to time constraints, this was not fully implemented. This is discussed in more detail in the Appendix.

4.3. Old Pipeline

The old pipeline is shown in Figure 4.3. There it can be seen that first a range FFT is done along the fast-time ADC-sample axis. This gives the complex range spectrum for every chirp and every receive antenna. The pipeline then splits into a range-Doppler branch and a range-angle branch. The range-Doppler branch subtracts the mean across chirps for every range bin to suppress static clutter, runs the slow-time Doppler FFT, and incoherently averages the receive channels into a single range-Doppler map. The range-angle branch keeps the receive-channel dimension. It estimates a covariance matrix R_x for every (frame, range bin) pair and runs MVDR or MUSIC on each R_x over the full $-50^\circ \ldots +50^\circ$,the full angle range visible to the radar at 1° steps. The choice to only scan this angle range is explained in Chapter 5.

The CFAR algorithms then run on both maps independently to produce detections, which are clustered with DBSCAN. The resulting RD and RA clusters are matched by range: an RA cluster is kept only if an RD cluster exists at the same range. Each surviving cluster carries range, velocity and angle, and is passed to the tracking subgroup.

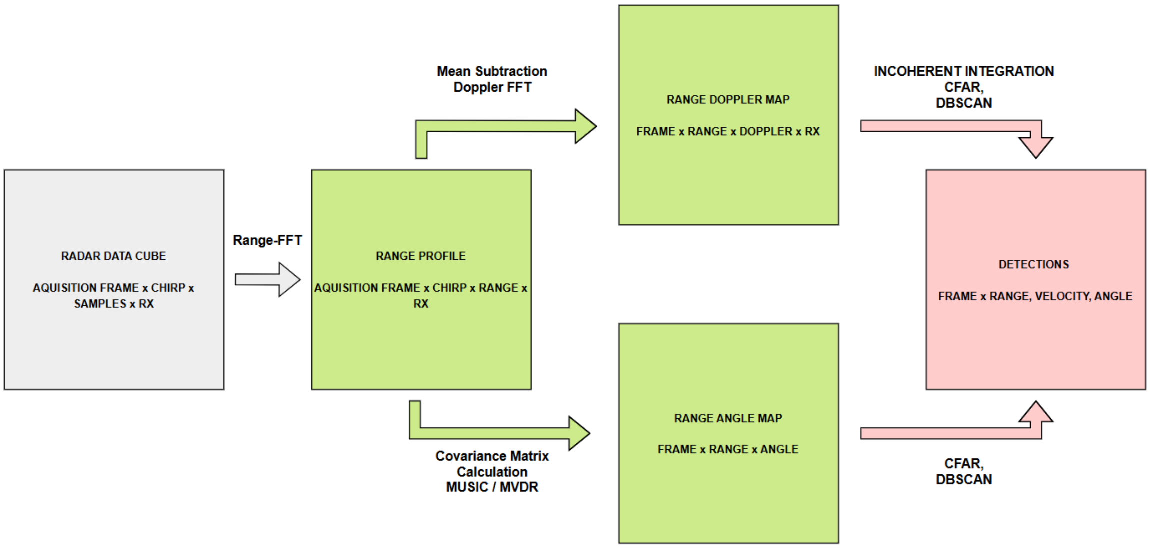


Figure 4.3: Old DSP pipeline

The computational cost is concentrated in two stages (Table 4.1). The range FFT is the single most expensive step, since it transforms every chirp of every receive channel. The covariance estimation is the second largest, because it forms one 4×4 matrix for every (block, range bin) pair across the whole range axis, whether or not a target is present. Together with the MVDR/MUSIC search these make up the range-angle branch.

Table 4.1: Per-frame compute cost of the old pipeline on measured data. MVDR and MUSIC use a 1° angle grid over ±50°. The full covariance forms 5 723 matrices over the 1000 acquisition frames.

Step	Total (ms)	Avg (ms/frame)
Covariance (full)	299.4	0.2994
RA (MVDR, full)	22.4	0.0224
RA (MUSIC, full)	25.7	0.0257

4.4. New Pipeline

The new pipeline is illustrated in Figure 4.4. It builds the range-Doppler map first. The CFAR algorithms and DBSCAN mark the ranges that hold detections in each frame, and only those ranges enter the covariance stage. The covariance is estimated on the detected range bin, and MVDR or MUSIC process only those matrices. From the resulting range-angle spectrum the target angle is taken either by a 1D-CFAR or by selecting the strongest angle bin, the argmax method. The pipeline assumes the range-Doppler stage detects every target of interest.

This ordering follows the convention recommended in the literature: "If the range-Doppler matrix is determined first, followed by target detection using a CFAR algorithm, then an angle estimation is carried out only for range-Doppler cells that contain a target" [21, p. 87].

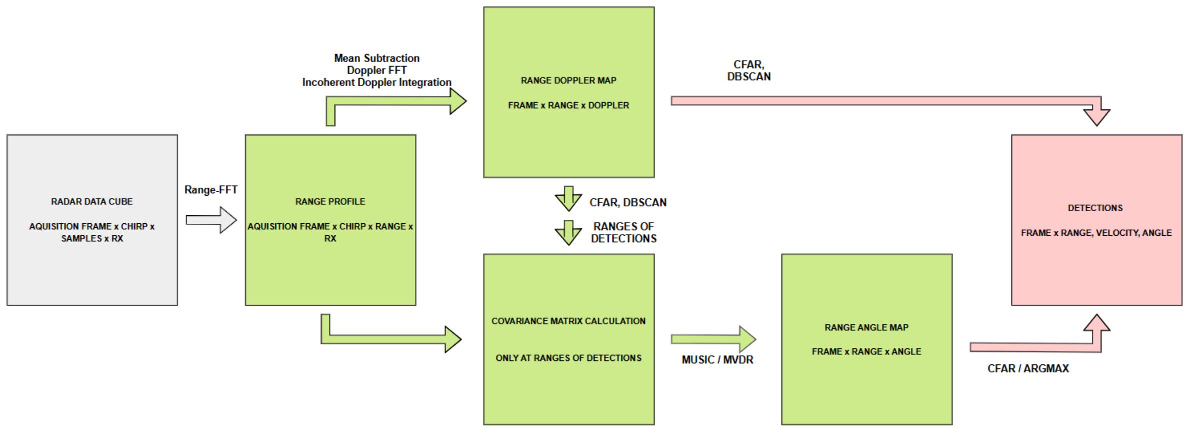


Figure 4.4: New DSP pipeline

On the same measured flight mentioned in the old pipeline subsection, the range-Doppler stage flagged only 101 detections across the 1000 acquisition frames, so the guided covariance forms 101 matrices instead of 5 723, a 56.7× reduction. Table 4.2 shows the covariance and DoA stages shrink accordingly while the shared parts of the pipelines are unchanged.

Table 4.2: Per-frame compute cost of the new (RD-guided) pipeline on Flight 2. The guided covariance forms 101 matrices.

Step	Total (ms)	Avg (ms/frame)
Covariance (guided)	2.2	0.0022
RA (MVDR, guided)	1.5	0.0015
RA (MUSIC, guided)	0.5	0.0005

4.4.1. Background Spectrum Normalization

This works in a similar way to how MS removes static clutter from the range-Doppler map. First, the MVDR and MUSIC power spectra are computed from an empty-scene recording. The spectrum obtained from the actual flight recording is then divided by the corresponding empty-scene spectrum at every range and angle. In the dB domain, this operation is equivalent to subtracting the empty-scene spectrum from the flight spectrum. The returns of the static clutter at the different angles and ranges are reduced, and thus the remaining peaks are more likely to come from the actual targets.

4.4.2. Argmax Angle Selection

After MVDR/MUSIC is applied on the target range-bins, selecting the angle bin with the highest power return is an efficient technique to find the DoA of a target. This technique does not work for multiple targets that are located in the same range bin. Adding a minimum dB requirement on the selected peak further reduces false detections, since a noisy range-Doppler detection can still produce a low peak in the DoA spectrum that does not come from a real target.

4.4.3. 1D CFAR Angle Selection

The same CFAR techniques can also be run on a one dimensional vector, here the power returns across the angle bins at one fixed range bin. A single angle estimate only gives the strongest direction of arrival, so if two drones sit at the same range but at different angles, a plain argmax reports only the stronger one. By sweeping a CFAR window across the angle bins instead, each peak is tested against its own surroundings, and thus both targets can be picked out. This comes at a bigger computational cost compared with argmax, but it is what lets the pipeline distinguish the DoA of multiple targets sharing the same range bin.

Figure 4.5: 1D CFAR

4.5. Speed Improvements

Table 4.3: Per-frame compute cost on flight-2, full (old) versus RD-guided (new) pipeline, with the per-step speedup. The DoA branches sum the covariance with one DoA stage.

Step / Path	Old (ms/frame)	New (ms/frame)	Speedup
Covariance	0.2994	0.0022	138.4×
RA (MVDR)	0.0224	0.0015	15.2×
RA (MUSIC)	0.0257	0.0005	49.1×
DoA branch (cov. + MVDR)	0.3218	0.0036	88.7×
DoA branch (cov. + MUSIC)	0.3251	0.0027	121.3×

In (Table 4.3) it can be seen that there is a significant speedup in the computation of the covariance matrix, as well as the computations of the power spectra for music and MVDR. This is because the covariance matrix is now only computed for the ranges where there is a detection in the range-Doppler map. The number of covariance matrices that need to be computed decrease by a ratio of 56.7, and the actual obtained speedup is even higher. The speedups in MVDR and MUSIC are due to the fact that the power spectra have to be computed for less ranges compared to before.
Although the RD-guided pipeline reduces the computation time of the DoA stage, it may be that this reduction affects the detection and angle estimation performance. Chapter 5 therefore compares the old and new pipeline to determine whether the decrease in computation time comes at the cost of performance degradation.

As seen in the Programme of Requirements, for the radar to operate in real time, the processing time of a frame of 50 chirps needs to be under 11.2ms. The current pipeline sits well under this requirement.

5 Benchmarking

Throughout this report, many different algorithms with each having various different purposes have been explored. All of these algorithms contain different amounts of input parameters that can be varied. First, the way accuracy will be defined is explored, then the different algorithms will be tested against each other, with different parameters with a focus on accuracy and computational speed.

These are the following parameters we will explore:

- N_{GUARD}, N_{TRAIN}, α for the various CFAR algorithms
- k^{th} index for OS-CFAR
- Learning rate β for CM-CFAR
- Number of chirps in a Coherent Integration Interval N_{DOPPLER}

5.1. Benchmarking Pipeline

To evaluate the developed DSP pipelines under realistic conditions, several outdoor test flights were performed using the DJI Air 3S, mentioned in the introduction. This drone is equipped with a GPS-module, from which the ground truth values can be obtained. During each flight, the two radars mentioned in the introduction recorded the reflected signals. Several flight patterns were used to evaluate the pipelines under different ranges, angles, and motion conditions. In this section 3 flights will be discussed. Flight 2: the drone stays close to the radar, and goes side to side. Flight 3: the drone goes in a straight line at broadside far away and returns. Flight 6: drone is again at broadside, every 10 meters it hovers for approximately 5 seconds. These flight patterns are discussed in more detail in Appendix A.

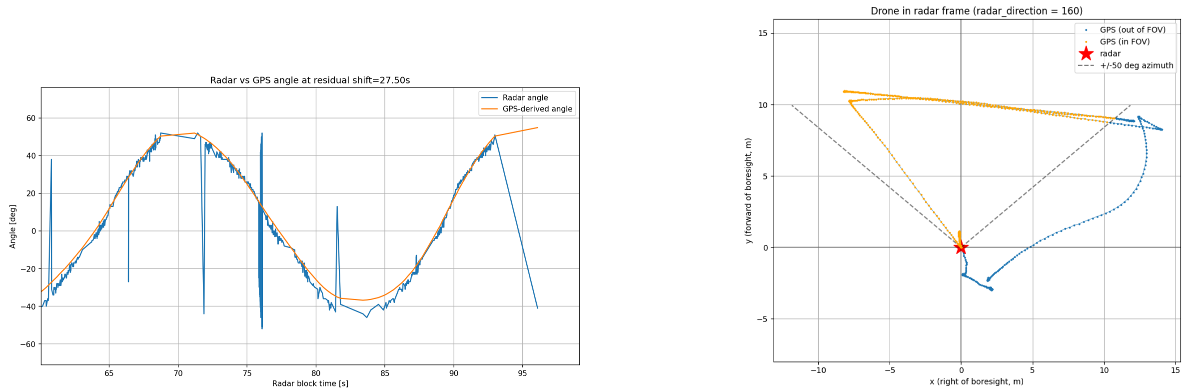
5.1.1. Ground Truth

The test-flight drones carry a GPS module. They store the position using the geographic coordinate system, i.e: longitude, latitude and altitude. The radar position is fixed and known, so the GPS track gives ground truth for both range and azimuth. By comparing the algorithm outputs and the GPS data that will be used as the ground truth, the accuracy of the methods can be measured.

Before testing, the GPS data requires pre-processing. To start, the geographic coordinates will be converted into a local (x, y, z) coordinate system with the radar sitting at $(0, 0, 0)$. To achieve this the geographic coordinates have been converted to a local East-North-Up (ENU) coordinate system using the known geographic coordinates of the radar as the reference point. The direction of the zero azimuth angle was found using the method described in the Appendix.

Since the drone's GPS logs data every 100ms, whereas the algorithms only provide measurements depending on the number of chirps used during a coherent integration process, this can range from 11.2ms for 50 chirps, to 114.688ms for 512 chirps. This requires the drone's position to be interpolated to line up with the measurement times, which may induce small errors.

In addition, the drone's GPS clock and radar clock are not synced up due to internal delays and slightly varying measurement start times. The former issue can be solved by adding an offset in the GPS data with the found time discrepancy, the latter issue can be solved by sweeping through a large range of small time offsets, measuring the RMSE between the range and angle values. Ideally, a single time offset which provides the smaller RMSE value would be used. However, this time offset changed slightly across flights. This was likely due to the drone's GPS logging the time in seconds and not sub-seconds. So the final offset used differed slightly for each flight, with a difference of 0.7 seconds. For every flight, the chosen offset was confirmed visually by using figures as the one shown below.



(a) Drone GPS derived angle versus radar derived angle

(b) Bird-eye view of Test Flight 2

Figure 5.1: GPS versus radar derived angle alongside the bird-eye trajectory for Test Flight 2.

For faithful testing, the ground truth labels should only be present in the benchmarking pipeline for time blocks in which the drone is in view of the radar. Although the Tinyrad has a horizontal 3 dB beam width of 76.5 (which corresponds to ±38.25°azimuth) degrees, the measured results from Flight 2 (see Figure 5.1a) show that the radar-derived angle follows the GPS-derived angle up to approximately ±50°. To this end, an empirical operational Field-Of-View (FOV) mask was applied with limits of 100m of range and ±50°azimuth.

5.1.2. Pipeline

The accuracy and computational runtime of the algorithms will be evaluated using both simulated and real data of specific test flights. The old and new pipelines will be tested, to assess whether range-guided DoA methods perform as well as unguided ones.

After computing the Range-Doppler Map, the different CFAR algorithms with varying parameters will be applied, using DBSCAN the detections will be clustered. The position of these clusters will be compared to the ground truth. A cluster is counted as a correct detection if its center lies within one range bin $\sim 2.15m$ and within a fixed radial velocity tolerance of 0.5 m/s from the ground truth, in accordance with the Programme of Requirements. Clusters outside the defined tolerances are counted as false detections, while ground truth targets without a matching cluster present within the defined boundaries will count towards the missed detections. It was decided to choose a fixed velocity tolerance, instead of basing it on the velocity bin width. This was done, because a bin based tolerance would correspond to different physical radial velocity tolerances since the Doppler-bin width depends on N_{Doppler}. Choosing a fixed radial velocity tolerance ensures that the results obtained are all judged on the same physical criterion.

The performance of the RD-CFAR algorithms will be quantified by looking at the F_2 score, which prioritizes recall (real detections of the drone) over the precision (accuracy of predictions). This metric aligns with the constraint mentioned in the Chapter 4. If a target is missed in the RD stage, it cannot be processed in the DoA stage either.

The same process will be repeated on the different DoA algorithms for both pipelines. A cluster will be counted as a hit, if it is within one range bin $\sim 2.15m$ and $\sim 10°$ from the ground truth, as described in the Programme of Requirements. Since the DoA algorithms also produce a Range-Angle Map, only their detections will be passed onto the Tracking subgroup. As a target will have to appear in both the RD and RA map, noisy RD detections will be filtered out, reducing the number of false detections. The performance of the DoA algorithms will be quantified by looking at the $F_{0.5}$ score, which prioritizes minimizing false detections over having a high recall. Through this method, the noisy RD detections will be filtered out as mentioned above.

5.1.3. Test Flights

During benchmarking, the performance of the algorithms was assessed on three flights. The specific details of each flight, results of the algorithms can be found in the Appendix.

These flights include:

- Test-Flight 2: Side to Side
- Test-Flight 3: Max Range
- Test-Flight 6: Hovering at Different Ranges

5.1.4. Flight Simulations

Using the methods explained in Chapter 3, simulations of the three drone flights were created. Each one follows the same path, velocity and angle as the real flight counterpart.

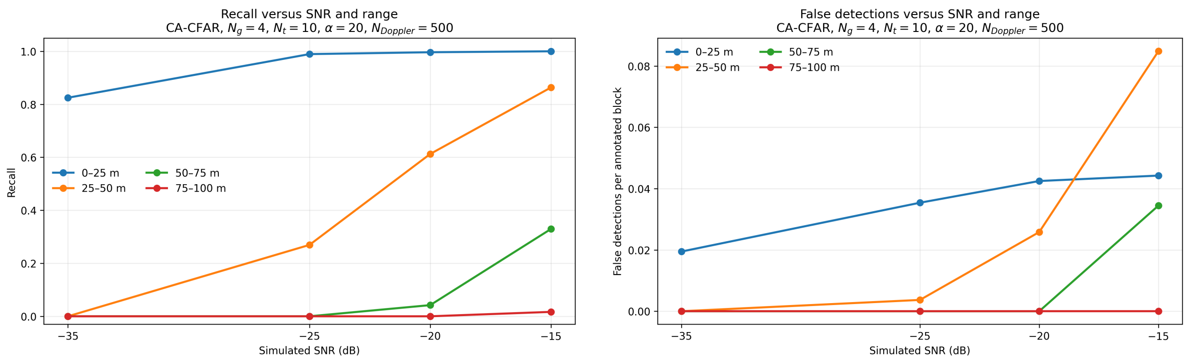
5.2. Results and Discussion

The comprehensive results of the different flights can be found in the corresponding appendices. This section will cover the findings from the current algorithms with the parameters that were swept

5.2.1. Simulation results

Several key results are discussed in this section. First, the effect of the SNR on the recall and on the false detections will be shown. Then the effect of choosing different threshold factors will be compared. This will be followed by analyzing how changing the amount of chirps that were used to coherently integrate changes the recall and false detections at each range.

Effect of SNR on performance The averaged recall and false detections as a function of target SNR and target range are shown in Figure 5.2. It can be seen that both the recall and the false detection rate increase as the target SNR increases. This is expected, because a higher SNR makes it more likely for the target return to exceed the threshold. The slight increase in the false detections might be due to the target-related sidelobes also exceeding the threshold as SNR increases. The averaged recall is calculated as the ratio between the number of blocks in which the drone was in the field of view of the radar at each range interval and the amounts of blocks in which the drone was correctly detected in that range interval. The averaged false detections was calculated in a similar manner.



(a) Averaged recall as a function of simulated SNR and target range.

(b) Averaged false detections as a function of simulated SNR and target range.

Figure 5.2: Detection performance as a function of simulated SNR and target range: (a) averaged recall and (b) averaged false detections.

Effect of different threshold factors/CFAR algorithms on performance In Figure 5.3 the effect of choosing different threshold factors can be seen. As expected, choosing a higher threshold factor does decrease the amount of false detections as well as the recall. It can also be seen that for the same threshold factors SO-CFAR gives the highest recall as well as the highest number of false detections, while GO-CFAR gives the lowest recall as well as the lowest number of false detections. CA-CFAR performs between these models. This is expected, since GO-CFAR is considered more conservative, while SO-CFAR is more sensitive.

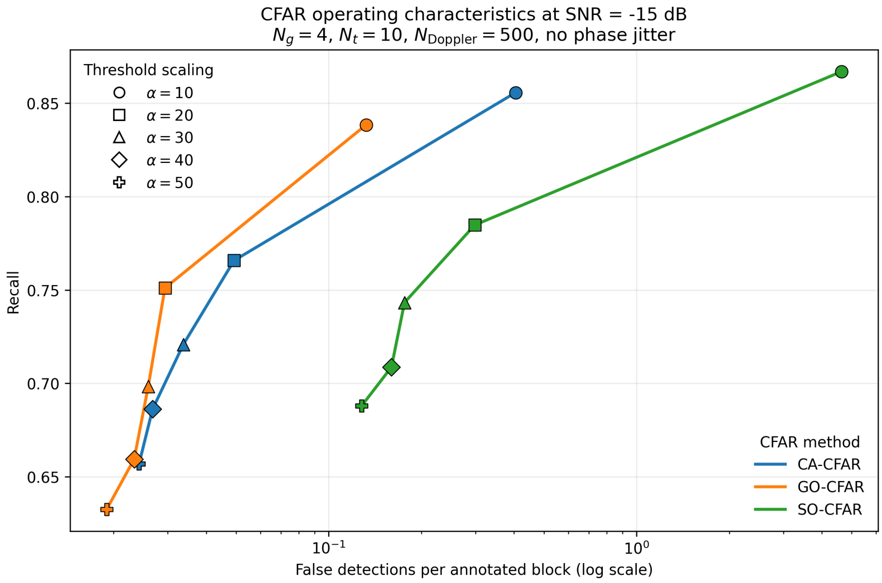
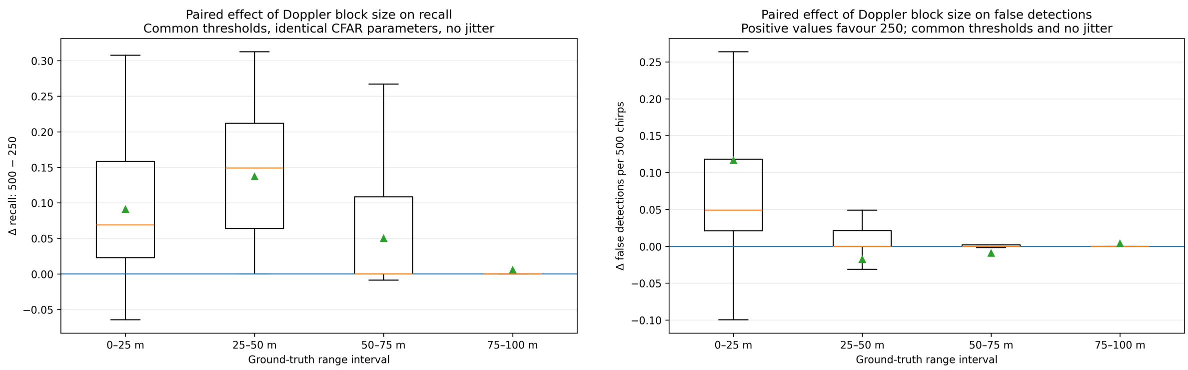


Figure 5.3: Operating characteristic for simulated data at an SNR of −15 dB.

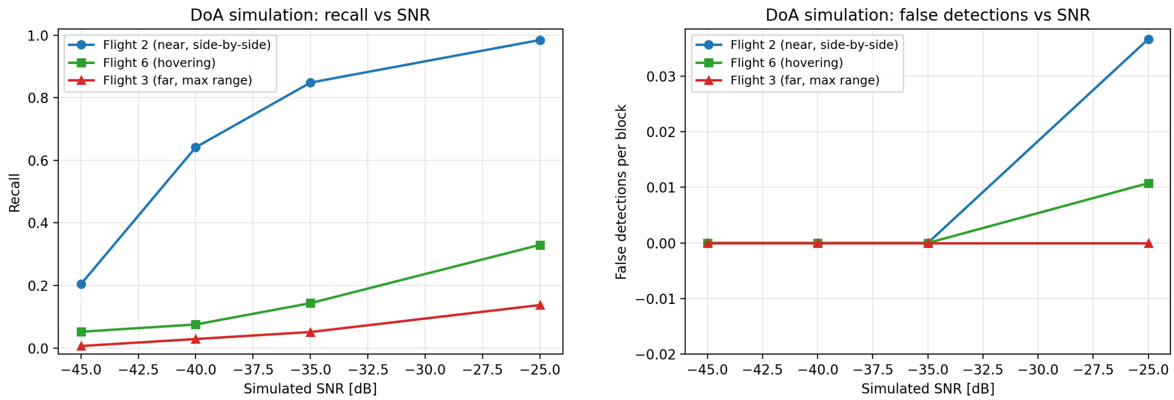
Effect of the frame size In Figure 5.4 it can be seen that increasing the frame size generally increases the recall. This is likely because the coherent integration window is increased, which provides a higher integration gain, making weaker target returns easier to detect. It can also be seen that the number of false detections also increases for increasing frame sizes. This is once again likely due to the target-related sidelobes becoming more apparent and being detected as targets.



(a) Paired difference in recall between Doppler block sizes 500 and 250. **(b)** Paired difference in false detections between Doppler block sizes 500 and 250.

Figure 5.4: Paired effect of increasing the Doppler block size from 250 to 500 samples across the ground-truth range intervals: (a) change in recall and (b) change in false detections.

Direction of arrival results For the DoA evaluation, even lower SNR values were considered than those used in the range–Doppler evaluation. This is because the DoA algorithms seemed to produce extremely idealistic results at even the lowest SNR values, as can be seen from Figure 5.5. These results show that the direction of arrival works well, especially for shorter ranges.



(a) Recall as a function of simulated SNR for each flight.

(b) False detections per block as a function of simulated SNR.

Figure 5.5: DoA simulation performance against the simulated SNR, with each flight standing for a different range regime: (a) recall and (b) false detections.

From Figure 5.5 and Figure 5.2 it can be concluded that the simulations are too idealistic. The false detections stay very low, and are even zero in some cases. Therefore, the simulation results were not used to guide the design choices.

5.2.2. Real Recording results

This section gathers the primary findings from the three measured flights. The detailed per-flight tables and discussions are given in Appendix A, here key patterns and findings are summarized and analyzed.

The Texas Instruments radar gave poor results. We suspect that this is due to the higher atmospheric attenuation at 60 GHz. The radar was not able to reliably detect the drone, as seen in Figure 5.6, where the drone leaves almost no trace in the range-time plot beyond 10m. For this reason the rest of this section only explores the results obtained from the Analog Devices radar in SIMO operation.

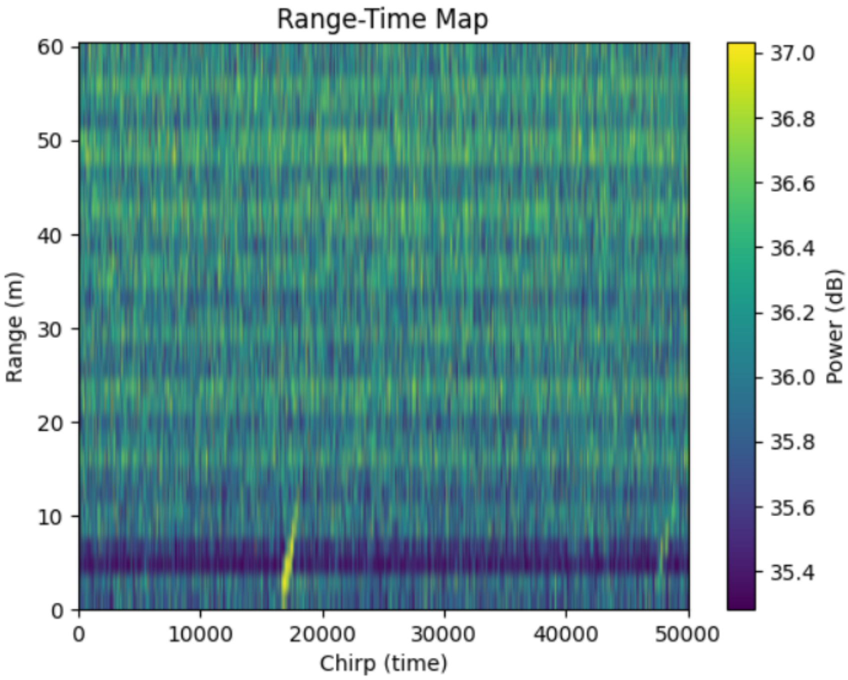


Figure 5.6: Range-time plot of Flight 3 recorded with the TI radar.

CFAR Algorithm Computational Times The computational times of the different CFAR algorithms are shown in Table 5.1. It can be seen that OS-CFAR takes significantly more time, while CA-CFAR is the fastest spatial CFAR variant. All of these times satisfy the requirements 4.1-4.5 that are set in the Programme of requirements.

Table 5.1: Average computation time of the CFAR algorithms over 10 runs using 1000 input radar acquisition frames and $N_{\text{Doppler}} = 512$. All spatial CFAR methods used $N_{\text{GUARD}} = 2$, $N_{\text{TRAIN}} = 6$, and a threshold scaling of 20. OS-CFAR used $k = 0.75$.

CFAR algorithm	Total time (s)	Time (ms/input frame)
CA-CFAR	0.584	0.584
GO-CFAR	1.058	1.058
SO-CFAR	1.190	1.190
OS-CFAR	37.892	37.892
CM-CFAR	0.352	0.352

The average time it took to compute the range-FFT of a 1000 frames was 1.29 s. The average time it took to do mean subtraction was 0.05 s. And lastly the average time it took to compute the range-Doppler FFT was 0.223 seconds. With these results, requirements 4.6-4.8 that were mentioned in the Programme of Requirements are satisfied.

Effects of the range The results obtained from Flight 3 and 6 demonstrate that the range strongly affects the performance, while Flight 2 demonstrates the relatively strong performance that is obtained when the drone remains within 25 m. This can be seen clearly in Table 5.2. Here it can also be seen that Flight 6 produces the lowest F_2 scores in the close and intermediate range interval. This relatively poor performance may be caused by the hovering occurring in Flight 6. When the drone is hovering its radial velocity is close to zero, which can cause mean subtraction to suppress parts of its returns. This is the same mechanism that helps CA-CFAR in the 0-25 m interval, which will be covered shortly, by removing the static hill, only here it works against detection because the target itself sits near zero Doppler.

Table 5.2: Best F_2 score per range interval for the best spatial CFAR algorithm and CM-CFAR.

Range interval	Method	Flight 2	Flight 3	Flight 6
0–25 m	Spatial CFAR	0.670 (CA)	0.683 (CA)	0.508 (CA)
0–25 m	CM-CFAR	0.606	0.608	0.466
25–50 m	Spatial CFAR	–	0.579 (GO)	0.289 (SO)
25–50 m	CM-CFAR	–	0.706	0.382
50–75 m	Spatial CFAR	–	0.067 (SO)	0.203 (SO)
50–75 m	CM-CFAR	–	0.386	0.182
75–100 m	Spatial CFAR	–	0.012 (SO)	–
75–100 m	CM-CFAR	–	0.056	–

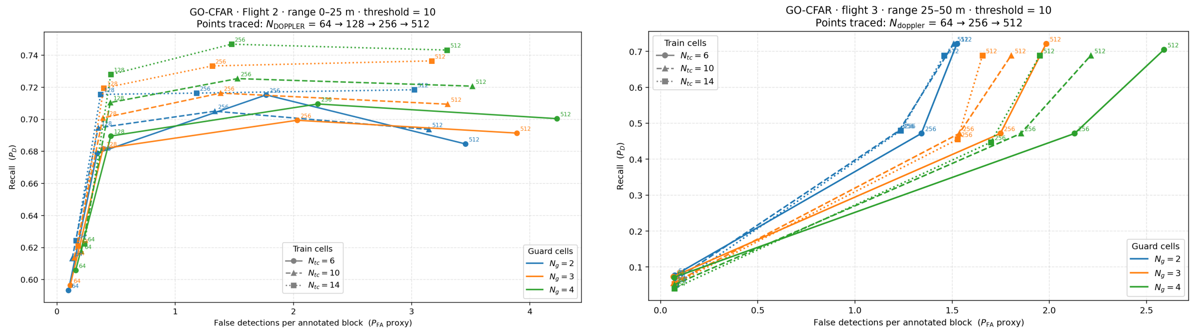
Since the algorithm and parameter combinations were optimized independently of each other, Table 5.2 presents the best obtained F_2 score in the full sweep. Therefore, it does not offer a comparison where range is the only changing variable.

No single CFAR variant performs the best under all conditions. In the 0-25 m range interval CA-CFAR performed the best, which may be due to the strong drone returns at close ranges together with the post-mean-subtraction range-Doppler map being relatively homogeneous in this range. Although a small hill produces strong returns at the 20-25 m range, these returns are mainly concentrated around the zero-Doppler bin, and mean subtraction suppresses most of its stationary component. Therefore CA-CFAR performs well in this case, but this does not serve as evidence that the original scene itself was homogeneous, only that the post-mean-subtraction RD-map was approximately homogeneous. Farther out, CM-CFAR produced the highest F_2 score in four of the five evaluated flight-range combinations. Between the spatial methods, SO-CFAR took the lead four out of five times in the longer ranges. This is consistent with SO-CFAR being more sensitive, which increases the recall at the cost of more false detections, which can be seen quite clearly in the Appendix A. Because the F_2 score weighs recall more severely, this sensitivity can still produce high F_2 scores despite the increase in false alarms. As mentioned previously, the settings of the radar in the recordings do not allow for detection of targets with radial velocities higher than $\pm 14.8 \mathrm{m/s}$. This can negatively impact the detection performance of the algorithms depending on the scene.

Effects of tunable parameters

From all the tunable parameters we tested, the number of chirps in the coherent processing interval (CPI) N_{DOPPLER} and threshold factor α had the biggest effect. The number of guard N_{GUARD} and training N_{TRAIN} cells had less substantial effects.
Higher N_{DOPPLER} raised the recall at the cost of more false alarms. The longer CPI allows for sufficient phase variation, which may cause slowly moving clutter to survive mean subtraction. This change in Doppler resolution also allows the micro-Doppler signatures of the drone to become more distinct. These bins get registered as independent clusters which may get flagged as false detections. The increase in recall is likely caused by the greater coherent processing gain obtained at higher N_{DOPPLER} values.

In Figure 5.7a it can be seen that increasing N_{DOPPLER} from 128 to 256 slightly increases the recall, while substantially increasing the amount of false detections for the 0-25 m range of Flight 2. Increasing it further to 512 does not give any noticeable benefit, while the false detections continue to rise. In contrast, Figure 5.7b shows that increasing N_{DOPPLER} further continues to increase recall in the 25-50 m range of Flight 3, although the number of false detections also increases. These results suggest that the preferred $N_{DOPPLER}$ depends on the range and scene.



(a) Flight 2, 0–25 m

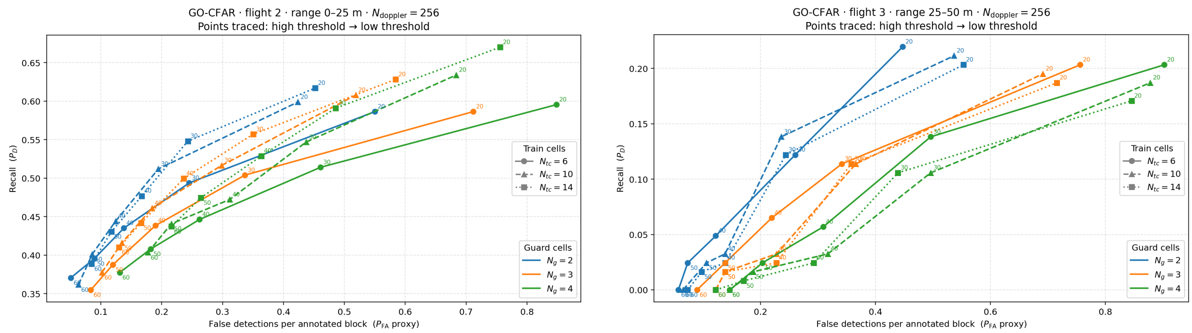
(b) Flight 3, 25–50 m

Figure 5.7: GO-CFAR N_{DOPPLER} sweeps for Flights 2 and 3 using a threshold-scaling factor of 10. The additional value $N_{\text{DOPPLER}} = 128$ was evaluated only for Flight 2.

In Figure 5.8 it can be seen that a higher threshold factor caused a decrease in both the false detections and the recall. From Figure 5.8a it can be seen that for a fixed n_{guard} and a fixed threshold factor, the recall generally increases as n_{train} increases. In the same Figure it can also be seen that increasing n_{guard} for a fixed threshold factor and a fixed n_{train} generally shifts the results toward higher recall and more false detections.

In contrast, in Figure 5.8b it can be seen that using a smaller n_{train} frequently produces a higher recall for a fixed threshold factor and a fixed n_{guard} . This opposite trend may be explained by the difference of the local clutter between the different ranges. A higher n_{train} produces a more spatially averaged clutter estimate, but it may also include cells that are less representative of the local background surrounding the CUT. The Figure also shows that changing n_{guard} for a fixed threshold factor and a fixed n_{train} does not show a consistent trend like in Flight 2. The inconsistent effect of n_{guard} may also be explained by the difference of the local clutter between the ranges. Choosing a higher n_{guard} reduces the target-contamination of the training cells, but it also pushes the training cells further away from the CUT.

These results demonstrate that the preferred n_{train} and n_{guard} also depend on the range and scene.



(a) Flight 2, 0–25 m

(b) Flight 3, 25–50 m

Figure 5.8: GO-CFAR threshold-scaling sweeps for Flights 2 and 3 using $N_{\text{DOPPLER}} = 256$.

Direction of Arrival For the GO-guided version of the new pipeline MUSIC combined with argmax and an additional fixed threshold level produced the highest $F_{0.5}$ score in all three flights. The same configuration was best in the CM-CFAR guided pipeline for flights 3 and 6, while for flight 2 MUSIC combined with 1D SO-CFAR produced a higher $F_{0.5}$ score. The best results were all obtained with the use of background-spectrum normalization.

For the old pipeline, the 2D GO-CFAR, with background-spectrum normalization, was consistently the best pick based on the $F_{0.5}$ score. The old pipeline's best picks produced a higher $F_{0.5}$ score than the new pipeline's picks for flights 3 and 6, while the results for Flight 2 were very similar. These results are shown in Table 5.3

Table 5.3: Best $F_{0.5}$ score per DoA pipeline and flight. The corresponding DoA and angle-selection configuration is shown in parentheses.

Pipeline	Flight 2	Flight 3	Flight 6
Old (2D RA-CFAR)	0.418 (MUSIC + GO)	0.360 (MUSIC + GO)	0.340 (MUSIC + GO)
New — GO-guided	0.420 (MUSIC + argmax)	0.335 (MUSIC + argmax)	0.331 (MUSIC + argmax)
New — CM-guided	0.360 (MUSIC + SO)	0.277 (MUSIC + argmax)	0.276 (MUSIC + argmax)

This table also demonstrates that the angle estimation results depend on the parameters that were chosen for the range-Doppler detections in the new pipeline. Two different range-Doppler CFAR parameter sets were used to assess the angle estimation pipeline. These were the GO-CFAR, and the CM-CFAR with the following parameters : GO-CFAR with $n_{guard} = 2$, n_{train}=6 and $\alpha = 20$, CM-CFAR with learning rate = 5×10^{-4} and threshold factor of 15. The GO-CFAR guided angle estimation consistently yields better results compared to the CM-CFAR guided angle estimation. This can be explained by looking at the obtained results using these parameters, which are shown in Table 5.4 and 5.5. From these tables it becomes clear that the recall and the F_2 score of the CM-CFAR guided angle estimation is higher. However, it also provides more false detections to the DoA stage. The RD results are chosen based on the F_2 score, which favors higher recall, and the DoA results are evaluated using the $F_{0.5}$ score, which favors a lower number of false detections. This can cause a configuration that is better in the RD stage to have a worse performance in the DoA stage.

Table 5.4: GO-CFAR performance across the three measured flights using $N_{\text{Doppler}} = 512$, $N_{\text{GUARD}} = 2$, $N_{\text{TRAIN}} = 6$, and a threshold scaling of 20.

Flight	Recall	FA/block	F_2
Flight 2	0.606	1.169	0.525
Flight 3	0.266	0.805	0.263
Flight 6	0.254	0.851	0.248

Table 5.5: CM-CFAR performance across the three measured flights using $N_{\text{Doppler}} = 500$, a threshold scaling of 15, and a learning rate of 5×10^{-4}.

Flight	Init. blocks	Recall	FA/block	F_2
Flight 2	455	0.732	1.697	0.569
Flight 3	455	0.277	0.808	0.273
Flight 6	455	0.306	1.258	0.275

General findings Overall, RD detection is strongest at close ranges, but suffers substantially after 50 m. CM-CFAR generally retains better performance than the spatial CFAR methods at longer ranges. Increasing N_{Doppler} can improve recall in these longer-range conditions, but it also increases the number of false detections. A higher threshold can reduce the number of false detections here, but this also reduces the recall.
For the angle estimation, the old pipeline produces higher $F_{0.5}$ scores than the new pipeline in Flight 3 and 6, while it produces practically identical results in Flight 2. The GO-GUIDED variant consistently outperforms the CM-CFAR guided variant. This illustrates that the quality of the RD detection affects the overall DoA performance. The GO-CFAR-Guided configuration is evaluated on an additional flight in the subsection 5.2.3. Due to time constraints, the CM-CFAR-guided configuration was not validated on an additional flight.

5.2.3. Range and angle RMSE and validation

In addition to the recall and false detections, the localization error was estimated using the range and angle Root-Mean-Square-Error (RMSE) of the final detections. This was only evaluated for the GO-CFAR guided pipeline with the following parameters: $n_{guard} = 2$, $n_{train} = 6$ and threshold factor = 20.

To evaluate the RMSE, the detected range and angle were compared with the GPS range and angle at the corresponding detection time. The RMSE quantifies the deviation of the generated detections from the actual trajectory. It should be noted that missed detections do not influence the RMSE, and are instead solely evaluated through the recall. Furthermore, false detections can increase the RMSE, which will be seen briefly. Due to this, the RMSE values should not be solely interpreted as the localization error of correctly associated detections.

Flight specific configurations First, the RMSE was evaluated using the configuration that obtained the highest $F_{0.5}$ score for both MVDR and MUSIC, and for each flight. These configurations can be found in Appendix A. The resulting range and angle RMSE can be seen in Table 5.6.

Table 5.6: Range and angle RMSE obtained using the selected GO-CFAR-guided DoA configurations.

Flight	DoA algorithm	Range RMSE [m]	Angle RMSE [°]
Flight 2	MVDR	5.669	14.549
	MUSIC	3.318	9.109
Flight 3	MVDR	38.796	18.854
	MUSIC	7.011	8.715
Flight 6	MVDR	5.178	11.889
	MUSIC	5.501	7.210

It can be seen that MUSIC in general produced both a lower range RMSE as well as a lower angle RMSE, with the exception of the range RMSE in Flight 6. In Flight 3 it can be seen that the MVDR range RMSE was the highest at 38.796 m. This can be explained using Figure 5.9. Here it can be seen that the radar produces many detections at short range at approximately 90-95 seconds. In this time period the drone is at its maximal range however. So the RMSE is increased substantially due to these false detections.

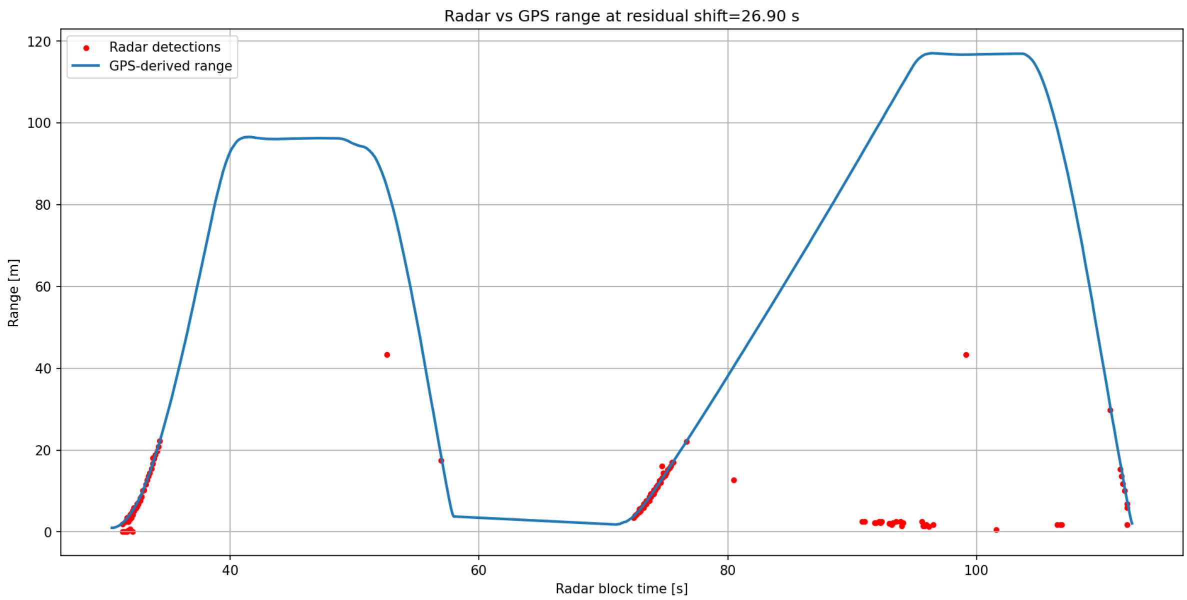
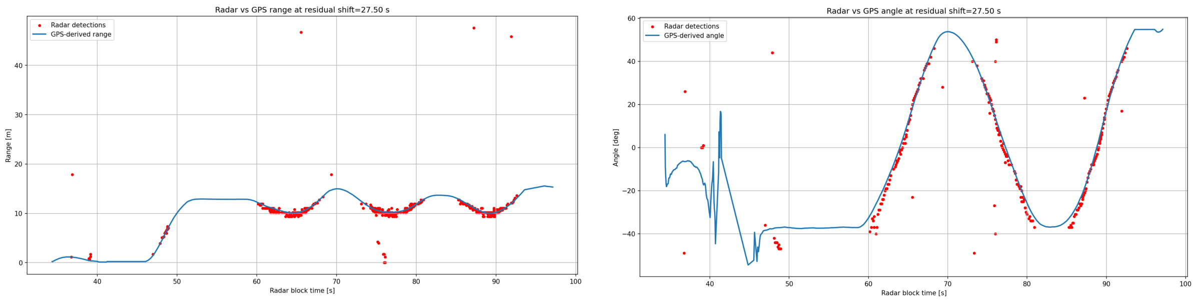


Figure 5.9: Comparison of the radar detections with the GPS derived range for Flight 3, using MVDR.

Figure 5.10 shows the comparison between the radar detections and the GPS derived range and angle for Flight 2. It can be clearly seen that the radar detections mostly follow the GPS derived range and angle. Similar figures for Flight 3 and 6 can be found in Appendix C.4.



(a) Range estimates.

(b) Angle estimates.

Figure 5.10: Comparison of the radar detections with the GPS-derived range and angle for Flight 2. The isolated radar points far from the GPS trajectory correspond to false detections.

Fixed-threshold comparison. The previously presented results used a separately optimized argmax threshold for each flight. Here the threshold will be fixed for both MUSIC as well as for MVDR. The median of the flight-specific optimal argmax thresholds have been used, which was an Argmax threshold of 11 dB for MVDR and 20 dB for MUSIC. The results are shown in Table 5.7.

Table 5.7: Range and angle RMSE obtained using fixed Argmax thresholds across all flights. The threshold was set to 11 dB for MVDR and 20 dB for MUSIC.

Flight	DoA algorithm	Range RMSE [m]	Angle RMSE [°]
Flight 2	MVDR	6.723	15.680
	MUSIC	3.318	9.109
Flight 3	MVDR	38.796	18.854
	MUSIC	7.359	8.586
Flight 6	MVDR	5.178	11.889
	MUSIC	4.888	5.905

The results demonstrate that the same trend persists: MUSIC produces a lower range and angle RMSE than MVDR. These RD and DoA parameters will now be used on the validation flight. Also notice that none of these configurations seem to achieve a range RMSE below 2.15 m, This is mainly due to the false detection increasing the RMSE. This is why this does not necessarily contradict the requirements mentioned in the PoR (2.2), since the two metrics are not directly comparable. It can also be seen that all the MUSIC configurations satisfy the maximum angle RMSE of 10°, while none of the MVDR configurations satisfy it.

Flight 4 validation. Due to time constraints only Flight 4 could be used as a validation flight. In this flight, the drone travels approximately 120 m at an angle of 40° relative to the radar, returns, and then flies approximately 15 m at an angle of −40°. Here the new pipeline with the following configuration will be used: $N_{\text{DOPPLER}} = 512$, RD parameters: GO-CFAR, with $n_{guard} = 2$, $n_{train} = 6$ and threshold factor = 20. For the DoA stage MUSIC + argmax is used with a minimum threshold of 20 dB. This is the same configuration that has been used in the preceding analysis, and none of the parameters have been retuned for Flight 4. The range dependent results using this configuration on flight 4 are shown in Table 5.8. It can be seen that the same trend as in the other flights can be seen here: performance drops sharply for longer ranges.

Table 5.8: Flight 4 validation results for the fixed GO-CFAR range–Doppler configuration. The parameters were $N_{\mathrm{Doppler}} = 512$, $g = 2$, $n = 6$, and $\alpha = 20$. The range intervals are determined using the GPS-derived target range.

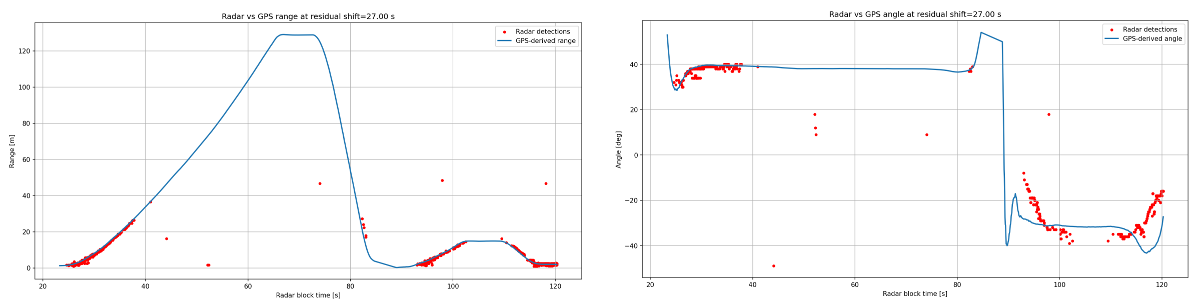
Range interval	Recall	FA/block	F_2
0–25 m	0.711	2.643	0.483
25–50 m	0.351	0.230	0.383
50–75 m	0.042	0.681	0.044
75–100 m	0.000	0.333	0.000
Overall	0.519	1.894	0.404

The final range-angle detections achieved a recall of 0.346 and a $F_{0.5}$ score of 0.204, as can be seen from Table 5.9. Flight 3 is quite similar to this flight, with the main difference being is that it exclusively flew near broadside. Using the exact same configuration, it had achieved a recall of 0.187, 0.296 false alarms per block and a $F_{0.5}$ score of 0.32 as shown in Appendix A.2. So Flight 4 achieved both a higher recall as well as a higher number of false alarms, producing the lower $F_{0.5}$ score. The selected configuration transferred only partially to Flight 4: it retained a high short-range recall, but this came at the cost of many false detections, which reduced the overall $F_{0.5}$ score.

Table 5.9: Flight 4 validation result for the fixed GO-CFAR-guided DoA configuration. Background-spectrum normalization was enabled.

DoA	Selection	Parameters	Recall	FA/block	$F_{0.5}$
MUSIC	Argmax	θ_{min}=20 dB, bg	0.346	1.523	0.204

The range RMSE of Flight 4 was 5.5 m, while the angle RMSE was 10.4°. Neither of these satisfy the requirements set in the Programme of Requirements (2.2). However, from Figure 5.11a it can be seen that the range estimation seems to be quite accurate. The relatively high range RMSE value is once again due to the false detections that can be seen in the figure. So once again, these values do not by themself contradict the requirements set in the PoR. The angle estimation seems to be less accurate, as can be seen from Figure 5.11b.



(a) Range estimates.

(b) Angle estimates.

Figure 5.11: Comparison of the final radar detections with the GPS-derived range and angle for the Flight 4 validation recording. The detections were obtained using the fixed GO-CFAR-guided MUSIC configuration with background-spectrum normalization and an Argmax threshold of 20 dB.

Limitations. Several limitations should be mentioned. First of all, the simulation was simplified. Inserting the synthetic drone into a measured empty background recording would be one way to make it more realistic, as discussed in Appendix C.2. Secondly, the results obtained for both the simulated data as well as for the real data have used Q=2 for the MUSIC algorithm. This choice has been made on empirical observations, rather than a formal source-number estimation technique. This is discussed in more detail in Appendix C.3.
Furthermore, the selected configuration was only validated on one flight. It should also be noted that the RMSE values are strongly influenced by the false detections, which means that they are not directly

equivalent to the accuracy tolerances stated in the PoR. Finally, only the GO-CFAR-guided pipeline has been tested on the validation flight, while the CM-CFAR-guided pipeline could not be validated due to time constraints.

6 Conclusion

The main objective of the signal-processing subgroup was to develop and evaluate a pipeline that could detect drone returns, suppress clutter and estimate the range, radial velocity and azimuth angle of the drone within the computational requirements. These objectives have been largely met. The implemented pipeline is able to suppress clutter by using techniques such as mean subtraction and background power spectrum normalization. Multiple CFAR variants for range-Doppler detection, and DoA methods like MUSIC and MVDR, were used to produce detections containing range, radial velocity and azimuth angle. The computational requirements were also met, especially in the new RD-guided pipeline, which significantly reduced the processing time. However, the accuracy requirements were only partially satisfied in practice, since the final RMSE errors were strongly influenced by false detections. Therefore, the reported RMSE values are not directly equivalent to the accuracy tolerances stated in the PoR.

6.1. Results

For the current radar, the best-performing CFAR variant depends on the target range:

- CA-CFAR in the 0–25 m range,
- CM-CFAR in the 25–50 m range,
- SO-CFAR in the 50–75 m range, while CM-CFAR remains competitive.

The detection performance generally worsens for longer ranges.

The tunable parameters influence performance in distinct ways:

- For the spatial CFAR variants no single guard-cell and training-cell setting performed best across all ranges and scenes.
- Raising N_{DOPPLER} increases recall at the cost of more false detections.
- Raising the threshold factor α reduces false detections at the cost of recall.

MUSIC achieves a higher $F_{0.5}$ score than MVDR, but requires the number of sources to be known beforehand.

Estimating DoA only at the cells detected on the range-Doppler map, is around 80 to 120 times more computationally efficient depending on the algorithm used, but in general produces worse $F_{0.5}$ scores.

6.2. Future work

Several directions remain open for further investigation:

- Evaluating alternative CFAR frame shapes, in particular a circular frame
- Weighted-window CFAR [22]

- Sweeping a wider range of parameter values
- Sweeping additional parameters, such as those of DBSCAN
- A 3D CFAR implementation [23]
- Calibrated injection of synthetic targets into measured background.
- MIMO evaluation.
- Choosing the number of sources needed for MUSIC non-empirically.
- Validating the CM-CFAR guided configuration.
- Validation of the selected configurations on additional independent flight recordings.

References

[1] Z. Cao, J. Li, C. Song, Z. Xu, and X. Wang, "A novel cfar algorithm for multi-target detection with fmcw radar," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6. DOI: 10.1109/GLOBECOM42002.2020.9322140.

[2] H. Zhang et al., "Robust fmcw radar clutter suppression for uav detection via learning physics-informed priors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 62, pp. 6180–6195, 2026. DOI: 10.1109/TAES.2026.3664357.

[3] T. Zhou and D. Tao, "GoDec: Randomized lowrank & sparse matrix decomposition in noisy case," in *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011, pp. 33–40.

[4] T. Holton, *Digital Signal Processing: Principles and Applications*. Cambridge University Press, 2021.

[5] F. D. Enggar, A. M. Muthiah, O. D. Winarko, O. N. Samijayani, and S. Rahmatia, "Performance comparison of various windowing on fmcw radar signal processing," in *2016 International Symposium on Electronics and Smart Devices (ISESD)*, 2016, pp. 326–330. DOI: 10.1109/ISESD.2016.7886743.

[6] H. Yang and J. Wang, "Sidelobe suppression of sar images by spectrum shaping," in *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, 2022, pp. 1668–1671. DOI: 10.1109/IGARSS46834.2022.9884635.

[7] W. Wang, R. Wang, R. Jiang, H. Yang, and X. Wang, "Modified reference window for two-dimensional CFAR in radar target detection," *The Journal of Engineering*, vol. 2019, no. 21, pp. 7924–7927, 2019. DOI: 10.1049/joe.2019.0687.

[8] H. Rohling, "Radar cfar thresholding in clutter and multiple target situations," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-19, no. 4, pp. 608–621, Jul. 1983. DOI: 10.1109/TAES.1983.309350. [Online]. Available: https://ece.iisc.ac.in/~cmurthy/E1_244/Slides/Rohling.pdf.

[9] M. Xiangwei and M. Yuan, "Modified rank sum nonparametric cfar to combat clutter edge," *EURASIP Journal on Advances in Signal Processing*, vol. 2023, no. 75, Jun. 2023. DOI: 10.1186/s13634-023-01032-z. [Online]. Available: https://link.springer.com/article/10.1186/s13634-023-01032-z.

[10] A. Safa et al., "A low-complexity radar detector outperforming os-cfar for indoor drone obstacle avoidance," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 9162–9175, 2021. DOI: 10.1109/JSTARS.2021.3107686.

[11] D. Kurniawan, O. Heriana, T. Praludi, and R. Sariningrum, "Data acquisition and signal processing on fmcw navigation radar system," in *2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, 2017, pp. 47–52. DOI: 10.1109/ICRAMET.2017.8253143.

[12] R. Nitzberg, "Clutter map cfar analysis," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-22, no. 4, pp. 419–421, 1986. DOI: 10.1109/TAES.1986.310777.

[13] R. D. Yates and D. J. Goodman, *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers*, 3rd ed. John Wiley & Sons, 2014, ISBN: 978-1-118-32456-1.

[14] S. Korlapati and R. Nikandish, *Adaptive mimo radar architecture for energy-efficient wireless sensing in the d-band*, 2025. arXiv: 2309.17110 [eess.SP]. [Online]. Available: https://arxiv.org/abs/2309.17110.

[15] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*. New York, NY, USA: John Wiley & Sons, 2002, ISBN: 978-0-471-09390-9.

[16] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986. DOI: 10.1109/TAP.1986.1143830.

[17] S. U. Pillai and B. H. Kwon, "Forward/backward spatial smoothing techniques for coherent signal identification," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 1, 1989. DOI: 10.1109/29.17496.

[18] T. J. Shan, M. Wax, and T. Kailath, "On spatial smoothing for direction-of-arrival estimation of coherent signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 4, pp. 806–811, 1985. DOI: 10.1109/TASSP.1985.1164649.

[19] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, AAAI Press, 1996, pp. 226–231.

[20] D. C. de Ramos et al., "Evaluation of cluster algorithms for radar-based object recognition in autonomous and assisted driving," *Sensors*, vol. 24, no. 22, p. 7219, 2024. DOI: 10.3390/s24227219.

[21] C. Waldschmidt, C. Bonfert, and T. Grebner, "Signal processing principles," in *Millimeter Wave Radar: Hardware and Signal Processing*, Springer, 2025, pp. 83–115. DOI: 10.1007/978-3-031-89118-2_5.

[22] M. Xiangwei, G. Jian, and H. You, "Performance analysis of the weighted window cfar algorithms [radar signal processing]," in *2003 Proceedings of the International Conference on Radar (IEEE Cat. No.03EX695)*, 2003, pp. 354–357. DOI: 10.1109/RADAR.2003.1278766.

[23] S. Wang and R. Herschel, "Fast 3d-cfar for drone detection with mimo radars," in *2021 18th European Radar Conference (EuRAD)*, 2022, pp. 209–212. DOI: 10.23919/EuRAD50154.2022.9784486.

A

Test Flights

Sweeping Parameters

For real data recordings:

CA, SO, GO, OS and CM-CFAR were tested with

- Number of Guard Cells $N_{\text{GUARD}} = [2, 3, 4]$
- Number of Train Cells $N_{\text{TRAIN}} = [6, 10, 14]$
- Threshold $\alpha = [10, 20, 30, 40, 50, 60]$
- K^{th} Order Fraction $k_{\text{FRAC}} = [0.55, 0.75, 0.95]$
- Learning Rate $\beta = [5 \times 10^{-5}, 5 \times 10^{-4}, 5 \times 10^{-3}, 5 \times 10^{-2}, 0]$

MVDR and MUSIC, both the CFAR and ARGMAX variants, were tested with both the old and the new pipeline, using

- $N_{\text{GUARD}} = [1, 2, 3, 4, 6]$ for the CA, GO, SO - CFAR variants
- $N_{\text{TRAIN}} = [4, 6, 8, 10, 12, 16]$ for the CA, GO, SO - CFAR variants
- $\alpha = [1.5, 2.5, 4, 5, 7.5, 10]$
- Minimum dB for Argmax = $[-\infty, 1, 2.5, 4, 5, 6, 7.5, 9, 10, 11, 12.5, 14, 15, 17.5, 20, 22.5, 25]$

All of these parameters were tested with (except for CM, due to the specific implementation making it very difficult to align detection times with ground-truth times)

- $N_{\text{DOPPLER}} = [64, 256, 512]$
- $N_{\text{CM-DOPPLER}} = [50, 100, 250, 500]$

MUSIC was performed with Q=2. See Appendix C.3 for more details.

General Observations

The radar can only capture radial velocities of up to ±14.8m/s. If the drone travels faster than this, it won't be detected on the RD Map, and subsequently not in the new DoA pipeline either.

The farther the drone travels, the lower the returns on the RD Map.

Drone returns are weak at large off-boresight angles. The field of view is therefore restricted to ±50°, beyond which detections are considered unreliable.

A.1. Test Flight 2: Side by Side

Flight Description

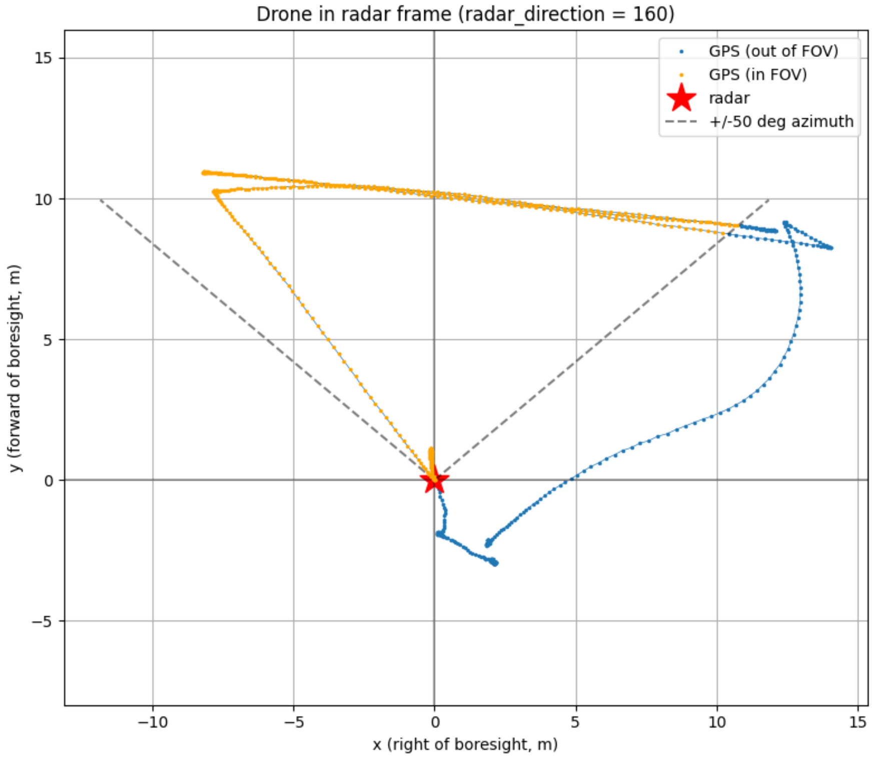


Figure A.1: Bird's-eye view of Test Flight 2

The drone starts behind the radar and flies directly over it for a couple of seconds, so the GPS position of the radar can be obtained. Then it flies out to the left diagonally, then to the right, then to the left, then to the right again, and goes back behind the radar.

Video Recording of the flight
Video Recording of RD Map
Video recording of RA Map

Results RD-CFAR

Table A.1: Best F_2 configuration per CFAR variant and Doppler resolution (whole range).

N_d	Algorithm	Guard	Train	k	Threshold	Recall	FA/block	F_2
64	CA	2	14	–	10	0.656	0.243	0.670
64	GO	2	14	–	10	0.624	0.160	0.653
64	OS	2	6	0.55	10	0.644	0.429	0.635
64	SO	4	14	–	20	0.611	0.273	0.626
256	CA	2	14	–	30	0.725	1.005	0.633
256	GO	4	14	–	20	0.670	0.756	0.617
256	OS	2	6	0.75	50	0.694	1.080	0.601
256	SO	2	14	–	50	0.797	1.862	0.598
512	CA	2	14	–	40	0.691	1.586	0.551
512	GO	2	6	–	40	0.523	0.291	0.543
512	OS	2	6	0.75	50	0.775	2.428	0.538
512	SO	2	6	–	50	0.797	4.259	0.440

Table A.2: Top three F_2 configurations for each CFAR variant (whole range).

N_d	Algorithm	Guard	Train	k	Threshold	Recall	FA/block	F_2
64	CA	2	14	–	10	0.656	0.243	0.670
64	CA	2	10	–	10	0.648	0.203	0.668
64	CA	3	10	–	10	0.649	0.239	0.664
64	GO	2	14	–	10	0.624	0.160	0.653
64	GO	3	14	–	10	0.621	0.178	0.647
64	GO	2	10	–	10	0.613	0.128	0.647
64	OS	2	6	0.55	10	0.644	0.429	0.635
64	OS	4	10	0.75	10	0.623	0.292	0.634
64	OS	3	10	0.75	10	0.622	0.283	0.634
64	SO	4	14	–	20	0.611	0.273	0.626
64	SO	4	10	–	20	0.606	0.258	0.623
64	SO	4	6	–	20	0.603	0.243	0.622

Table A.3: Top three F_2 configurations for CM-CFAR over all Doppler resolutions for Flight 2 (whole range).

N_d	Init. blocks	Learning rate	Threshold	Recall	FA/block	F_2
250	911	5×10^{-5}	20	0.633	0.591	0.606
250	911	0	20	0.648	0.714	0.604
500	455	5×10^{-4}	20	0.703	1.152	0.601

Table A.4: Top three F_2 configurations for CM-CFAR at each Doppler resolution for Flight 2 over the whole range.

N_d	Init. blocks	Learning rate	Threshold	Recall	FA/block	F_2
50	4558	0	10	0.320	0.312	0.346
50	4558	0	12.5	0.313	0.243	0.343
50	4558	0	15	0.308	0.199	0.342
100	2279	0	15	0.554	0.292	0.572
100	2279	0	20	0.538	0.176	0.571
100	2279	0	12.5	0.563	0.404	0.566
250	911	5×10^{-5}	20	0.633	0.591	0.606
250	911	0	20	0.648	0.714	0.604
250	911	5×10^{-5}	15	0.661	0.915	0.593
500	455	5×10^{-4}	20	0.703	1.152	0.601
500	455	5×10^{-5}	20	0.721	1.420	0.587
500	455	0	20	0.719	1.585	0.570

Table A.5: Best F_2 configuration for each CM-CFAR learning rate for Flight 2 over the whole range.

Learning rate	N_d	Init. blocks	Threshold	Recall	FA/block	F_2
0	250	911	20	0.648	0.714	0.604
5×10^{-5}	250	911	20	0.633	0.591	0.606
5×10^{-4}	500	455	20	0.703	1.152	0.601
5×10^{-3}	500	455	20	0.596	0.679	0.565
5×10^{-2}	500	455	12.5	0.571	0.793	0.533

Discussion RD-CFAR

Unlike the other two flights, the drone stays within the 0-25m range during the whole duration of the recording. The returns are stronger, which allows for recall to be high, up to 0.8.

CA-CFAR performs the best at every $N_{DOPPLER}$ (Table A.1). GO, SO and OS-CFAR follow closely behind, but do not achieve the highest F_2 scores. SO-CFAR generally obtains higher false detections due to it being more sensitive. GO-CFAR has a lower number of false detections, but also a lower recall due to it using a more conservative threshold. CM-CFAR on the other hand does not obtain the highest F_2 scores. The performance does however improve as $N_{DOPPLER}$ is increased, so much so that it outperforms the spatial CFAR methods at higher $N_{DOPPLER}$. The guard and training cell counts do not seem to have much effect on the parameters, compared to the effects of Doppler resolution and threshold factor (Table A.2). There it can be seen that for a fixed CFAR variant and threshold factor, varying n_{guard} and n_{train} resulted in only minor changes in the F_2 score.

Results DoA

Table A.6: Best $F_{0.5}$ DoA configuration per algorithm for the old pipeline using the same GO-CFAR parameters for the RD detections (Flight 2), followed by the three next-best configurations.

DoA	CFAR	Parameters	Recall	FA/block	$F_{0.5}$
MVDR	GO-CFAR	g=6, n=10, α=10, bg	0.455	0.770	0.385
MVDR	GO-CFAR	g=6, n=8, α=10, bg	0.453	0.766	0.385
MUSIC	GO-CFAR	g=6, n=16, α=10, bg	0.507	0.759	0.418
MUSIC	GO-CFAR	g=6, n=10, α=10, bg	0.502	0.761	0.415
MUSIC	CA-CFAR	g=6, n=16, α=10, bg	0.527	0.813	0.414
MUSIC	GO-CFAR	g=6, n=8, α=10, bg	0.502	0.766	0.414
MUSIC	CA-CFAR	g=4, n=16, α=10, bg	0.527	0.836	0.409

Table A.7: Best $F_{0.5}$ DoA configuration per algorithm and selection method for the GO-CFAR-guided pipeline (Flight 2), followed by the three next-best configurations. $F_{0.5}$ weights precision over recall to penalize false detections. All listed configurations use background-spectrum normalization.

DoA	Selection	Parameters	Recall	FA/block	$F_{0.5}$
MUSIC	Argmax	θ_{min}=20 dB,bg	0.441	0.624	0.420
MVDR	Argmax	θ_{min}=12.5 dB,bg	0.423	0.644	0.402
MUSIC	CA-CFAR	g=2, n=6, α=10, bg	0.300	0.586	0.330
MVDR	CA-CFAR	g=1, n=12, α=4, bg	0.203	0.462	0.277
MUSIC	Argmax	θ_{min}=17.5 dB, bg	0.457	0.673	0.414
MUSIC	Argmax	θ_{min}=22.5 dB ,bg	0.412	0.599	0.408
MUSIC	Argmax	θ_{min}=15 dB ,bg	0.473	0.736	0.405

Table A.8: Best $F_{0.5}$ DoA configuration per algorithm and selection method for the CM-CFAR-guided pipeline (Flight 2), followed by the three next-best configurations. All listed configurations use background-spectrum normalization.

DoA	Selection	Parameters	Recall	FA/block	$F_{0.5}$
MUSIC	SO-CFAR	g=1, n=10, α=7.5	0.580	1.187	0.360
MUSIC	Argmax	θ_{min}=22.5 dB	0.404	0.776	0.353
MVDR	SO-CFAR	g=4, n=16, α=7.5	0.626	1.446	0.337
MVDR	Argmax	θ_{min}=5 dB	0.604	1.404	0.334
MUSIC	SO-CFAR	g=2, n=6, α=7.5	0.578	1.182	0.359
MUSIC	SO-CFAR	g=2, n=8, α=10	0.567	1.176	0.356
MUSIC	SO-CFAR	g=2, n=6, α=10	0.516	1.055	0.354

Discussion DoA

Since the drone is close to the radar the entire flight duration, the DoA algorithms can more reliably determine the azimuth angle of the drone.

Old Pipeline results In the results obtained using the old pipeline, it can be seen that MUSIC outperforms MVDR here. It achieves a slightly higher recall than MVDR while having a similar false detection rate. Consequently, the $F_{0.5}$ score obtained by MUSIC is higher. It should also be noted that all the best results (based on the $F_{0.5}$ score) are obtained using the aforementioned background ratio technique

Guided RD results Guided by GO-CFAR detections (Table A.7), the most robust DoA algorithm is MUSIC + argmax selection (selecting the angle bin with the strongest return). The CFAR methods lead to a lower recall, while not causing a proportional decrease in the number of false detections.

On the other hand, guided by CM-CFAR (Table A.8), MUSIC outperforms MVDR again. The SO-CFAR variant barely outperforms the argmax, since it is more sensitive and achieves a higher recall, which boosts its $F_{0.5}$ score. The GO-CFAR guided pipeline performs better than the CM-CFAR guided pipeline in this case. This demonstrates that the choice of the chosen RD configuration affects the obtained angle estimation results in the new pipeline.

Comparison of the pipelines The best results (based on the $F_{0.5}$ score) obtained using the old pipeline are very similar to the best results obtained using the rd-guided pipeline. The old pipeline's best pick, being the MUSIC algorithm with GO-CFAR, achieves a higher recall than the new pipeline's best algorithm. This, however, comes at the cost of a slightly higher amount of false detections. However, it should also be noted that the 2D CFAR used in the old pipeline outperforms the 1D CFAR used in the new pipeline. The highest achieved $F_{0.5}$ score achieved using the 1D cfar was in the CM-CFAR guided pipeline, where the SO-CFAR was used.

A.2. Test Flight 3: Max Range

Flight Description and Discussion

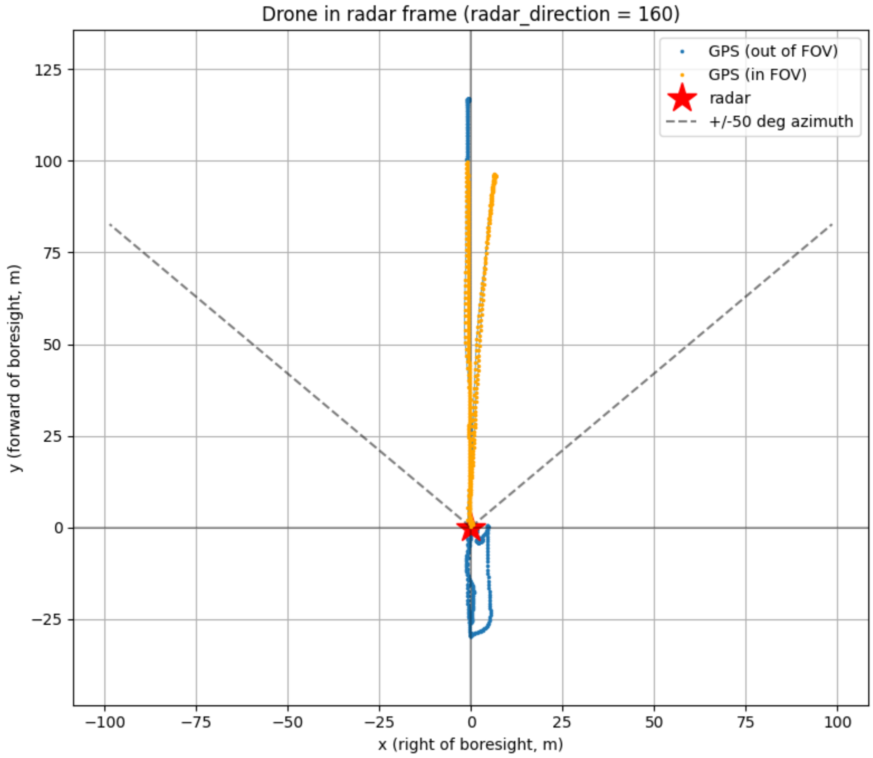


Figure A.2: Bird's-eye view of Test Flight 3

The drone starts behind the radar, flies out to ~120m with a high velocity before coming back behind the radar, then flies out to ~80m at a constant velocity of 5m/s and comes back behind the radar again. All while staying at ~ 0° azimuth.

The drone disappears from the RD Map once its velocity exceeds the maximal velocity that can be captured by the radar. The farther out it is, the lower the signature, and thus the fewer detections occur. Due to this, there is a ceiling on how high the recall of the CFAR-algorithms can get. The returns on the RD Map are simply not strong enough for any of the algorithms we use to faithfully detect them without an extremely high number of false detections.

Video Recording of the flight on YouTube
Video Recording of RD Map
Video Recording of RA Map

Results RD-CFAR

Table A.9: Best F_2 configuration per CFAR variant and Doppler resolution (whole range).

N_d	Algorithm	Guard	Train	k	Threshold	Recall	FA/block	F_2
64	CA	2	6	–	10	0.183	0.140	0.212
64	SO	2	6	–	10	0.190	0.361	0.209
64	OS	2	6	0.55	10	0.181	0.248	0.204
64	GO	3	6	–	10	0.168	0.112	0.197
256	GO	2	14	–	10	0.266	1.350	0.237
256	OS	2	6	0.95	10	0.219	1.056	0.207
256	CA	2	6	–	30	0.203	1.166	0.189
256	SO	3	10	–	40	0.203	2.209	0.158
512	GO	2	6	–	20	0.266	0.805	0.263
512	OS	2	6	0.95	10	0.259	1.127	0.240
512	CA	2	6	–	30	0.243	1.953	0.196
512	SO	3	10	–	10	0.332	6.021	0.161

Table A.10: Top three F_2 configurations for each CFAR variant (whole range).

N_d	Algorithm	Guard	Train	k	Threshold	Recall	FA/block	F_2
64	CA	2	6	–	10	0.183	0.140	0.212
64	CA	3	6	–	10	0.182	0.141	0.211
64	CA	2	10	–	10	0.181	0.131	0.210
512	GO	2	6	–	20	0.266	0.805	0.263
512	GO	2	10	–	20	0.269	0.897	0.260
512	GO	2	14	–	20	0.266	0.942	0.256
512	OS	2	6	0.95	10	0.259	1.127	0.240
512	OS	2	10	0.95	10	0.274	1.821	0.225
512	OS	3	6	0.95	10	0.272	1.887	0.221
64	SO	2	6	–	10	0.190	0.361	0.209
64	SO	4	6	–	10	0.188	0.361	0.207
64	SO	2	10	–	10	0.188	0.371	0.206

Table A.11: Best F_2 CFAR configuration per range interval.

Range	N_d	Algorithm	Guard	Train	Threshold	Recall	FA/block	F_2
0–25 m	64	CA	2	14	10	0.668	0.216	0.683
25–50 m	512	GO	2	10	10	0.721	1.508	0.579
50–75 m	512	SO	2	6	10	0.125	5.146	0.067
75–100 m	512	SO	3	6	10	0.023	5.343	0.012

Results RD-CM-CFAR

Table A.12: Top three F_2 CM-CFAR configurations for each Doppler resolution for Flight 3 (whole range).

N_d	Init. blocks	Learning rate	Threshold	Recall	FA/block	F_2
50	4558	5×10^{-5}	5	0.198	0.234	0.223
50	4558	0	5	0.200	0.290	0.223
50	4558	5×10^{-4}	5	0.191	0.240	0.216
100	2279	5×10^{-5}	5	0.247	0.582	0.255
100	2279	0	5	0.245	0.637	0.250
100	2279	5×10^{-4}	5	0.240	0.550	0.250
250	911	5×10^{-4}	7.5	0.285	0.875	0.276
250	911	5×10^{-4}	5	0.317	1.450	0.275
250	911	5×10^{-5}	7.5	0.287	0.951	0.274
500	455	5×10^{-4}	10	0.298	1.132	0.274
500	455	5×10^{-4}	12.5	0.285	0.920	0.274
500	455	5×10^{-3}	5	0.389	2.733	0.273

Table A.13: Best F_2 CM-CFAR configuration for each learning rate for Flight 3 (whole range).

Learning rate	N_d	Init. blocks	Threshold	Recall	FA/block	F_2
0	250	911	7.5	0.287	0.969	0.273
5×10^{-5}	250	911	7.5	0.287	0.951	0.274
5×10^{-4}	250	911	7.5	0.285	0.875	0.276
5×10^{-3}	500	455	5	0.389	2.733	0.273
5×10^{-2}	500	455	5	0.334	2.415	0.248

Table A.14: Best F_2 CM-CFAR configuration per range interval for Flight 3.

Range	N_d	Init. blocks	Learning rate	Threshold	Recall	FA/block	F_2
0–25 m	50	4558	0	7.5	0.628	0.538	0.608
25–50 m	500	455	0	7.5	0.762	0.635	0.706
50–75 m	500	455	5×10^{-3}	5	0.375	0.479	0.386
75–100 m	500	455	5×10^{-3}	5	0.056	0.972	0.056

Discussion RD-CFAR

If the performance of the algorithms is measured over the entire range, CM-CFAR performs consistently the best. It is followed behind by GO and OS.

If the performance of the algorithms is measured over the different ranges, CA-CFAR performs the best in the 0-25m range as can be seen from Table A.11. The F_2 score of both approaches drops significantly once the drone is farther than 50m, but CM-CFAR degrades much more slowly and retains a F_2 score of 0.386 in the 50-75 m interval. After 75 m, however detection performance collapses due to a near zero recall across the board. This is supported by Tables A.14 and A.11.

Results DoA

Table A.15: Best $F_{0.5}$ DoA configuration per algorithm for the old pipeline using the same GO-CFAR parameters for the RD detections (Flight 3), followed by the three next-best configurations. All listed configurations use background-spectrum normalization.

DoA	CFAR	Parameters	Recall	FA/block	$F_{0.5}$
MVDR	SO-CFAR	g=6, n=16, α=10, bg	0.158	0.272	0.291
MVDR	SO-CFAR	g=6, n=10, α=10, bg	0.156	0.298	0.276
MUSIC	GO-CFAR	g=6, n=16, α=10, bg	0.222	0.298	0.360
MUSIC	GO-CFAR	g=6, n=10, α=10, bg	0.219	0.309	0.352
MUSIC	GO-CFAR	g=1, n=16, α=7.5, bg	0.179	0.211	0.350
MUSIC	GO-CFAR	g=6, n=10, α=7.5, bg	0.224	0.327	0.350
MUSIC	GO-CFAR	g=6, n=16, α=7.5, bg	0.222	0.330	0.346

Table A.16: Best $F_{0.5}$ DoA configuration per algorithm and selection method for the GO-CFAR-guided pipeline (Flight 3), followed by the three next-best configurations. All listed configurations use background-spectrum normalization.

DoA	Selection	Parameters	Recall	FA/block	$F_{0.5}$
MUSIC	Argmax	θ_{min}=25 dB	0.161	0.190	0.335
MVDR	Argmax	θ_{min}=11 dB	0.132	0.259	0.257
MUSIC	GO-CFAR	g=4, n=16, α=10, bg	0.145	0.417	0.223
MVDR	SO-CFAR	g=6, n=16, α=10, bg	0.113	0.435	0.178
MUSIC	Argmax	θ_{min}=22.5 dB	0.179	0.261	0.325
MUSIC	Argmax	θ_{min}=17.5 dB	0.198	0.317	0.324
MUSIC	Argmax	θ_{min}=20 dB	0.187	0.296	0.320

Table A.17: Best $F_{0.5}$ DoA configuration per algorithm and selection method for the CM-CFAR-guided pipeline (Flight 3), followed by the three next-best configurations. All listed configurations use background-spectrum normalization.

DoA	Selection	Parameters	Recall	FA/block	$F_{0.5}$
MUSIC	Argmax	θ_{min}=9 dB	0.227	0.550	0.277
MVDR	Argmax	θ_{min}=2.5 dB	0.222	0.633	0.251
MUSIC	GO-CFAR	g=1, n=16, α=7.5, bg	0.147	0.344	0.248
MVDR	GO-CFAR	g=6, n=16, α=2.5, bg	0.155	0.649	0.184
MUSIC	Argmax	θ_{min}=12.5 dB	0.217	0.522	0.274
MUSIC	Argmax	θ_{min}=10 dB	0.225	0.550	0.274
MUSIC	Argmax	θ_{min}=11 dB	0.220	0.532	0.274

Discussion DoA

Old Pipeline results From table A.15 it can be seen that the best results using MVDR is obtained by using the SO-CFAR algorithm. For MUSIC the best results are obtained using the GO-CFAR. Once again, the results obtained using MUSIC are better than the results obtained using MVDR.

RD-Guided results For this flight, MUSIC + Argmax selection method once again clearly performs the best as can be seen from table A.16, The CM-RD guided angle estimation produces a higher recall as well as a larger number of false detections compared to the GO-RD guided angle estimation. The GO-RD Stages achieves the highest $F_{0.5}$ scores.

Comparison of the pipelines The results obtained using the old pipeline are slightly better than the results obtained using the new pipeline. The difference between the top picks is about 0.025 $F_{0.5}$ points. It can be seen that the old pipeline's best pick once again achieves a higher recall, and also has a higher false alarm rate.

A.3. Test Flight 6: Hovering at Different Ranges

Flight Description and Discussion

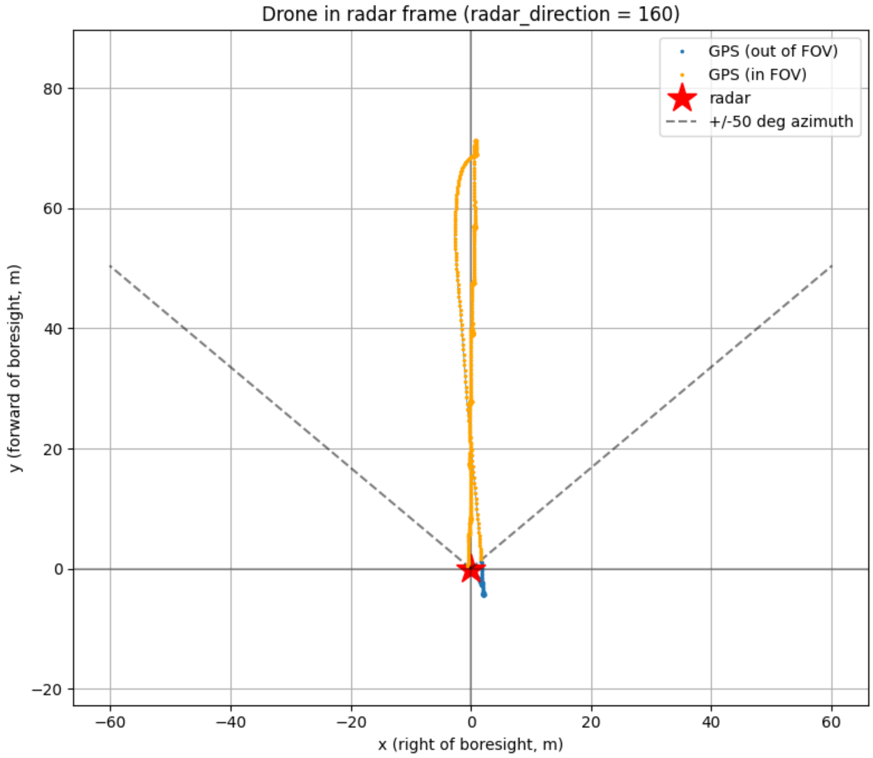


Figure A.3: Bird's-eye view of Test Flight 6

The drone starts behind the radar, flies from 10m to 70m, each time stopping for ~ 2s, and flies back at the end of the recording. As seen in the video, there are a couple of birds in the scene, although they are far away and most likely didn't contribute to any clutter in the RD Map.

Video Recording of flight YouTube
Video Recording of RD Map
Video Recording of RA Map
Video Recording of RD Map GODEC

Results RD-CFAR

Table A.18: Best F_2 configuration per CFAR variant and Doppler resolution (whole range).

N_d	Algorithm	Guard	Train	k	Threshold	Recall	FA/block	F_2
64	CA	2	10	–	10	0.191	0.319	0.212
64	SO	4	14	–	10	0.205	0.902	0.201
64	OS	2	6	0.55	10	0.194	0.645	0.201
64	GO	2	10	–	10	0.169	0.196	0.193
256	SO	2	6	–	20	0.409	4.127	0.239
256	OS	2	6	0.55	20	0.382	3.727	0.235
256	CA	3	6	–	10	0.409	4.488	0.230
256	GO	2	6	–	10	0.269	1.883	0.218
512	GO	2	6	–	10	0.340	2.312	0.255
512	CA	2	6	–	30	0.300	1.962	0.240
512	SO	2	10	–	10	0.545	6.876	0.239
512	OS	2	6	0.95	10	0.268	1.366	0.238

Table A.19: Top three F_2 configurations for each CFAR variant (whole range).

N_d	Algorithm	Guard	Train	k	Threshold	Recall	FA/block	F_2
512	CA	2	6	–	30	0.300	1.962	0.240
512	CA	2	6	–	20	0.360	3.219	0.238
512	CA	3	6	–	10	0.485	5.781	0.236
512	GO	2	6	–	10	0.340	2.312	0.255
512	GO	2	6	–	20	0.254	0.851	0.248
512	GO	3	6	–	10	0.350	2.936	0.240
512	OS	2	6	0.95	10	0.268	1.366	0.238
512	OS	2	6	0.75	20	0.437	4.773	0.237
256	OS	2	6	0.55	20	0.382	3.727	0.235
256	SO	2	6	–	20	0.409	4.127	0.239
512	SO	2	10	–	10	0.545	6.876	0.239
256	SO	2	6	–	30	0.369	3.418	0.237

Table A.20: Best F_2 CFAR configuration per range interval.

Range	N_d	Algorithm	Guard	Train	Threshold	Recall	FA/block	F_2
0–25 m	256	CA	2	14	30	0.648	1.729	0.508
25–50 m	512	SO	3	10	10	0.455	3.423	0.289
50–75 m	512	SO	2	6	10	0.381	4.986	0.203

Results RD-CM-CFAR

Table A.21: Top three F_2 CM-CFAR configurations per Doppler resolution for Flight 6 (whole range).

N_d	Init. blocks	Learning rate	Threshold	Recall	FA/block	F_2
50	4558	5×10^{-5}	5	0.150	0.398	0.165
50	4558	0	5	0.149	0.454	0.162
50	4558	5×10^{-4}	5	0.140	0.318	0.157
100	2279	5×10^{-5}	5	0.234	0.903	0.228
100	2279	5×10^{-4}	5	0.225	0.788	0.224
100	2279	0	5	0.232	0.964	0.223
250	911	5×10^{-5}	5	0.351	2.135	0.271
250	911	5×10^{-4}	5	0.341	1.962	0.271
250	911	5×10^{-3}	5	0.323	1.658	0.270
500	455	5×10^{-2}	10	0.318	1.044	0.296
500	455	5×10^{-2}	12.5	0.296	0.792	0.291
500	455	5×10^{-3}	10	0.340	1.541	0.289

Table A.22: Best F_2 CM-CFAR configuration per learning rate for Flight 6 (whole range).

Learning rate	N_d	Init. blocks	Threshold	Recall	FA/block	F_2
0	500	455	10	0.356	2.073	0.277
5×10^{-5}	500	455	10	0.356	2.036	0.279
5×10^{-4}	500	455	10	0.353	1.889	0.283
5×10^{-3}	500	455	10	0.340	1.541	0.289
5×10^{-2}	500	455	10	0.318	1.044	0.296

Table A.23: Best F_2 CM-CFAR configuration per range interval for Flight 6.

Range	N_d	Init. blocks	Learning rate	Threshold	Recall	FA/block	F_2
0–25 m	500	455	5×10^{-2}	20	0.524	1.097	0.466
25–50 m	500	455	5×10^{-4}	7.5	0.394	0.765	0.382
50–75 m	500	455	5×10^{-3}	5	0.183	0.844	0.182

Table A.24: Performance of the fixed CM-CFAR configuration across all measured flights. The configuration uses $N_d = 500$, 455 initialization blocks, a learning rate of 5×10^{-4}, and a threshold factor of 15.

Flight	Annotated blocks	Correct	Missed	False	Recall	FA/block	F_2
Flight 2	424	299	125	771	0.705	1.818	0.540
Flight 3	386	107	279	312	0.277	0.808	0.273
Flight 6	702	215	487	883	0.306	1.258	0.275

Table A.25: Whole-range RD detection performance of Candidate B across the measured flights. Candidate B uses GO-CFAR with $N_d = 512$, two guard cells, six training cells, and a threshold factor of 20.

Flight	Recall	FA/block	F_2
Flight 2	0.606	1.169	0.525
Flight 3	0.266	0.805	0.263
Flight 6	0.254	0.851	0.248

Table A.26: Whole-range RD detection performance of Candidate C across the measured flights. Candidate C uses CA-CFAR with N_d = 512, two guard cells, six training cells, and a threshold factor of 30.

Flight	Recall	FA/block	F_2
Flight 2	0.680	2.155	0.498
Flight 3	0.243	1.953	0.196
Flight 6	0.300	1.962	0.240

Discussion RD-CFAR

As was the case in the other flights, it can be seen that the spatial CFAR algorithms produce better F_2 results in the 0-25 m interval compared to the CM-CFAR. For the 25-50 m interval it can be seen that the best CM-CFAR configuration outperforms the best spatial CFAR configuration. The F_2 score is 0.093 points higher. While for the 50-75 m range the sensitive SO-CFAR configuration outperforms the CM-CFAR. Across all evaluated detectors, CM-CFAR reaches the highest whole-range score of 0.296 (Table A.21), while GO-CFAR is the best spatial variant with 0.255 (Table A.18).

Results DoA

Table A.27: Best $F_{0.5}$ DoA configuration per algorithm for the old pipeline using the same GO-CFAR parameters for the RD detections (Flight 6), followed by the three next-best configurations. All listed configurations use background-spectrum normalization.

DoA	CFAR	Parameters	Recall	FA/block	$F_{0.5}$
MVDR	GO-CFAR	g=6, n=16, α=10, bg	0.217	0.445	0.298
MVDR	GO-CFAR	g=6, n=16, α=7.5, bg	0.224	0.488	0.291
MUSIC	GO-CFAR	g=6, n=8, α=10, bg	0.248	0.413	0.340
MUSIC	GO-CFAR	g=4, n=8, α=10, bg	0.246	0.410	0.340
MUSIC	GO-CFAR	g=6, n=10, α=10, bg	0.249	0.427	0.336
MUSIC	GO-CFAR	g=6, n=16, α=10, bg	0.251	0.434	0.335
MUSIC	GO-CFAR	g=4, n=10, α=10, bg	0.246	0.424	0.335

Table A.28: Best $F_{0.5}$ DoA configuration per algorithm and selection method for the GO-CFAR-guided pipeline (Flight 6), followed by the three next-best configurations. All listed configurations use background-spectrum normalization.

DoA	Selection	Parameters	Recall	FA/block	$F_{0.5}$
MUSIC	Argmax	$\theta_{\min}$=17.5 dB	0.245	0.429	0.331
MVDR	Argmax	$\theta_{\min}$=11 dB	0.220	0.464	0.295
MUSIC	GO-CFAR	g=3, n=16, α=10, bg	0.184	0.475	0.253
MVDR	GO-CFAR	g=1, n=16, α=4, bg	0.103	0.251	0.214
MUSIC	Argmax	$\theta_{\min}$=20 dB	0.232	0.395	0.330
MUSIC	Argmax	$\theta_{\min}$=22.5 dB	0.211	0.345	0.327
MUSIC	Argmax	$\theta_{\min}$=25 dB	0.190	0.296	0.322

Table A.29: Best $F_{0.5}$ DoA configuration per algorithm and selection method for the CM-CFAR-guided pipeline (Flight 6), followed by the three next-best configurations. All listed configurations use background-spectrum normalization.

DoA	Selection	Parameters	Recall	FA/block	$F_{0.5}$
MUSIC	Argmax	θ_{min}=14 dB	0.276	0.724	0.276
MVDR	Argmax	θ_{min}=1 dB	0.339	0.949	0.275
MUSIC	GO-CFAR	g=3, n=16, α=10, bg	0.255	0.748	0.254
MVDR	SO-CFAR	g=6, n=16, α=10, bg	0.213	0.764	0.217
MUSIC	Argmax	θ_{min}=9 dB	0.297	0.802	0.275
MUSIC	Argmax	θ_{min}=10 dB	0.292	0.787	0.274
MUSIC	Argmax	θ_{min}=25 dB	0.179	0.390	0.274

Discussion DoA

Old pipeline results From Table A.27 it can be seen that MUSIC once again outperforms MVDR and that all the top picks use the 2D GO-CFAR algorithm.

RD-guided results From table A.28 and A.29 it can be seen that the GO-CFAR guided pipeline achieves the highest $F_{0.5}$, primarily because its argmax configurations outperform their CM-CFAR guided counterparts. The latter consistently obtains a higher recall, but this comes at the cost of more false detections. The best CFAR-based configurations produce a similar $F_{0.5}$ score. From these tables it can also be seen that once again the MUSIC algorithm with the argmax selection and additional detection threshold performs the best.

Comparison of the pipelines The results obtained using the different pipelines are very similar for this flight, with the results obtained using the top pick of the old pipeline slightly outperforming results obtained using the top pick of the new pipeline.

Results GODEC

Testing of the GODEC algorithm was done on the simulated version of Flight 6. This let us control the noise floor of the flight.

Table A.30: GoDec detection performance on the synthetic Flight 6 across the injected SNR. Fixed settings: n_{mov} = 1, rank 1, N_d = 250, and a CA-CFAR with three guard cells, ten training cells and threshold scaling 10.

SNR [dB]	Recall	FA/block
−45	0.172	66.0
−35	0.335	65.8
−25	0.616	63.6
−20	0.736	51.2
−15	0.905	56.6

Discussion GODEC

As seen in the table and video, the GODEC algorithm was unable to properly separate the drone and noise in the scene, leading to an extremely high number of false detections per block, making it unusable for the detection pipeline.

B

Ethical Implications of the Product and Engineering Process

Societal impacts (product). If our product were fully implemented, it would be a radar system that can reliably detect drones. Used on its own it would serve as an early warning system, and paired with a defense system it could protect people, for example by throwing nets to trap a drone or through a missile or gun system. That is the positive case: people protected from drone attacks. The problem is that no perfect radar exists. A system that misses a detection gives people a false sense of security during a military attack, and a system that falsely detects or misclassifies a target can cause psychological harm to people who believe they are being attacked. The concern multiplies when the radar is paired with an autonomously controlled weapon, since a false detection or misclassifications could trigger that weapon and seriously injure or kill people and animals, with disastrous consequences for the environment. The stakeholders are our BAP group, TU Delft, and Dopplium, who design the product and are responsible for the harm it may cause, along with the potential victims and the people the system was meant to protect when it fails.

Ethical aspects of development (process). Our radar emits 24 GHz electromagnetic waves. These are non-ionizing, so they should not cause health problems. Still, during data collection an unconsenting person could have passed near the radar and been exposed without the chance to agree to it, and the surrounding environment was exposed as well. We did not get official permission, but the recordings were done in an empty field. For the recordings with a human in the scene, everyone was a member of our group and consented, so there was no risk there. Beyond the recordings, our subgroup's work was mainly coding on a computer, so few other process concerns arose. The people affected were therefore possible bystanders, the animals in the field, and the environment.

Ethical evaluation. The one genuine concern in our process was that an unconsenting human could have received a small dose of EM radiation, with the same exposure reaching birds and the environment. In practice this stayed minor: because we were present at the recordings, we could tell people not to come closer, and no one did. The animals and environment were not negatively affected, apart from the noise of the drone. Several aspects of how we worked are ethically valuable: our participants consented, we kept the recording site controlled, and we tested the system rigorously. The clearest value position concerns the product itself. A defense system should never cause the death of a human, and it should never do so autonomously. This is a trade-off between fully autonomous operation and keeping a human in control, and our stance is firmly on keeping human control.

Reflection on design choices. Our main ethical consideration was to make the product perform as perfectly as possible: no missed detections, no false detections, proper classification, and range, velocity, and angle extracted as accurately as possible. That consideration directly shaped our choice to test rigorously, first in simulation with varying levels of noise and then with real-life recordings. Looking back, the step we would add is to make any user clearly aware of the exact capabilities of the system, including its possible points of failure, before relying on it.

Forward-looking responsibility. If this concept demonstrator were developed into a real application, it would first need to be even more accurate than what we built. All of its failure points should be clearly documented and researched, so that nothing bad happens if it is used in combination with other devices. The designers and sellers also have an obligation to ensure the system is not paired with other systems that could harm people in the event of an accident. In short, the product should only be released once its accuracy is high enough and its limitations are clearly documented, and the recordings used to develop it should be done in remote places where no one is accidentally exposed to the radar's electromagnetic waves.

C
Extra information

C.1. Finding radar heading direction

The radar's orientation was not directly recorded during the taking of the measurements. Therefore, the radar's boresight direction was estimated by using the geographic coordinates of the radar and the geographic coordinates of the drone when it was straight in front of the radar, for which flight 6 was used. The geographic locations of the radar and of the drone when it was right in front of the radar can be seen in Figure C.1. These coordinates were transformed into a local East-North coordinate system centered at the radar. The direction from the radar to this reference point was then calculated and used as the radar's boresight direction. This resulted in an estimated radar orientation of 160°, measured clockwise from the geographic north.

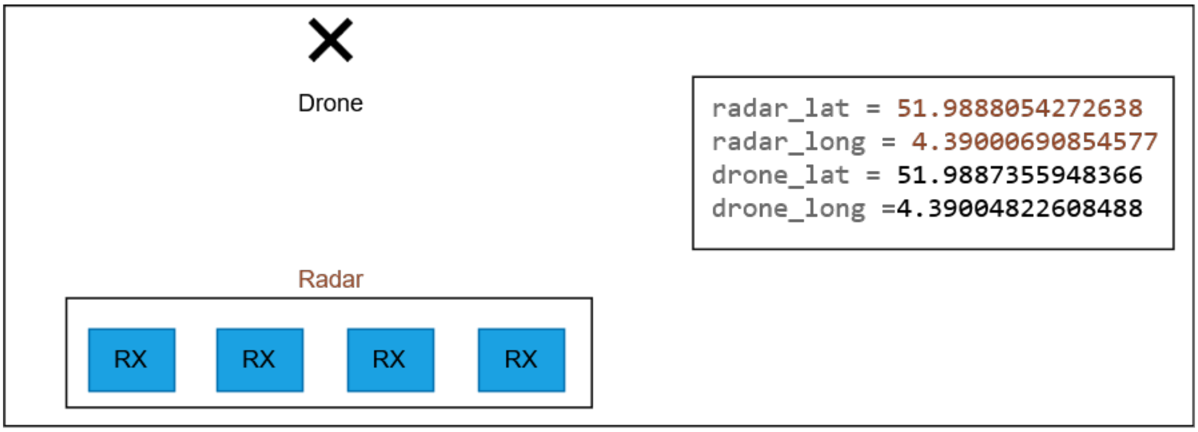


Figure C.1: Radar and drone position, when drone is straight in front of radar.

C.2. Using empty background

The background scene of the simulated trajectory was too idealistic. Mean subtraction took care of almost all of the stationary clutter that ended at the zero velocity bin. Adding phase jitter to the clutter made negligible difference. A different approach was attempted. Here the drone was put into an already existing background scene. The recording that was taken from the field without the drone present was used for this. The synthetic drone could be injected into this scene by forming the beat signal of the scene where only the drone was present, and adding this to the beat signals obtained from the empty scene recordings.
Several problems arise here. The main problem is the discrepancy between the amplitude of the beat signal of the synthetic drone and the amplitude of the beat signal of the empty scene. The former is determined by the radar equation. The latter's amplitude is also determined by the radar equation,

but it is also dependent on the ADC gain as well as any other gains and losses that might occur in the radar. The first thing that was attempted was to add the ADC to the power calculated using the radar equation. The ADC gain was equal to 21 dB. This was however not enough, the drone was still not visible in neither the range-time graph nor the range-Doppler graph. To confirm the drone was correctly injected, a very high gain was used. This confirmed that the drone was correctly injected and the issue was caused by the amplitude.
This could be solved by finding the ratio between the theoretical received power and the amplitude of the beat signal that the radar outputs. To do this, a recording of a dihedral with a theoretical received power of 1.17541e-7 W had been used. This recording was conducted in an anechoic chamber by subgroup 1. The dihedral was at a distance of about 3.8 meters from the radar, which was confirmed by the range time graph. Next, the mean power at the corresponding range bin was calculated. This was then divided by the FFT gain that was obtained by doing the fft over the fast-time axis. Using this value, the ratio between the theoretical received power and the amplitude of the beat signal given by the radar can be obtained. This came up to be 1.67×10^{11}. Unfortunately, this was still not quite enough, and an additional multiplication by about 40 was needed before the synthetic drone's range time graph resembled the real measurement of the same trajectory. Since this extra factor could not be justified properly, and due to time constraints, it was decided to not further investigate this method.

C.3. MUSIC Q=2 justification

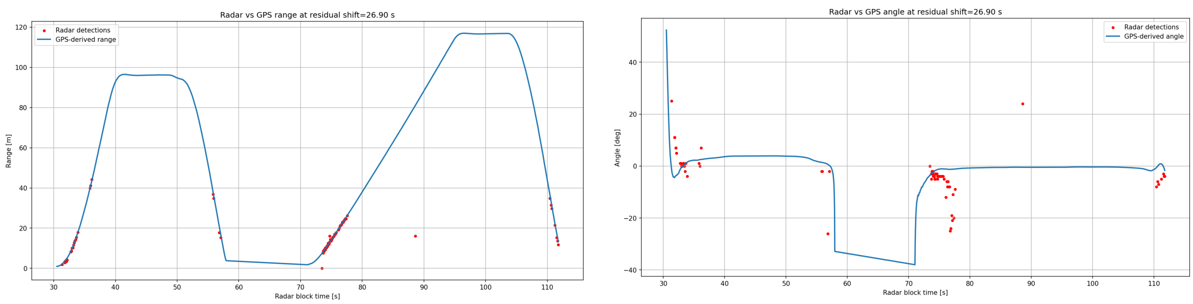
Running the new pipeline with (Q=1) and (Q=2) gave the results in Table C.1. Setting (Q=2) tells MUSIC to look for two sources. The table shows that the recall improves significantly if Q=2 is selected instead of Q=1. For this reason, it has been decided to use Q=2 to generate all results. The exact reason why Q=2 outperforms Q=1 could not be investigated further due to time constraints. Looking at the range-angle map, or at the eigenvalues would have possibly given us a more definite answer.

Table C.1: DoA recall for MUSIC subspace dimension $Q = 1$ versus $Q = 2$, for Flight 2.

Selection	$Q = 1$ recall	$Q = 2$ recall
Argmax, 0 dB	0.187	0.600
Argmax, 20 dB	0.002	0.470

C.4. Localization Trajectory Comparisons

Figure C.2 shows the comparison between the radar detections and the GPS derived range and angle for FLight 3.

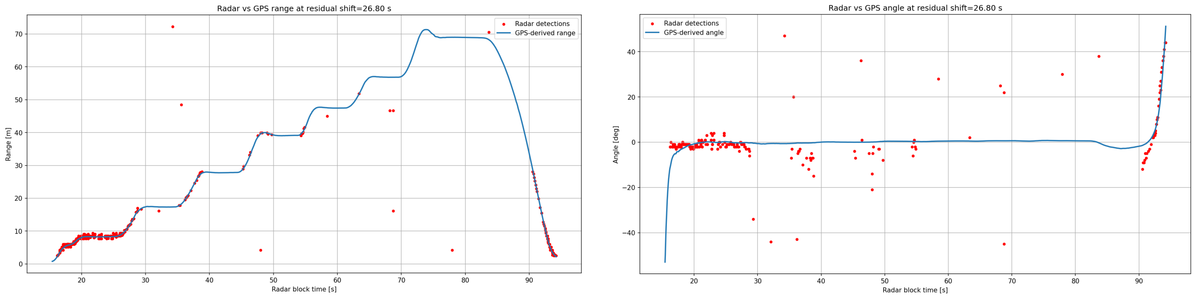


(a) Range estimates.

(b) Angle estimates.

Figure C.2: Comparison of the radar detections with the GPS-derived range and angle for Flight 3.

Figure C.2 shows the comparison between the radar detections and the GPS derived range and angle for Flight 6. The effects of Mean Subtraction can be clearly seen here. There are almost no detections when the drone is stationary.



(a) Range estimates.

(b) Angle estimates.

Figure C.3: Comparison of the radar detections with the GPS-derived range and angle for Flight 6.