RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Électronique

End of Studies Project Thesis

Submitted for the fulfillment of State Engineer Degree in Electronics

---

Adaptive Beamforming and Spectrum Sensing for Radio-Frequency Source Detection and Tracking

---

HAMMADI Lyes
KEBIECHE Fares

Under the supervision of

Prof. BELOUCHRANI Adel and **Dr. TEBACHE Soufiane**

Presented and publicly defended on : Juin, 27^{th} 2026

Jury Members:

President :	Dr. BOUADJENEK Nesrine	ENP, Algiers
Supervisor :	Pr. BELOUCHRANI Adel	ENP, Algiers
Co-Supervisor :	Dr. TEBACHE Soufiane	ENP, Algiers
Examiner :	Dr. LAHOULOU Atidel	ENP, Algiers

ENP 2026

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Électronique

End of Studies Project Thesis

Submitted for the fulfillment of State Engineer Degree in Electronics

---

Adaptive Beamforming and Spectrum Sensing for Radio-Frequency Source Detection and Tracking

---

HAMMADI Lyes
KEBIECHE Fares

Under the supervision of

Prof. BELOUCHRANI Adel and **Dr. TEBACHE Soufiane**

Presented and publicly defended on : Juin, 27^{th} 2026

Jury Members:

President :	Dr. BOUADJENEK Nesrine	ENP, Algiers
Supervisor :	Pr. BELOUCHRANI Adel	ENP, Algiers
Co-Supervisor :	Dr. TEBACHE Soufiane	ENP, Algiers
Examiner :	Dr. LAHOULOU Atidel	ENP, Algiers

ENP 2026

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Électronique

Mémoire de Projet de Fin d'Études

Présenté en vue de l'obtention du diplôme d'Ingénieur d'État en Électronique

---

Formation de voie adaptative et analyse spectrale pour la détection et la poursuite d'une source radiofréquence

---

HAMMADI Lyes
KEBIECHE Fares

Sous l'encadrement de

Prof. BELOUCHRANI Adel et **Dr. TEBACHE Soufiane**

Présenté et soutenu publiquement le : 27 juin 2026

Membres du jury :

Présidente :	Dr. BOUADJENEK Nesrine	ENP, Alger
Encadrant :	Pr. BELOUCHRANI Adel	ENP, Alger
Co-encadrant :	Dr. TEBACHE Soufiane	ENP, Alger
Examinatrice :	Dr. LAHOULOU Atidel	ENP, Alger

ENP 2026

الملخص :

تقدم هذه الدراسة طريقة تشكيل حزمة (Beamforming) تكيفية قوية مخصصة لتتبع هدف متحرك واحد يُصدر إشارة تردد راديوي (RF). بالاعتماد على خوارزمية تقدير من نوع LMS وبنية تشكيل حزمة تكيفية، كما تقدم أيضًا بنية كشف CMAP-CFAR قادرة على كشف وجود الهدف وتقدير تردد الإرسال الخاص به. أظهرت نتائج التنفيذ العتادي، المنجز على منصة Kraken SDR متعددة القنوات، أن النظام المقترح يحافظ على تتبع دقيق للهدف مع تمكين كشف إشارة RF وتقدير ترددها، مما يبرز إمكاناته في تطبيقات المراقبة الفورية وتحديد الاتجاه.

الكلمات المفتاحية: تشكيل الحزمة، مصفوفات هوائيات خطية ودائرية، معالجة إشارات المصفوفات، تتبع الأهداف، خوارزمية LMS، استشعار الطيف، الراديو معرف برمجياً (SDR)، تحديد الاتجاه، كشف CMAP-CFAR.

Résumé

Cette étude présente une méthode de formation de voie (FdV) adaptative robuste dédiée à la poursuite d'une seule cible mobile émettrice d'un signal radiofréquence. Fondée sur un algorithme d'estimation de type LMS et une architecture de formation de voie adaptative, elle présente également une structure de détection CMAP-CFAR permettant de détecter la présence d'une cible et d'estimer sa fréquence d'émission. L'implémentation matérielle, réalisée sur une plateforme SDR Kraken multi-canaux, a donné des résultats qui confirment que le système maintient un verrouillage précis de la cible et permet de détecter la présence d'un signal RF ainsi que d'en estimer la fréquence, soulignant ainsi son potentiel pour les applications de surveillance et de radiogoniométrie en temps réel.

Mots-clés : Formation de Voie, Réseaux d'antennes linéaires et circulaires, traitement d'antennes, poursuite de cible, algorithme LMS, analyse spectrale, radio logicielle (SDR), radiogoniométrie, détection CMAP-CFAR.

Abstract

This study presents a robust adaptive beamforming method dedicated to tracking a single mobile target emitting a radio frequency (RF) signal. Based on an LMS estimation algorithm and an adaptive beamforming architecture, it also introduces a CMAP-CFAR detection structure capable of detecting the presence of a target and estimating its transmission frequency. The hardware implementation, carried out on a multi-channel Kraken SDR platform, produced results demonstrating that the proposed system maintains accurate target tracking while enabling the detection of an RF signal and the estimation of its frequency, highlighting its potential for real-time surveillance and direction-finding applications.

Keywords: Beamforming, linear and circular antenna arrays, array signal processing, target tracking, LMS algorithm, spectrum sensing, software-defined radio (SDR), direction finding, CMAP-CFAR detection.

Acknowledgments

First and foremost, we would like to express our sincere gratitude to our supervisor, Professor Belouchrani Adel, Professor with Ecole Nationale Polytechnique, for the critical and creative spirit he instilled in us throughout this work. His ability to sharpen our thinking and push the boundaries of our approach has been one of the most valuable things we take away from this experience.

We would like to express our deepest gratitude to our co-supervisor, Dr. Tebbache, for his constant encouragement and for making available everything we needed to carry this project forward. His support was a true driving force behind the completion of this work.

We would like to thank the President of the jury, Dr. Bouadjenek Nesrine, Teacher-Researcher with Ecole Nationale Polytechnique, for agreeing to chair this committee and for everything she has taught us over the last three years. Her dedication to her students goes far beyond the classroom, and we are sincerely grateful for it.

We would also like to thank our examiner, Dr. Lahoulou Atidel , for agreeing to be part of the reading committee and for her time and careful review of the present work.

Dedication

I dedicate this work, first and foremost, to my younger self, who believed in his dreams and had the courage to trust his own potential. Every achievement I have accomplished in my life is the result of that faith, perseverance, and determination.

I also dedicate this work with my deepest gratitude to my beloved parents, whose unconditional love, endless sacrifices, financial and emotional support, guidance, and education have shaped the person I am today. Their encouragement has been the foundation of all my successes.

To my dear sisters, Keltoum, Bouchra, and Rihab, thank you for your love, support, and constant encouragement.

To my project partner, Lyes, I sincerely appreciate your seriousness, dedication, and commitment throughout this project. It has been a pleasure working alongside you.

Finally, I dedicate this work to my friends at ENP, Abderaouf, Lyes, Abdelalim, Adnane, and Wassim, whose friendship, support, and shared experiences have made this journey unforgettable.

May this work reflect the gratitude and respect I have for all those who have contributed, directly or indirectly, to my academic and personal growth.

Fares

I Yemma d Vava 3zizen, i Gma Tassa.

To you who gave me everything without ever counting the cost. This degree is yours as much as it is mine.

To my partner and friend, Kebieche Fares. We took our first steps at Polytechnique together, went through doubts and victories side by side. It is with immense joy that I walked this path alongside you. May this last milestone be only the beginning of what comes next.

To all those who are dear to me.

Lyes

Contents

List of Acronyms

General Introduction **15**

I Fundamental Concepts **17**

1.1 RF Source Geolocation Problem . . . 17
1.2 Array Signal Processing for Direction of Arrival estimation . . . 17
1.2.1 Signal Model . . . 18
1.2.2 Basic Principles of Antenna Beamforming . . . 19
1.2.3 Subspace Methods for Direction of Arrival Estimation . . . 19
1.3 The Least Mean Squares Adaptive Beamformer . . . 21
1.4 Adaptive Tracking Filters . . . 22
1.5 The Kalman Filter . . . 22
1.6 Cognitive Radio and Spectrum Sensing . . . 23
1.6.1 Spectrum Sensing Framework . . . 23
1.6.2 The concept of the Binary Hypothesis Test Statistic . . . 23
1.6.3 Classical Spectrum Sensing Methods . . . 24
1.7 Constant false alarm rate detection . . . 25
1.8 CFAR Detection for Spectrum Sensing . . . 25
1.9 State of the Art . . . 26
1.9.1 AI-Based Approaches for DoA Tracking . . . 26
1.9.2 AI-Based Approaches for Spectrum Sensing . . . 26
1.10 Conclusion . . . 27

II Adaptive spectrum sensing using CMAP-CFAR **28**

2.1 Clutter-Map CFAR overview . . . 28
2.1.1 Clutter Statistical Models . . . 29
2.1.2 Recursive Clutter Map Estimation . . . 30
2.2 Application to FFT-Based Spectrum Sensing . . . 31
2.3 Conclusion . . . 32

III Robust Beamforming for RF Source Tracking **33**

3.1 Signal Model and Problem Statement . . . 33

3.1.1 Observation Model . . . 33
3.1.2 Assumptions . . . 33
3.2 Identification Problem . . . 34
3.3 Projection Stage . . . 35
3.4 Algorithmic Application for Uniform Linear Arrays . . . 36
3.4.1 Signal Propagation Model for ULA . . . 36
3.4.2 Adaptive GSC Algorithm for ULA . . . 37
3.5 Performance Analysis . . . 39
3.5.1 ULA Stability . . . 39
3.5.2 ULA Steady-State MSE . . . 40
3.6 Extension to Uniform Circular Arrays . . . 40
3.6.1 Signal Propagation Model for UCA . . . 40
3.6.2 Adaptive GSC Algorithm for UCA . . . 41
3.7 Conclusion . . . 42

IV Simulations and Performance Analysis **43**
4.1 CMAP-CFAR Spectrum Sensing Simulations . . . 43
4.1.1 Simulation conditions and proof of concept . . . 43
4.1.2 Detection Performance as a Function of SNR . . . 45
4.1.3 Impact of the Detection Threshold K . . . 47
4.1.4 Robustness under Narrowband Interference and non Gaussian environment . . . 50
4.1.5 Influence of the Filter Parameter λ . . . 51
4.2 Source Tracking Simulations for Uniform Linear Arrays . . . 53
4.2.1 Azimuth Tracking simulation . . . 53
4.2.2 Impact of the Adaptation Step Size μ . . . 53
4.2.3 Impact of the LMS Step Size η . . . 55
4.2.4 Joint Parameter Optimisation: 3D Stability Landscape (μ, η) . . . 56
4.2.5 Robustness to jamming . . . 57
4.2.6 Influence of Array Size . . . 59
4.3 RF Source Tracking Simulations for UCA . . . 60
4.3.1 Impact of SNR degradation . . . 61
4.3.2 Influence of the Number of Antennas . . . 62
4.4 Conclusion . . . 62

V SDR Implementation and Experimental Results **63**
5.1 Hardware Platform . . . 63
5.1.1 Transmitting *TX* Setup . . . 63
5.1.2 Receiver *RX* Setup . . . 63
5.2 IQ Acquisition . . . 64
5.3 DoA Signal Preprocessing . . . 65
5.3.1 Phase Synchronisation and Decimation . . . 65
5.3.2 Baseband FIR Filtering . . . 66
5.3.3 MUSIC Initialisation, SNR and Source-Number Estimation . . . 66

5.4 Test Environment . . . 67
5.4.1 Testbench configuration . . . 67
5.5 DoA Tracking Results . . . 69
5.5.1 MUSIC Localisation for landmark setting . . . 69
5.5.2 Tracking Test for 433 MHz . . . 71
5.5.3 Tracking Test for 1 GHz . . . 74
5.5.4 Results Summary . . . 78
5.6 Spectrum Sensing Implementation . . . 79
5.6.1 Single-Band Mode and Subsystem Fusion . . . 79
5.6.2 Real-Time Tracking Test . . . 80
5.6.3 Five-Hop Sweep Mode . . . 81
5.6.4 Spectral CFAR-Master Architecture . . . 82
5.6.5 Scan State Machine . . . 84
5.7 Multi-band scanning Results . . . 85
5.8 Conclusion . . . 86

General Conclusion **87**

References **88**

Webography **91**

Appendices **93**

A Kraken Hardware Specifications **93**
1.1 KrakenSDR . . . 93

B Heimdall DAQ Firmware **95**
2.0.1 Global Architecture . . . 95
2.0.2 Principal modules used for Integration . . . 96
2.0.3 Role of `daq_chain_config.ini` . . . 97
2.0.4 OOT Block Modifications . . . 98

C Software Tools **99**
3.1 GNU Radio . . . 99
3.2 PyQt . . . 99
3.3 MATLAB . . . 99

D GNU Radio Flowgraphs **100**
4.1 HackRF Transmission Flowgraph . . . 100
4.2 Frequency Sweep Flowgraph . . . 101
4.3 MonoBand Detection and Adaptive Tracking Flowgraph . . . 101
4.4 Offline IQ Recording Flowgraph . . . 102

List of Figures

1.1 Architecture of the CA CFAR algorithm [15] 25

2.1 Sliding window technique [26] 29
2.2 Recursive clutter-map filter (EWMA). 30

3.1 Source tracking approach: overall block diagram showing the iterative correction of the steering vector and the beamformer adaptation. 34
3.2 Uniform Linear antenna array [27]. 36
3.3 Adaptive GSC structure for signal source extration (classical beamforming bloc from 3.1. The main branch performs coherent combining using the prior steering vector. The auxiliary branch captures residual signal, and the LMS updates the auxiliary weights. 37
3.4 Uniform circular array (UCA) [28]. 41

4.1 Proof of concept I: Example of a successful signal detection under the H_1 hypothesis. . 44
4.2 Proof of concept II: Temporal tracking and binary decision output matching the ground truth. 44
4.3 Threshold degradation effect: Binary decision contamination by false alarms due to an underestimated threshold factor. 45
4.4 False-alarm probability P_{fa} as a function of SNR for different threshold values. 46
4.5 Detection probability P_d as a function of SNR for threshold values $K \in \{5, 10, 15, 20\}$. 46
4.6 ROC curves P_d as a function of P_{fa} for different noise conditions. 47
4.7 P_d and P_{fa} as a function of the threshold factor K for low SNR conditions ($P_{fa} = 0.05$, $\lambda = 0.95$). 48
4.8 P_d and P_{fa} as a function of the threshold factor K for low and high SNR conditions ($P_{fa} = 0.0001$, $\lambda = 0.95$). 48
4.9 Spectral analysis illustrating a single isolated false alarm event across the 1024 FFT bins ($K = 7$, SNR $= 20\,\text{dB}$, $\lambda = 0.95$). 49
4.10 Effect of over-thresholding. Signal of interest is completely missed. 49
4.11 Detection probability versus SNR under AWGN and Bernoulli-Gaussian impulsive noise. 50
4.12 (a) effect of frequency separation (Δf), (b) effect of reference window size (N_{ref}), and (c) effect of interference power (INR), $\lambda = 0.95$. 51
4.13 P_d and P_{fa} as a function of λ in a high-SNR regime (SNR $= 20\,\text{dB}$, $K = 10$). 52
4.14 P_d and P_{fa} as a function of λ for ($SNR = 5\text{dB}$, $K = 10$). 52
4.15 Initial demonstration of azimuth tracking execution on a 5-element ULA. 53

4.16 DoA tracking performance grid under severe noise conditions (SNR = −10 dB) for varying step sizes μ. The vertical magenta dotted line denotes the empirical convergence snapshot index n_c. 54
4.17 DoA tracking performance grid under intermediate noise conditions (SNR = 0 dB). . . 54
4.18 DoA tracking performance grid under high signal integrity (SNR = 10 dB). 55
4.19 Summary of the influence of the adaptation step-size μ at SNR = 10 dB. 55
4.20 Steady-state RMSE as a function of the LMS step size η for various SNR levels ($\mu = 0.050$). 56
4.20 3D stability maps showing the steady-state RMSE as a function of μ and η under different SNR conditions. 57
4.21 Static & mobile jammer with abrupt jumps (SNR = 10 dB, JNR = 10 dB) 58
4.22 Static & mobile jamme with angular drift (SNR = 10 dB, JNR = 12 dB) 59
4.23 Evolution of the acceptable locking interval as a function of the number of array elements. 60
4.24 2D Azimuth and Elevation tracking ($M = 5$, SNR = 10 dB). 60
4.25 3D trajectory tracking on the unit sphere ($M = 5$, SNR = 10 dB). 61
4.26 UCA 2D tracking RMSE θ, ϕ vs SNR 61
4.27 Effect of increasing the number of antenna in uniform circular arrays. 62

5.1 HackRF One transmission setup. 64
5.2 KrakenSDR with noise source activated during phase-calibration startup. 65
5.3 Frequency response of the complex bandpass FIR filter. 66
5.4 Proposed DoA tracking pipeline 67
5.5 representative top view of the test bench 68
5.6 Transmitter positions at 433 MHz $d = 33$ cm. 68
5.7 Array reconfigured for 1 GHz tests, $d = 15$ cm. 69
5.8 UCA configured for 433 MHz , diameter = 33 cm, angular separation 72° 69
5.9 MUSIC Localisation at 433 MHz 70
5.10 MUSIC Localisation at 1 GHz. 70
5.11 Normalized cross-correlation between all channel pairs of the 5-element ULA at 433 MHz. 72
5.12 Comparison of the spectra before and after filtering. 73
5.13 Estimated DoA trajectory during the 433 GHz experiment. 73
5.14 Initial and final beam patterns at 433 Ghz 74
5.15 Normalized cross-correlation between all channel pairs of the 5-element ULA at 1 GHz. 74
5.16 Comparison of the spectra before and after filtering at 1 GHz. 75
5.17 Estimated DoA trajectory during the 1 GHz experiment. 76
5.18 DoA tracking at 1 GHz comparisons between the original technique and the kalman enhanced version. 77
5.19 Tracking at 433 MHz with and without kalman post-filtering. 78
5.20 Real-time Qt DoA interface during the single-band fusion test at 1 GHz for RF signal and 1 MHz for baseband signal. The SCMAP block detected the emitter at $f_{\text{abs}} = 1000.984$ MHz and commanded the Xlating FIR filter accordingly. The system reports $\hat{\theta} = -26.97°$. 80
5.21 Real-time Qt DoA interface initialisation at broadside ($\hat{\theta} = 0°$). The SCMAP block has detected the emitter at $f_{\text{abs}} = 1000 + 1 = 1001$ MHz. 81

5.22 Real-time Qt DoA interface successful tracking after the transmitter is moved to the left. The beam follows the source direction. The displayed baseband frequency (1 MHz modulate onto 1 GHz) confirms that both subsystems are operating jointly in real time. 81
5.23 Multi-band or Wideband spectrum sensing pipeline 83
5.24 Sweep system locked at $f_{\text{abs}} = 1000.406\,\text{MHz}$. 85

1.1 KrakenSDR PCB component overview. (1) SMA antenna inputs, (2) Bias Tee, (3) ESD protection, (4) Noise calibration switches, (5) R820T2 tuner, (6) RTL2832U ADC, (7) Noise source, (8) USB Hub, (9) Individual tuner on/off DIP switches, (10) USB TypeC DATA, (11) USB TypeC PWR. 93

2.1 Heimdall DAQ firmware block diagram. 96

4.1 GNU Radio flowgraph for RF signal transmission via HackRF One. Blocks: Signal Source → Throttle → Osmocom Sink. 100
4.2 GNU Radio flowgraph used for wideband frequency sweep and spectrum sensing. . . . 101
4.3 GNU Radio flowgraph used for monoband detection and adaptive DoA tracking. . . . 101
4.4 Frequency Xlating FIR Filter block in GNU Radio. 102
4.5 GNU Radio flowgraph used for multichannel IQ data recording. 102

List of Tables

5.1 IQ acquisition parameters. 65
5.2 Array geometry per frequency band. 68
5.3 Initial DoA localisation results. 71
5.4 Estimated inter-channel delay matrix (nanoseconds). 72
5.5 Estimated inter-channel delay matrix (nanoseconds) at 1 GHz. 75
5.6 Kalman filter parameters at 1 GHz tracking. 77
5.7 Tracking performance comparison: Original GSC-LMS vs Kalman-enhanced, at 433 MHz and 1 GHz. 78
5.8 GNU Radio embedded Python blocks 83
5.9 Frequency sweep parameters. 84
5.10 Inter-block PMT message protocol. 85

1.1 KrakenSDR technical specifications. 94

2.1 Key `daq_chain_config.ini` parameters adjusted during integration. 97

List of Acronyms

AWGN Additive White Gaussian Noise

CA-CFAR Cell Averaging Constant False Alarm Rate

CFAR Constant False Alarm Rate

CMAP Clutter Map

CUT Cell Under Test

DoA Direction of Arrival

EWMA Exponentially Weighted Moving Average

ESPRIT Estimation of Signal Parameters via Rotational Invariance Techniques

FFT Fast Fourier Transform

GLRT Generalized Likelihood Ratio Test

GO-CFAR Greatest Of CFAR

GSC Generalized Sidelobe Canceller

IIR Infinite Impulse Response

IQ In-phase and Quadrature

KF Kalman Filter

LMS Least Mean Squares

MAE Mean Absolute Error

MED Maximum Eigenvalue Detector

MMED Minimum to Maximum Eigenvalues Detection

MMSE Minimum Mean Square Error

MSE Mean Square Error

MUSIC Multiple Signal Classification

MVDR Minimum Variance Distortionless Response

OOT Out Of Tree

OS-CFAR Ordered Statistics CFAR

PDF Probability Density Function

PMT Polymorphic Type

PU Primary User

RSS Received Signal Strength

RF Radio Frequency

RMSE Root Mean Square Error

ROC Receiver Operating Characteristic

SCMAP Spectral Clutter Map

SDR Software Defined Radio

SNR Signal-to-Noise Ratio

SOI Signal of Interest

SVD Singular Value Decomposition

TCP Transmission Control Protocol

TDOA Time Difference of Arrival

UAV Unmanned Aerial Vehicle

UCA Uniform Circular Array

ULA Uniform Linear Array

UHF Ultra High Frequency

VHF Very High Frequency

General Introduction

The radio frequency spectrum is the epicenter of global technological stakes, drawing the intense focus of major international entities and institutions. The drastic development of telecommunications and tele-control technologies over the last decade has fueled ambitions of telecommunication and industry stakeholders, who seek to control evry part of the spectrum for multiple objectives.

Despite the advent of innovative, high-bandwidth technologies such as fiber optics, radio communications remain the primary medium where the vast majority of public and critical transactions occur. Whether it is cellular networks, wireless local area networks, radar systems, ground-to-ground/air-to-air VHF communications, or lastly, UAV control, the omnipresence of RF is undeniable. Consequently, possessing optimal tools that allow them not only to access the spectral resource but also to monitor it with maximum efficiency, has become a must.

Motivations & Problem Statement

Currently, the primary challenges in this domain present themselves under two distinct but complementary facets. On the civil front, the main objective is to optimize the allocation of spectral resources principally to maximize user coexistence, these problems are among the cognitive radio field. Conversely, in the realm of defense and electronic warfare, the RF spectrum is no longer merely a communications channel, it represents a continuous source of potential threats. This dual reality is particularly evident with the rapid proliferation of Unmanned Aerial Vehicles, that can be easily exploited for malicious purposes. Consequently, the detection, classification, and tracking of radio-frequency sources has become a highly complex, yet essential, task.

Report Structure

- **Chapter I – Fundamental Concepts**

  Presents the main classical methods used for Spectrum sensing, radio frequency source detection and tracking. It also introduces the basic concepts of beamforming, direction finding, and adaptive processing.

- **Chapter II – Adaptive Spectrum Sensing using CMAP-CFAR**

  Explains the proposed CMAP-CFAR detector. It presents the reasons behind the choice of this method, its operating principle.

- **Chapter III – Robust Beamforming for RF Source Tracking**

  This chapter describes the core of the RF source tracking algorithm proposed by *Gazor* [1] for Uniform Linear Array (ULA), and extend it to the (UCA) case.

- **Chapter IV – Simulations and Performance Analysis**

  This chapter presents the simulation results and the performance analysis.

  - **Part I :** The CMAP-CFAR system.
  - **Part II :** The robust beamforming for RF source tracking algorithm.

- **Chapter V – SDR Implementation and Experimental Results**

  In this chapter, we focus on the acquisition of real-world data using the Kraken SDR platform and the validation of the proposed algorithms. We also aim to implement these algorithms in real time, followed by experimental tracking results and a comprehensive evaluation of their practical performance.

Chapter I

Fundamental Concepts

Radio-frequency source detection and tracking remains a major research topic due to the complexity of electromagnetic environments and the growing need for reliable localization. This chapter introduces the theoretical foundation and most common and performant technics present in the litterature.

1.1 RF Source Geolocation Problem

The geolocation of RF emitters refers to the physical process of estimating the spatial coordinates of an active signal source. This capability is of paramount importance across modern defense operations, electronic warfare. Depending on the system architecture and hardware constraints, RF source localization techniques are categorized into three fundamental approaches [2]:

- **Time Difference of Arrival (TDOA):** Relies on the relative time delay of the incoming signal across multiple spatially distributed receiver nodes.
- **Received Signal Strength (RSS):** Estimates the emitter's position based on the relative signal power attenuation over distance.
- **Direction of Arrival (DoA):** Determines the angular incidence of the incoming wavefront using antenna arrays. The difference between the received signals at the array elements, in terms of phase or time delay, is exploited to estimate the direction from which the signal originates.

1.2 Array Signal Processing for Direction of Arrival estimation

We focuse in our work on the DoA approach, array signal processing is precisely the discipline of exploiting these structured inter-element differences: rather than discarding the phase relationships introduced by propagation geometry, it treats them as the primary information-bearing quantity from which source directions can be estimated and interferers suppressed. Three algorithmic families address this problem: **classical beamforming methods** (Bartlett/

MVDR), which steer a beam across candidate directions and measure received power, **subspace methods** (MUSIC/ ESPRIT), which decompose the spatial covariance matrix to achieve super-resolution, and **adaptive methods**, which update weights iteratively to track moving sources. All three operate on the same underlying signal model, derived below [3].

1.2.1 Signal Model

For an array of M receiving elements observing p far-field sources, the single-source case is first considered. In this case, the signal received at the m-th sensor is modeled as a delayed replica of the emitted waveform [3] :

$$x_m(t) = s(t - \Delta t_m), \tag{I.1}$$

where Δt_m denotes the propagation delay relative to a reference sensor.

Under the narrowband assumption, the envelope of the signal varies slowly with respect to the inter-sensor delays, so that the temporal shift can be approximated by a phase term only. The received signal at sensor m then becomes [3]:

$$x_m(t) \approx s(t)e^{-j2\pi f_c \Delta t_m}, \tag{I.2}$$

where f_c is the carrier frequency.

The received baseband signal is sampled every T seconds. Since the symbol period is assumed much larger than the inter-sensor propagation delays, i.e.

$$T \gg \Delta t_m, \qquad m = 1, \ldots, M,$$

the narrowband approximation remains valid in discrete time, and the observation at the m-th sensor reduces to a direction-dependent phase shift of the same waveform. Denoting by $x_m[n] = x_m(nT)$ and $s[n] = s(nT)$, one obtains

$$x_m[n] \approx a_m(\boldsymbol{\theta})\, s[n], \tag{I.3}$$

where

$$a_m(\boldsymbol{\theta}) = e^{-j2\pi f_c \Delta t_m} \tag{I.4}$$

is the steering coefficient associated with sensor m.

Stacking the M sensor observations yields the array snapshot model

$$\mathbf{x}[n] = \mathbf{a}(\boldsymbol{\theta})\, s[n] + \mathbf{n}[n], \tag{I.5}$$

with $\mathbf{a}(\boldsymbol{\theta}) \in \mathbb{C}^M$ the steering vector associated with the source direction $\boldsymbol{\theta}$.

For p simultaneous sources, the received signal is obtained by superposition,

$$\mathbf{x}[n] = \sum_{k=1}^{p} \mathbf{a}(\boldsymbol{\theta}_k) s_k[n] + \mathbf{n}[n]. \tag{I.6}$$

Introducing the source vector

$$\mathbf{s}[n] = \begin{bmatrix} s_1[n] & s_2[n] & \cdots & s_p[n] \end{bmatrix}^T, \tag{I.7}$$

and the array manifold matrix

$$\mathbf{A}(\boldsymbol{\Theta}) = \begin{bmatrix} \mathbf{a}(\boldsymbol{\theta}_1) & \mathbf{a}(\boldsymbol{\theta}_2) & \cdots & \mathbf{a}(\boldsymbol{\theta}_p) \end{bmatrix} \in \mathbb{C}^{(M,p)}, \tag{I.8}$$

the multi-source array model becomes

$$\mathbf{x}[n] = \mathbf{A}(\boldsymbol{\Theta})\mathbf{s}[n] + \mathbf{n}[n]. \tag{I.9}$$

1.2.2 Basic Principles of Antenna Beamforming

By applying a weighted linear combination to the outputs of each array element, the overall array output is obtained as [3]:

$$y[n] = \mathbf{w}^H \mathbf{x}_n, \tag{I.10}$$

where the M-row, 1-column complex vector $\mathbf{w}$ contains the beamforming weights w_m for $m = 0, 1, \ldots, M-1$. The array response to incoming signals is controlled by adjusting the elements of $\mathbf{w}$. This process is known as *spatial filtering*, or *beamforming* [3].

The beamformer amplitude and weighting coefficients define the antenna radiation pattern and its directivity. The array factor (beampattern) is expressed as

$$W(\theta) = \sum_{m=0}^{M-1} w_m \, e^{-jm\phi} = \mathbf{w}^H \mathbf{a}(\theta), \qquad \text{with} \quad \phi = \frac{2\pi d}{\lambda} \sin\theta, \tag{I.11}$$

where the steering vector is defined by

$$\mathbf{a}(\theta) = \begin{bmatrix} 1 & e^{-j\phi} & e^{-j2\phi} & \cdots & e^{-j(M-1)\phi} \end{bmatrix}^T. \tag{I.12}$$

The beamformer output can therefore be written as the inner product

$$y[n] = \mathbf{w}^H \mathbf{x}_n = s_0[n] \, \mathbf{w}^H \mathbf{a}(\theta_0) = s_0[n] \, W(\theta_0). \tag{I.13}$$

If $\mathbf{w}$ and $\mathbf{a}(\theta)$ are orthogonal, then $\mathbf{w}^H \mathbf{a}(\theta) = 0$, which implies that the corresponding signal component is completely suppressed (nulled) [3].

1.2.3 Subspace Methods for Direction of Arrival Estimation

Subspace-based approaches (High resolution approaches) exploit the eigenstructure of the spatial covariance matrix to separate the observation space into a low-dimensional *signal subspace* and an orthogonal *noise subspace*.

Eigenstructure of the Spatial Covariance Matrix

The spatial covariance matrix is defined under the assumption that $\mathbf{s}_n$ and $\mathbf{n}_n$ are uncorrelated. With $\mathbf{R}_{ss} = \mathbb{E}[\mathbf{s}_n \mathbf{s}_n^H]$ the signal covariance and $\mathbf{n}_n \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I})$, one obtains [3]:

$$\mathbf{R}_{xx} = \mathbb{E}[\mathbf{x}_n \mathbf{x}_n^H] \tag{I.14}$$

$$= \mathbf{A}\mathbf{R}_{ss}\mathbf{A}^H + \sigma^2 \mathbf{I}. \tag{I.15}$$

Because $\mathbf{R}_{xx}$ is Hermitian, it admits a unitary eigendecomposition with real eigenvalues. Arranged in descending order,

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p > \lambda_{p+1} = \cdots = \lambda_M = \sigma^2,$$

the first p eigenvalues are associated with the *signal-plus-noise* subspace, whereas the remaining $M - p$ eigenvalues correspond to the *noise-only* subspace. The eigendecomposition can be written compactly as

$$\mathbf{R}_{xx} = \begin{bmatrix} \mathbf{Q}_s & \mathbf{Q}_n \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_s & \mathbf{0} \\ \mathbf{0} & \sigma^2 \mathbf{I}_{M-p} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_s & \mathbf{Q}_n \end{bmatrix}^H, \tag{I.16}$$

where $\mathbf{Q}_s$ is an M-row, p-column matrix spanning the signal subspace, and $\mathbf{Q}_n$ is an M-row, $(M-p)$-column matrix spanning the noise subspace.

The steering vectors $\mathbf{a}(\boldsymbol{\theta}_i)$ of the p incoming sources lie in the column space of $\mathbf{Q}_s$, consequently they are orthogonal to every column of $\mathbf{Q}_n$. This orthogonality is the cornerstone of subspace-based DoA estimation: if a trial steering vector $\mathbf{a}(\theta)$ belongs to the signal subspace, then $\mathbf{Q}_n^H \mathbf{a}(\theta) = \mathbf{0}$. In practice, with a finite number N of snapshots, one uses the sample covariance matrix

$$\hat{\mathbf{R}}_{xx} = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{x}[n]\, \mathbf{x}^H[n], \tag{I.17}$$

whose eigenvectors approximate $\mathbf{Q}_s$ and $\mathbf{Q}_n$. Because SVD algorithms exist, many implementations decompose the data matrix $\mathbf{X}$, which is an M-row, N-column complex-valued matrix (whose rows are the snapshots) rather than performing an eigenvalue decomposition on $\hat{\mathbf{R}}_{xx}$ directly. The SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$ yields the same subspaces: the columns of $\mathbf{V}$ are the eigenvectors of $\hat{\mathbf{R}}_{xx}$, and the singular values are the square roots of its eigenvalues [3].

Multiple Signal Classification (MUSIC)

The MUSIC algorithm exploits the orthogonality between the signal and noise subspaces. Since the true steering vectors of the incoming sources belong to the signal subspace, they satisfy [4]:

$$\mathbf{a}^H(\theta)\, \mathbf{Q}_n = \mathbf{0}. \tag{I.18}$$

In practice, this condition is only approximately satisfied because $\mathbf{Q}_n$ is estimated from finite data. The DoAs are therefore located at the peaks of the MUSIC spatial spectrum [4]:

$$P_{\text{MUSIC}}(\theta) = \frac{1}{\mathbf{a}^H(\theta)\,\mathbf{Q}_n\mathbf{Q}_n^H\,\mathbf{a}(\theta)}. \tag{I.19}$$

For a Uniform Circular Array composed of M antenna elements, the received signal can be modeled as [5]:

$$\mathbf{X}(n) = \mathbf{a}(\theta,\phi)s(n) + \mathbf{n}(n) \tag{I.20}$$

The 2D MUSIC pseudospectrum is computed as [5]:

$$P_{MUSIC}(\theta,\phi) = \frac{1}{\mathbf{a}^H(\theta,\phi)\mathbf{E}_n\mathbf{E}_n^H\mathbf{a}(\theta,\phi)} \tag{I.21}$$

The DOA estimates correspond to the maximum of the pseudospectrum [5]:

$$(\hat{\theta},\hat{\phi}) = \arg\max_{\theta,\phi} P_{MUSIC}(\theta,\phi) \tag{I.22}$$

1.3 The Least Mean Squares Adaptive Beamformer

The **LMS** algorithm, was introduced by Widrow and Hoff as a stochastic approximation to the Wiener solution. Its application to antenna arrays by Widrow *et al.*[6] gave rise to the first practical adaptive beamformer. The appeal of LMS lies in a deceptively simple observation: the gradient of the mean-square error with respect to $\mathbf{w}$ can be *estimated* from a single snapshot, eliminating the need to compute and invert the spatial covariance matrix $\mathbf{R}_{xx}$.

Cost Function and Gradient Descent

Let $d[n]$ represent a reference signal correlated with the desired source and uncorrelated with interferers. The beamformer output is $y[n] = \mathbf{w}^H[n]\,\mathbf{x}[n]$, and the instantaneous error is:

$$e[n] = d[n] - y[n] = d[n] - \mathbf{w}^H[n]\,\mathbf{x}[n]. \tag{I.23}$$

The optimal Wiener weight vector optimizes MSE surface $\xi(\mathbf{w}) = \mathbb{E}[|e[n]|^2]$, whose unique global minimum is the Wiener Hopf solution [7]:

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\,\mathbf{r}_{dx}, \tag{I.24}$$

where $\mathbf{r}_{dx} = \mathbb{E}[d[n]\,\mathbf{x}^*[n]]$ is the cross-correlation vector. Computing (I.24) requires estimating and inverting $\mathbf{R}_{xx}$, an $\mathcal{O}(M^3)$ operation unsuitable for real-time adaptation.

LMS circumvents this by replacing the true gradient $\nabla_{\mathbf{w}}\xi = -2\mathbf{r}_{dx} + 2\mathbf{R}_{xx}\mathbf{w}$ with its instantaneous estimate $\hat{\nabla} = -2e[n]\,\mathbf{x}^*[n]$ [7]. The resulting weight update is:

$$\mathbf{w}[n+1] = \mathbf{w}[n] + \mu\,e[n]\,\mathbf{x}^*[n], \tag{I.25}$$

where $\mu > 0$ is the *step size*, the single parameter governing the algorithm's behaviour. Each iteration of (I.25) requires only $\mathcal{O}(M)$ multiplications, making LMS the computationally lightest adaptive beamformer available.

Convergence and Stability

The step size μ controls the fundamental trade-off between convergence speed and steady-state misadjustment. For convergence in the mean, μ must satisfy [7]:

$$0 < \mu < \frac{1}{\lambda_{\max}(\mathbf{R}_{xx})}, \tag{I.26}$$

where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{R}_{xx}$.

1.4 Adaptive Tracking Filters

Adaptive beamforming provides instantaneous spatial estimates, but in the presence of a *moving* RF source the angular trajectory $\theta(t)$ must be tracked over time. A raw sequence of DoA estimates is noisy and discontinuous, it represents measurements of a hidden dynamic state. The appropriate framework for optimal state estimation from noisy observations is the *Kalman filter* and its nonlinear extensions [8], which impose a dynamic motion model on the trajectory to regularise and smooth the estimates [7].

1.5 The Kalman Filter

Introduced in [9], the Kalman filter is regarded as the optimal linear estimator for the hidden state of a dynamical system under the assumption of Gaussian process and measurement noise. It is a recursive filter, which updates the state estimate using only the previous estimate and the current observation, makes it particularly well suited for real-time tracking applications [7]. The system is described by the state-space equations [10]:

$$\mathbf{x}_k = \mathbf{F}_k\,\mathbf{x}_{k-1} + \mathbf{B}_k\,\mathbf{u}_k + \mathbf{w}_k, \tag{I.27}$$

$$\mathbf{z}_k = \mathbf{H}_k\,\mathbf{x}_k + \mathbf{v}_k, \tag{I.28}$$

where $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ and $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ are the process and measurement noises, respectively.

The filter alternates between a **prediction** step, which propagates the state estimate forward in time using the dynamic model, and an **update** step, which corrects this prediction using the new observation :

Prediction :

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k\,\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k\,\mathbf{u}_k, \qquad \mathbf{P}_{k|k-1} = \mathbf{F}_k\,\mathbf{P}_{k-1|k-1}\,\mathbf{F}_k^T + \mathbf{Q}_k. \tag{I.29}$$

Update :

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\,\mathbf{H}_k^T\,\left(\mathbf{H}_k\,\mathbf{P}_{k|k-1}\,\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1}, \tag{I.30}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{H}_k\,\hat{\mathbf{x}}_{k|k-1}\right), \tag{I.31}$$

$$\mathbf{P}_{k|k} = \left(\mathbf{I} - \mathbf{K}_k\,\mathbf{H}_k\right)\mathbf{P}_{k|k-1}. \tag{I.32}$$

The *Kalman gain* $\mathbf{K}_k$ in (I.30) arbitrates between the model prediction and the observation: when measurement noise dominates ($\mathbf{R}_k$ large), $\mathbf{K}_k$ is small and the filter trusts the prediction, when process noise dominates ($\mathbf{Q}_k$ large), $\mathbf{K}_k$ increases and the filter follows the measurement. This principled balancing of uncertainty makes the KF the optimal MMSE estimator among all linear filters under the Gaussian noise assumption [10].

In the DoA tracking context, the state vector $\mathbf{x}_k = [\theta_k,\ \dot{\theta}_k]^T$ encodes the current angle and its rate of change, and $\mathbf{z}_k = \hat{\theta}_k$ is the noisy DoA estimate delivered by the spatial processing module at each time step [8].

1.6 Cognitive Radio and Spectrum Sensing

Spectrum sensing is the fundamental enabler of dynamic spectrum access in cognitive radio networks, allowing secondary users to detect primary-user occupancy without causing harmful interference [11]. Its core challenge lies in reliable detection under low SNR, unknown signal statistics, and time-varying propagation conditions.

In this work, spectrum sensing is used to detect the presence of a radio frequency signal (SOI) and to estimate its frequency.

1.6.1 Spectrum Sensing Framework

Let $x[n]$ denote the discrete-time baseband signal received at a cognitive radio terminal. The binary hypothesis testing problem for spectrum sensing in the l-th frequency subband is formulated as [12]:

$$x[n] = \begin{cases} w[n], & \mathcal{H}_0 \quad \text{(PU absent)} \\ h \cdot s[n] + w[n], & \mathcal{H}_1 \quad \text{(PU present)} \end{cases} \tag{I.33}$$

where $s[n]$ is the PU transmitted signal, h is the complex channel gain, and $w[n] \sim \mathcal{CN}(0, \sigma_N^2)$ is AWGN. The detector must choose between $\mathcal{H}_0$ and $\mathcal{H}_1$ based on a test statistic derived from the received samples, while maintaining a target probability of false alarm P_{FA} and maximising the probability of detection P_d [12].

1.6.2 The concept of the Binary Hypothesis Test Statistic

A binary hypothesis test aims to decide whether the observed data $x[n]$ supports the null hypothesis H_0 or the alternative hypothesis H_1. This decision is made by computing a test statistic from the data and comparing it to a predefined threshold [12].

The test statistics based on a ratio of two quantities, is the *likelihood ratio test*, defined as

$$T(x[n]) = \frac{p(x[n] \mid H_1)}{p(x[n] \mid H_0)}, \tag{I.34}$$

Here, $p(x[n] \mid H_1)$ and $p(x[n] \mid H_0)$ denote the PDF of the observed (possibly multivariate) signal vector $x[n]$ under hypotheses H_1 and H_0, respectively.

The decision rule is given by

$$T(x[n]) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma, \tag{I.35}$$

where γ is a threshold. If $T(\mathbf{y}) > \gamma$, the decision favors H_1; otherwise, H_0 is selected [12].

1.6.3 Classical Spectrum Sensing Methods

A number of sensing approaches have been proposed in the literature, ranging from optimal but knowledge-intensive detectors to simple blind techniques.

Energy Detection

Energy detection is the simplest and most widely deployed spectrum sensing technique, owing to its low computational complexity and the absence of any requirement for prior knowledge of the PU signal structure [11]. The decision statistic is the accumulated signal energy over a sensing window of N samples:

$$T_{\text{ED}} = \sum_{n=0}^{N-1} |x[n]|^2. \tag{I.36}$$

The key limitation of energy detection is its dependence on σ_N^2: any uncertainty in the noise power estimate directly degrades the threshold calibration and causes the P_{FA} to drift. This effect, known as the *SNR wall* [13], imposes a fundamental floor below which energy detection fails regardless of observation length. Furthermore, the energy detector cannot distinguish between the PU signal and interference sources, making it vulnerable in dense RF environments.

Eigenvalue-Based Detection Methods

In signal processing applications, the eigenvalues of a covariance matrix (or its estimate) are closely related to the received signal power and are widely used in the construction of detection test statistics [12].

After computation, the eigenvalues are usually sorted in descending order as

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m. \tag{I.37}$$

Based on these ordered eigenvalues, several test statistics can be defined [12]:

$$T_{\text{GLRT}} = \frac{\lambda_1}{\frac{1}{m}\sum_{i=1}^{m} \lambda_i}, \qquad T_{\text{MMED}} = \frac{\lambda_1}{\lambda_m}, \qquad T_{\text{MED}} = \frac{\lambda_1}{\sigma^2}. \tag{I.38}$$

The GLRT and MMED detectors are blind, as they do not require prior knowledge of the noise

variance. In contrast, the MED detector is semi-blind since it depends on the knowledge of σ^2.

1.7 Constant false alarm rate detection

Target detection is a fundamental task in radar systems. Among the various detection techniques, CFAR detectors are widely used because they adapt the detection threshold to variations in noise, clutter, and interference while maintaining a predefined probability of false alarm. Unlike fixed-threshold detectors, which are effective only in homogeneous environments, CFAR methods continuously adjust the threshold according to the local background conditions [14].

The basic principle of CFAR consists of estimating the noise or clutter level around the CUT using neighboring reference cells while excluding guard cells to avoid target contamination. A target is declared present when the signal level in the CUT exceeds the adaptive threshold [14].

The most common CFAR technique is the CA-CFAR, where the local clutter level is estimated by the arithmetic mean of the reference cells [15]:

$$Q = \frac{1}{M} \sum_{i=1}^{M} x_i, \tag{I.39}$$

where M is the number of reference cells and x_i denotes the power measured in the i-th reference cell. The resulting estimate is then scaled to obtain the detection threshold corresponding to the desired false alarm probability.

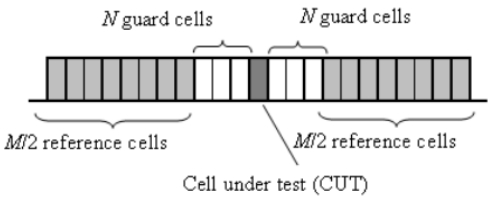


Figure 1.1: Architecture of the CA CFAR algorithm [15]

1.8 CFAR Detection for Spectrum Sensing

Originally developed for radar target detection in non-stationary clutter [16], Constant False Alarm Rate techniques have been progressively extended to spectrum sensing, where the challenge is analogous: deciding whether a frequency bin is occupied by a signal or by noise alone, under an unknown and time-varying noise floor. The common thread is the need to maintain a guaranteed P_{FA} without prior knowledge of the background power level.

The core mechanism is identical across all CFAR variants: the local noise power is estimated from *reference cells* surrounding the *cell under test* (CUT), and the detection threshold is set as a scaled version of this estimate:

$$P(k) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \alpha \cdot \hat{\sigma}^2(k), \tag{I.40}$$

where $P(k) = |X(k)|^2$ is the spectral power at bin k and α is chosen analytically to guarantee the target P_{FA} [17].

1.9 State of the Art

Spatial and spectral sensing techniques have been widely studied for RF source detection and localisation. In cognitive radio systems, spectrum sensing is a key step for identifying signal occupancy under low SNR conditions. Classical energy detection suffers from the SNR wall phenomenon [13], which motivates adaptive approaches such as CFAR that maintain a constant false alarm rate in non-stationary environments [14].

For spatial localisation, direction-of-arrival estimation is commonly performed using antenna arrays. Classical beamforming methods and high-resolution subspace techniques such as MUSIC and ESPRIT provide accurate angular estimation but rely on stationarity assumptions and covariance estimation [4]. In dynamic scenarios, adaptive beamforming and subspace tracking methods allow continuous updates of spatial information while reducing computational cost [3].

To handle time-varying sources, DoA tracking techniques model the angular evolution as a dynamic state estimation problem. Kalman Filter provides an optimal recursive solution for linear Gaussian systems [9]. For nonlinear models, Extended Kalman Filter introduces a local linearization, while Unscented Kalman Filter improves estimation accuracy by propagating sigma points without requiring Jacobians [18, 19]. These methods are widely used for smoothing noisy DoA estimates and tracking moving emitters over time [8].

1.9.1 AI-Based Approaches for DoA Tracking

Deep learning methods have recently been explored for Direction of Arrival . By framing DoA estimation as a supervised learning problem, neural networks can directly map array measurements or covariance information to angular estimates, offering improved robustness in low-SNR environments [20].

CNN and recurrent architectures have also been proposed for dynamic scenarios [21, 22]. These approaches can implicitly learn array manifold structures and may outperform classical methods in highly non-linear propagation conditions.

However, their applicability in real RF monitoring systems remains limited due to the need for large labelled datasets and prior knowledge of signal and channel characteristics, which are generally unavailable.

1.9.2 AI-Based Approaches for Spectrum Sensing

Deep-learning techniques have also been investigated for spectrum sensing. By learning discriminative features directly from raw data, these approaches can bypass explicit statistical modelling and improve detection performance. However, their applicability remains limited in passive RF monitoring due to the need for annotated datasets and offline training [14].

1.10 Conclusion

This chapter presented the fundamental concepts and state of the art in spectrum sensing, direction-of-arrival estimation, and target tracking. These three fields are not independent: sensing detects, beamforming localises. Together, they form the unified pipeline that drives the architecture developed in this work. The following chapters bridge theory and practice by detailing the algorithms deployed, justifying each design choice against the constraints reviewed here, and discussing their hardware implementation on SDR platforms.

Chapter II

Adaptive spectrum sensing using CMAP-CFAR

Constant False Alarm Rate detectors are widely adopted for their ability to adapt the detection threshold to local noise and clutter statistics without prior knowledge of the absolute background power. However, conventional sliding-window variants (CA-CFAR, OS-CFAR, GO-CFAR) [23] assume a *spatially homogeneous* noise environment: all reference cells are presumed to share the same statistical distribution as the cell under test, this assumption is routinely violated in dense RF environments, where multipath propagation, narrowband interference, and time-varying occupancy patterns produce a highly *non-homogeneous* spectral background that single-snapshot spatial averaging cannot track.

To address this limitation, this chapter introduces the **Spectral Clutter-Map CFAR**, a detection architecture that was, until now, applied exclusively in the radar range-Doppler domain [24, 25] and whose transfer to the FFT-based spectrum sensing context constitutes an original contribution. By replacing the instantaneous spatial estimate with a *recursive temporal map* of the spectral background, the SCMAP-CFAR continuously tracks background fluctuations across successive FFT frames, yielding a threshold that is simultaneously robust to non-stationarity and analytically tied to a target P_{FA}.

The chapter is organised as follows. Section 2.1 recalls the CMAP-CFAR principle in its original radar formulation, detailing the recursive estimator and its closed-form performance expressions. Section 2.2 presents the transposition to the spectral domain and the complete SCMAP-CFAR processing pipeline.

2.1 Clutter-Map CFAR overview

The sliding-window CFAR variants reviewed previously and illustrated in 2.1 from spatially neighbouring cells at a single instant. This strategy is optimal when the clutter[1] statistics are

[1] In radar processing, *clutter* refers to any unwanted echo or interference that competes with the target return ground reflections, weather, jamming. In the spectral domain, its analogue is the residual background power caused by multipath propagation, intermodulation products, or broadband jamming, which distorts the local noise floor across FFT bins.

spatially homogeneous but degrades rapidly in non-stationary environments where the background level fluctuates across scans [24].

Two complementary CFAR strategies have emerged to address this [24]:

- **Spatial averaging:** reference cells from neighbouring range/Doppler bins estimate the local background at a given scan, efficient in homogeneous clutter.
- **Temporal averaging (Clutter Map):** the detector output of each resolution cell is averaged *across successive scans*, building a persistent background map, preferred in non-homogeneous, time-varying environments.

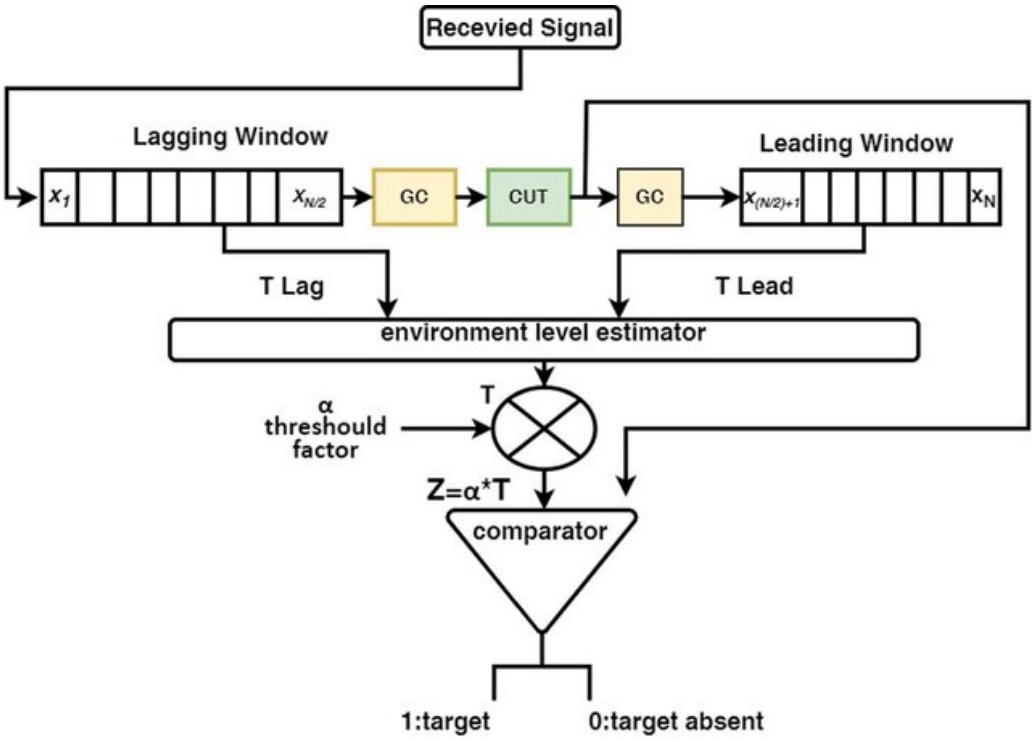


Figure 2.1: Sliding window technique [26]

2.1.1 Clutter Statistical Models

Accurate clutter modelling is a prerequisite for deriving closed-form expressions for P_d and P_{FA}. Three distributions dominate the radar and spectrum sensing literature [25]:

The **Weibull distribution**, parameterised by scale B and shape C, is the most general model, encompassing several limiting cases:

$$f(x) = \frac{C}{B}\, x^{C-1} \exp\left(-\left(\frac{x}{B}\right)^{C}\right). \tag{II.1}$$

For $C = 1$ it reduces to the **exponential distribution** $f(x) = \lambda e^{-\lambda x}$ ($\lambda = 1/B$), the standard model for squared-magnitude FFT bins under AWGN. For $C = 2$ it yields the **Rayleigh distribution**, applicable to the amplitude of a sum of independent scattered returns. The **Gaussian distribution** complements these for thermal-noise-dominated backgrounds [25].

2.1.2 Recursive Clutter Map Estimation

The spatial averaging baseline estimates the background at cell k as a symmetric moving-window integral [24]:

$$\hat{p}(k) = \sum_{m=1}^{M/2} \Big[q(k+m) + q(k-m) \Big]. \tag{II.2}$$

Equation (II.2) treats all reference cells equally and discards temporal history: it is well-suited for static clutter but insensitive to scene evolution.

The CMAP-CFAR replaces this by a **first-order recursive (AR) filter** that accumulates a weighted history of past scans [24]:

$$\hat{y}_n(k) = \lambda\, q_n(k) + (1-\lambda)\, \hat{y}_{n-1}(k), \quad \lambda \in [0,1], \tag{II.3}$$

which, unrolled over M scans, gives an exponentially weighted sum

$$\hat{y}_n(k) = \sum_{l=0}^{M} \lambda\, (1-\lambda)^l\, q_{n-l}(k), \tag{II.4}$$

where $q_{n-l}(k)$ is the received power at scan $n-l$ and cell k. The forgetting factor λ controls the *memory* of the estimator: a small λ retains a long history (slow adaptation), while a large λ reacts quickly to abrupt background changes. Note that equation (II.3) is updated only if no detection is observed.

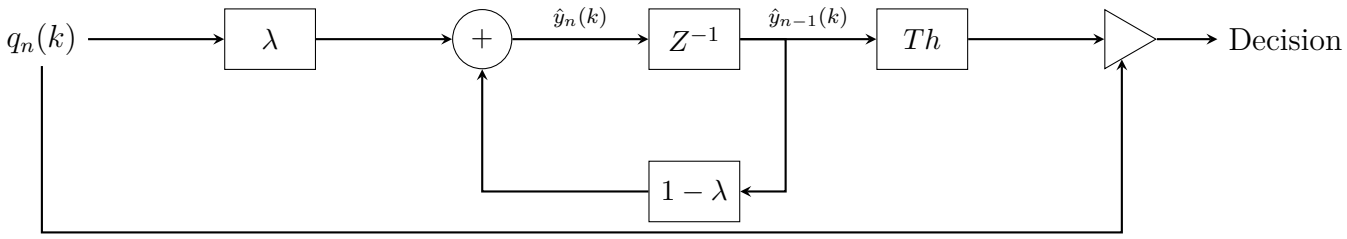


Figure 2.2: Recursive clutter-map filter (EWMA).

The detection decision compares the cell-under-test (CUT) output $q_n(k)$ against the adaptively scaled map estimate:

$$q_n(k) \underset{H_0}{\overset{H_1}{\gtrless}} K \cdot \hat{y}_{n-1}(k), \tag{II.5}$$

where the threshold factor K is chosen to guarantee a target P_{FA}. For a Gaussian clutter model with a single pulse, closed-form expressions exist [23]:

$$P_d = \prod_{l=0}^{M} \frac{1}{1 + \frac{\lambda(1-\lambda)^l\, K}{1+S}} \tag{II.6}$$

$$P_{\mathrm{FA}} = \prod_{l=0}^{M} \frac{1}{1 + \lambda(1-\lambda)^l \cdot K} \tag{II.7}$$

where $S = \sigma_s^2/\sigma_n^2$ is the signal-to-noise ratio. Equation (II.7) can be inverted numerically to obtain K for any desired P_{FA}, providing an analytically grounded threshold.

2.2 Application to FFT-Based Spectrum Sensing

The extension of CMAP-CFAR from the range–Doppler domain to the spectral domain is conceptually direct: each FFT bin plays the role of a radar range cell, and successive FFT frames replace antenna scans. Below the detailed steps, which begins at the RF front-end then spectral estimation before the CFAR detection stage.

Hardware front-end. The KrakenSDR coherent receiver down-converts and samples the RF signal at rate F_s, producing synchronised I/Q[2]samples $x[n]$ across all M channels. In what follows, these samples are assumed available at the input of the processing chain.

Step 1 – Spectral estimation. The buffered block undergoes an N_{FFT}-point FFT, yielding the power spectrum estimate:

$$P[k] = |X[k]|^2 \tag{II.8}$$

Step 2 – Reference cell averaging. The cells adjacent to the CUT are excluded to prevent target self-masking. The remaining N_r reference cells are averaged:

$$Z_n(k) = \frac{1}{N_r} \sum_{j \in \mathcal{R}(k)} P[j] \tag{II.9}$$

Step 3 – CMAP temporal filtering. The clutter-map estimate is updated via a first-order IIR filter:

$$\hat{\sigma}_n^2(k) = \lambda\, \hat{\sigma}_{n-1}^2(k) + (1-\lambda)\, Z_n(k), \quad \lambda \in [0,1] \tag{II.10}$$

where λ governs the filter memory: $\lambda \to 0$ yields instant adaptation, $\lambda \to 1$ freezes the threshold.

Step 4 – Adaptive threshold and detection. The adaptive threshold $T_n(k) = K\, \hat{\sigma}_n^2(k)$ is compared to the CUT power. The set of detected bins is:

$$\mathcal{D}_n = \Big\{ k \mid P[k] > T_n(k) \Big\} \tag{II.11}$$

where K is derived from the target P_{FA}.

Step 5 – Peak selection. If $\mathcal{D}_n \neq \emptyset$, the detected bin is:

$$k^* = \arg\max_{k \in \mathcal{D}_n} P[k] \tag{II.12}$$

Otherwise, the frame is declared empty.

[2]I/Q (In-phase/Quadrature) modulation represents a complex-valued signal using two orthogonal components: an in-phase (I) component and a quadrature (Q) component shifted by 90°. Modulation combines these components onto a carrier, while demodulation recovers the complex baseband signal, preserving both amplitude and phase information. Physically, the complex number does not correspond to two separate physical signals; rather, it is a mathematical representation in which the magnitude represents the signal amplitude and the angle represents its instantaneous phase.

Algorithm 1 CMAP-CFAR spectrum sensing (per-frame iteration)

Init: $\hat{\sigma}_0^2(k) = 0 \ \forall k$, set λ, K, N_{FFT}
for each scan frame n **do**
    // — *Spectral estimation* —
    $P[k] \leftarrow |\mathrm{FFT}\{x[n]\}|^2$
    // — *CFAR background estimation* —
    $Z_n(k) \leftarrow \dfrac{1}{N_r} \sum_{j \in \mathcal{R}(k)} P[j]$
    // — *CMAP temporal filtering* —
    $\hat{\sigma}_n^2(k) \leftarrow \lambda\, \hat{\sigma}_{n-1}^2(k) + (1-\lambda)\, Z_n(k)$
    // — *Detection* —
    $\mathcal{D}_n \leftarrow \left\{ k \mid P[k] > K\, \hat{\sigma}_n^2(k) \right\}$
end for
return $k^* = \arg\max_{k \in \mathcal{D}_n} P[k]$

2.3 Conclusion

This chapter introduces an adaptive framework for spectrum sensing, the spectral CMAP-CFAR designed to decrease the inherent heterogeneity of the frequency domain. By augmenting a standard FFT-based energy detector with a two-stage noise estimation process, this approach merges the instantaneous background rejection of a CA-CFAR window with the temporal stability of a recursive clutter map, governed by the forgetting factor λ.

Chapter III

Robust Beamforming for RF Source Tracking

This chapter presents the adaptive algorithm proposed by Gazor [1] for mobile source tracking. In our application, this method is used for tracking RF signal-emitting source.

Adaptive beamforming has been studied extensively since the 1960s to improve signal extraction. Gazor [1] proposed a robust adaptive beamforming approach based on temporal steering vector adaptation. The method uses a simple LMS-based tracking algorithm to update the steering vector iteratively, using the output of a conventional beamformer. The source parameters (direction of arrival or target coordinates) are then estimated by projecting the corrected steering vector onto the array manifold. These updated parameters refine the beamformer configuration and improve signal extraction.

Gazor [1] presented the method only for the ULA configuration. In this work, we extend it to the UCA case as well.

3.1 Signal Model and Problem Statement

3.1.1 Observation Model

The observation model is defined as:

$$\mathbf{X}_t = \mathbf{A}_t s_t + \mathbf{N}_t$$

where $\mathbf{X}_t \in \mathbb{C}^m$ is the observation vector, s_t is the narrowband signal of interest, $\mathbf{N}_t$ is the additive noise vector, and $\mathbf{A}_t = F(\theta_t)$ is the steering vector characterized by the localization parameter θ_t.

3.1.2 Assumptions

The following assumptions are used throughout this chapter [1] :

H1: $\mathbf{A}_t$, $\mathbf{N}_t$, and s_t are mutually independent.

H2: $\mathbf{A}_t$ varies slowly compared to the signal and the noise.

H3: $\mathbf{A}_t$ follows a Gaussian distribution with mean $\bar{\mathbf{A}}$ and covariance matrix $\mathbf{R}_A$.

H4: $\mathbf{N}_t$ is a zero-mean white Gaussian noise with covariance matrix $\mathbf{R}_N$.

H5: An initial approximate estimate of the angle θ_0, is provided by an estimation technique.

Under these assumptions, the core idea is to minimize the error between the true steering vector $\mathbf{A}$ and its estimate $\hat{\mathbf{A}}$. The steering vector is corrected continuously, and the conventional beamformer is adapted to the updated look direction. Only the observation vector $\mathbf{X}$ is available, and the parametric modeling function F is known [1].

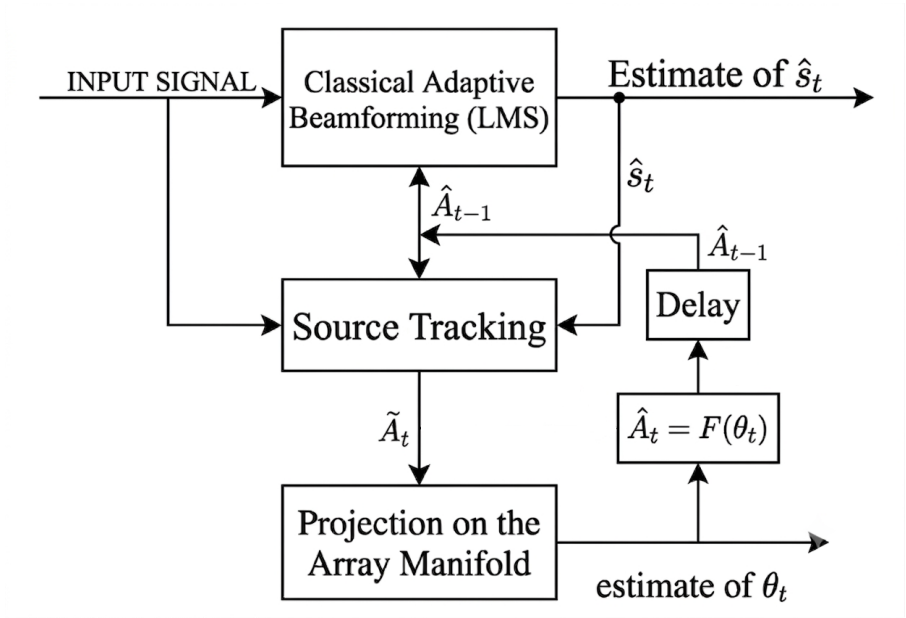


Figure 3.1: Source tracking approach: overall block diagram showing the iterative correction of the steering vector and the beamformer adaptation.

3.2 Identification Problem

Before tracking the DoA itself, the array response associated with the current source must be identified from the observations. This intermediate identification problem provides an estimate of $\mathbf{A}_t$, which is then projected onto the array manifold to recover the physical angle parameters. We already dispose of an initial estimate of the steering response, obtained from a prior estimation technique, which is used to start the recursive tracking procedure.

The identification problem is formulated as follows [1]:

$$(\tilde{\mathbf{A}}_t, \tilde{s}_t) \triangleq \arg \max_{(\mathbf{A}_t,\, s_t)} P\Big(X_t,\, A_t,\, s_t \mid \bar{\mathbf{A}}_t\Big) \tag{III.1}$$

The optimal estimates are:

$$\tilde{\mathbf{A}}_t = \bar{\mathbf{A}}_t + \mathbf{R}_A \left(|\tilde{s}_t|^2 \mathbf{R}_A + \mathbf{R}_N \right)^{-1} \left(\mathbf{X}_t - \bar{\mathbf{A}}_t \tilde{s}_t \right) \tilde{s}_t^H \tag{III.2}$$

$$\tilde{s}_t = \frac{\tilde{\mathbf{A}}_t^H \mathbf{R}_N^{-1} \mathbf{X}_t}{\tilde{\mathbf{A}}_t^H \mathbf{R}_N^{-1} \tilde{\mathbf{A}}_t + \sigma_s^{-2}} \tag{III.3}$$

Using assumptions H2 and H3:

$$\bar{\mathbf{A}}_t \approx \widehat{\mathbf{A}}_{t-1}$$

The two equations corresponding to the optimal estimates $\tilde{A}_t$ and $\tilde{s}_t$ do not admit a simple closed-form algebraic solution. Therefore, $\tilde{s}_t$ is replaced by an alternative estimate, denoted by $\hat{s}_t$, obtained using an adaptive beamforming method (GSC).
Given the linear model:

$$\mathbf{X}_t = \mathbf{A}\hat{s}_t + \mathbf{e}_t \tag{III.4}$$

where $\mathbf{e}_t$ is the estimation error, the optimal $\mathbf{A}$ minimizes the mean squared error [1]:

$$\tilde{\mathbf{A}}_t = \arg\min_{\mathbf{A}} \ \mathbb{E}\left[(\mathbf{X}_t - \mathbf{A}\hat{s}_t)^H (\mathbf{X}_t - \mathbf{A}\hat{s}_t) \right] \tag{III.5}$$

The solution to this problem leads to the LMS-type update which gives a intaneous estimate of the steering vector.

$$\widetilde{\mathbf{A}}_t = \widehat{\mathbf{A}}_{t-1} + \mu \left(\mathbf{X}_t - \widehat{\mathbf{A}}_{t-1} \hat{s}_t \right) \hat{s}_t^H \tag{III.6}$$

where μ is the adaptation step size.

3.3 Projection Stage

Once the steering vector estimate $\widetilde{\mathbf{A}}_t$ is obtained from the identification step, it must be projected onto the array manifold in order to recover the parameter α (e.g., the electrical angle) and the associated steering response. The steering vector is therefore written as a function of this parameter.

$$\mathbf{A} = F(\boldsymbol{\alpha}). \tag{III.7}$$

The array manifold is defined as

$$\Gamma = \{ \mathbf{A} \in \mathbb{C}^m \mid \exists\, \boldsymbol{\alpha} \in \Theta \subset \mathbb{R}^p,\ \mathbf{A} = F(\boldsymbol{\alpha}) \}\,. \tag{III.8}$$

The projection stage consists in finding the parameter vector whose steering response is the closest to the estimate $\widetilde{\mathbf{A}}_t$, i.e.

$$\widehat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha} \in \Theta} d\Big(\widetilde{\mathbf{A}}_t, \mathbf{F}(\boldsymbol{\alpha}) \Big)\,, \tag{III.9}$$

or equivalently

$$d(\widetilde{\mathbf{A}}_t, \Gamma) = \inf_{\mathbf{A} \in \Gamma} d(\widetilde{\mathbf{A}}_t, \mathbf{A}). \tag{III.10}$$

To obtain a practical recursive solution, Gazor linearizes the manifold around a prior estimate $\boldsymbol{\alpha}_0$ using a first-order Taylor expansion:

$$\mathbf{F}(\boldsymbol{\alpha}) \simeq \mathbf{F}(\boldsymbol{\alpha}_0) + \left.\frac{\partial \mathbf{F}(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}}\right|_{\boldsymbol{\alpha}_0} (\boldsymbol{\alpha} - \boldsymbol{\alpha}_0). \tag{III.11}$$

The minimization then reduces to a local least-squares problem, which yields the approximate update

$$\widehat{\boldsymbol{\alpha}}_t \simeq \boldsymbol{\alpha}_0 + \Re\left\{ \left[\left.\frac{\partial \mathbf{F}(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}}\right|_{\boldsymbol{\alpha}_0} \right]^{\#} \left(\widetilde{\mathbf{A}}_t - \mathbf{F}(\boldsymbol{\alpha}_0) \right) \right\}, \tag{III.12}$$

where $(\cdot)^{\#}$ denotes the pseudo-inverse.

This projection step maps the unconstrained estimate $\widetilde{\mathbf{A}}_t$ back onto the array manifold and provides the updated parameter α [1].

3.4 Algorithmic Application for Uniform Linear Arrays

3.4.1 Signal Propagation Model for ULA

Uniform Linear Array consists of m equally spaced antenna elements arranged along a straight line. For a narrowband plane wave arriving from direction θ, the phase shift between adjacent elements is:

$$\Phi = kd\sin(\theta) \tag{III.13}$$

where $k = 2\pi/\lambda$ is the wavenumber, d is the inter-element spacing, and Φ denotes the electrical angle. Note that the capital letter Φ is reserved for the ULA electrical angle, while lowercase ϕ will later be used for the UCA elevation angle.

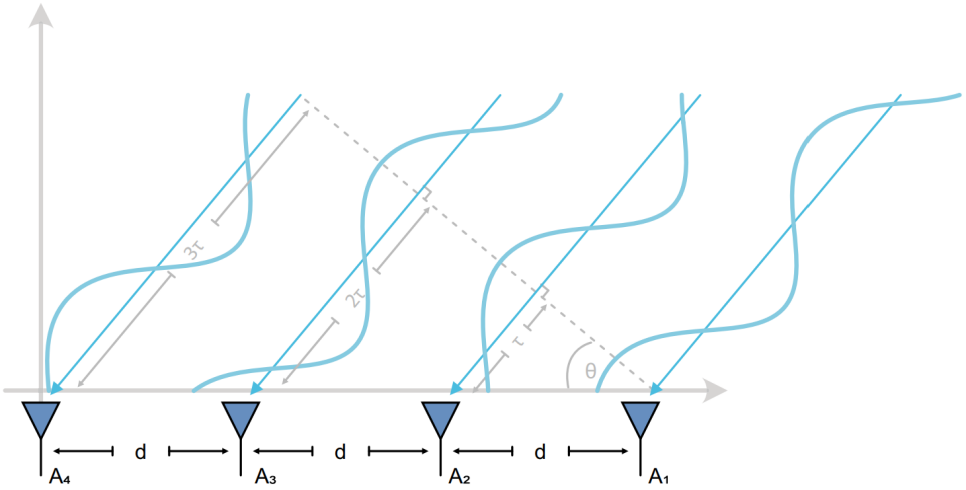


Figure 3.2: Uniform Linear antenna array [27].

The steering vector for electrical angle Φ is:

$$\mathbf{F}(\Phi) = \left[1,\ e^{-j\Phi},\ e^{-j2\Phi},\ \ldots,\ e^{-j(m-1)\Phi}\right]^T \tag{III.14}$$

3.4.2 Adaptive GSC Algorithm for ULA

The tracking algorithm can be interpreted through a GSC structure. The prior steering estimate is first used to align the received channels, and the coherent average of the aligned data provides a first estimate of the source symbol. The residual vector obtained after subtracting this coherent component from each aligned sensor output forms the auxiliary branch, which mainly contains mismatch and noise terms. An adaptive correction term is then estimated from this branch and subtracted from the coherent average to refine $\hat{s}_t$ [1].

It is important to note that the LMS recursion does not update the main steering vector directly. Instead, it adapts the auxiliary weights $\mathbf{W}_t^n$ used to estimate the residual leakage present in the coherent beamformer output. The steering vector itself is updated separately through the recursion on $\widetilde{\mathbf{A}}_t$ [1].

Unlike the classical GSC formulation, which relies on an explicit blocking matrix, Gazor's approach [1] builds the auxiliary branch implicitly through alignment, coherent averaging, and residual subtraction. This avoids an explicit blocking-matrix construction and leads to a lighter recursive implementation.

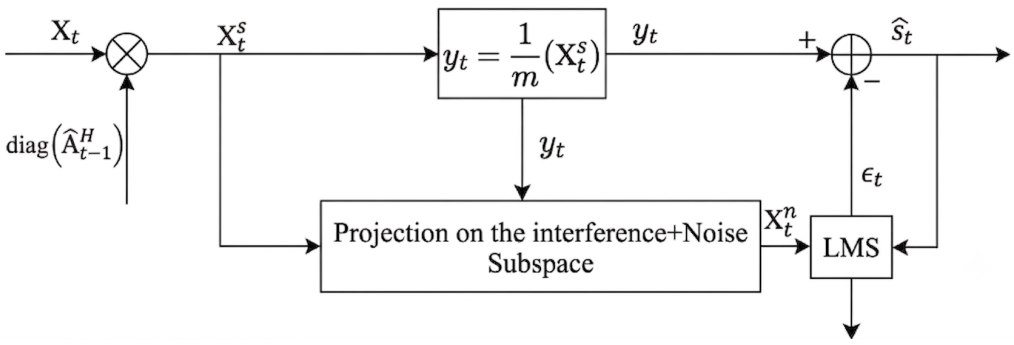


Figure 3.3: Adaptive GSC structure for signal source extration (classical beamforming bloc from 3.1. The main branch performs coherent combining using the prior steering vector. The auxiliary branch captures residual signal, and the LMS updates the auxiliary weights.

Step 1 : Estimate s_t

Given the prior steering vector $\widehat{\mathbf{A}}_{t-1}$, the observation vector is spatially aligned:

$$\mathbf{X}_t^s \triangleq \text{diag}\left(\widehat{\mathbf{A}}_{t-1}^H\right) \mathbf{X}_t \tag{III.15}$$

This phase-compensates each sensor output. The coherent average across sensors is:

$$y_t = \frac{1}{m} \sum_{i=1}^{m} X_{i,t}^s \tag{III.16}$$

The noise subspace component is extracted by removing the coherent part. This is exactly the same behavior as a classical constrained beamformer; but instead of extracting the noise by a subspace method like a simple GSC, here the synchronization and the mean ensure that all information concerning the useful signal is encoded in y_t , in section 3.4.2 we detail the benefits

of this technique:

$$\mathbf{X}_t^n \triangleq \begin{bmatrix} X_{1,t}^s - y_t & \cdots & X_{m,t}^s - y_t \end{bmatrix}^T \tag{III.17}$$

The signal estimate is refined by subtracting an adaptive correction term:

$$\widehat{s}_t = y_t - \varepsilon_t \tag{III.18}$$

where the residual term is obtained by filtering the noise branch:

$$\varepsilon_t \triangleq \mathbf{W}_t^{n\,H} \mathbf{X}_t^n \tag{III.19}$$

The adaptive weights are updated using the LMS rule:

$$\mathbf{W}_{t+1}^n = \mathbf{W}_t^n - \eta\, \widehat{s}_t^*\, \mathbf{X}_t^n \tag{III.20}$$

where η is the LMS step size

It is crucial to distinguish the two LMS rules used here. The first (step μ) acts as a gradient-descent update to track the steering vector $\widehat{\mathbf{A}}_t$, rather than a classical filter. The second (step η) is the standard LMS filter used to minimize residual component (noise and interference) in the GSC branch.. Note again that the LMS updates the auxiliary branch weights $\mathbf{W}_t^n$, not the main steering vector.

Step 2 : Update $\mathbf{A}_t$

After estimating $\widehat{s}_t$, the steering vector is updated through the equation III.6

Step 3 : Projection and Angle Extraction

Once the estimate of the steering vector is ready The electrical-angle update is directly derived from the projection stage (Section 3.3). It takes the form of a phase regression that projects $\widetilde{\mathbf{A}}_t$ back onto the ULA manifold [1]:

$$\hat{\Phi}_{t+1} = \hat{\Phi}_t - \frac{\sum_{i=1}^{m} x_i \,\mathrm{Im}\left(\tilde{A}_{t,i}\, e^{j x_i \hat{\Phi}_t}\right)}{\sum_{i=1}^{m} x_i^2} \tag{III.21}$$

Where x_i is the position of the antennas along the axe,

$$\mathbf{x} = d\left[-\frac{M-1}{2}, -\frac{M-3}{2}, \ldots, 0, \ldots, \frac{M-3}{2}, \frac{M-1}{2}\right].$$

The physical DoA is then recovered as:

$$\hat{\theta}_t = \arcsin\left(\frac{\hat{\Phi}_t}{kd}\right) \tag{III.22}$$

Algorithm 2 ULA GSC-LMS tracking

Init: $\widehat{\mathbf{A}}_0 = \mathbf{F}(\hat{\theta}_0)$, $\mathbf{W}_0^n = \mathbf{0}$, step sizes μ, η
for $t = 1, 2, \ldots, T$ **do**
  // — *GSC signal estimation* —
  $\mathbf{X}_t^s \leftarrow \mathrm{diag}\left(\widehat{\mathbf{A}}_{t-1}^H\right) \mathbf{X}_t$ — *spatial alignment*
  $\widehat{s}_t \leftarrow \frac{1}{m} \sum_{i=1}^m X_{i,t}^s \; - \; \mathbf{W}_t^{n\,H} \left(\mathbf{X}_t^s - \frac{1}{m} \sum X_{i,t}^s \cdot \mathbf{1}\right)$ — *beamformer output*
  $\mathbf{W}_{t+1}^n \leftarrow \mathbf{W}_t^n - \eta\, \widehat{s}_t^* \left(\mathbf{X}_t^s - \frac{1}{m} \sum X_{i,t}^s \cdot \mathbf{1}\right)$ — *LMS weight update*
  // — *Steering vector identification* —
  $\widehat{\mathbf{A}}_t \leftarrow \widehat{\mathbf{A}}_{t-1} + \mu \left(\mathbf{X}_t - \widehat{\mathbf{A}}_{t-1} \widehat{s}_t\right) \widehat{s}_t^H$ — *LMS-like update*
  // — *Manifold projection* —
  $\hat{\theta}_t \leftarrow \hat{\theta}_{t-1} - \dfrac{\sum_{i=1}^m x_i \,\mathrm{Im}\left(\tilde{A}_{t,i}\, e^{j x_i \hat{\Phi}_{t-1}}\right)}{kd \cos(\hat{\theta}_{t-1}) \sum_{i=1}^m x_i^2}$ — *angle extraction*
  $\widehat{\mathbf{A}}_t \leftarrow \mathbf{F}(\hat{\theta}_t)$ — *synchronize*
end for
return $\{\hat{\theta}_t\}_{t=1}^T$

Computational Complexity

Classical subspace-based approaches used to separate noise subspace for the signal may require covariance estimation and matrix decompositions, leading to a complexity of $\mathcal{O}(M^3)$. The algorithm exposed in previous section works in a snapshot-by-snapshot recursive form: the coherent average and the residual branch are built with simple vector operations, and the LMS correction is also a simple recursive update. This significantly reduces the computational burden and leads to a light per-snapshot complexity, approximately linear with the number of sensors m.

3.5 Performance Analysis

3.5.1 ULA Stability

The average of the estimated electrical angle is the output of a first-order low-pass AR filter [1]:

$$E[\hat{\Phi}_t] = E[\hat{\Phi}_{t-1}] \left(1 - \mu\sigma_s^2\right) + \mu\sigma_s^2 E[\Phi_t] \tag{III.23}$$

For convergence, the adaptation step size must satisfy:

$$\mu < \frac{2}{\sigma_s^2} \tag{III.24}$$

The time constant of this filter is:

$$\tau_{E[\Delta\Phi]} = \frac{1}{\mu\sigma_s^2} \tag{III.25}$$

The full derivation is given in [1].

3.5.2 ULA Steady-State MSE

Assume the mobile target follows a motion such that $w_t = \Phi_{t+1} - \Phi_t$ is a white random process independent of s_t, $\mathbf{N}_t$, and θ_t (for a stationary source, $w_t = 0$) [1]. The steady-state covariance of the localization error is:

$$\mathbb{E}\left[\Delta\Phi_\infty^2\right] = \left(\frac{1}{1-\mu\left(\sigma_s^2 + \dfrac{\sigma_N^2}{4m}\right)}\right)\left(\frac{\sigma_w^2}{2\mu\sigma_s^2} + \frac{\mu\sigma_N^2\left(1+\dfrac{1}{m\,\mathrm{SNR}}\right)}{4\sum_{i=1}^m x_i^2}\right) \tag{III.26}$$

The mean squared error (MSE) is defined as:

$$\mathrm{MSE} = \mathbb{E}\left[(\Delta\Phi_\infty)^2\right], \qquad \Delta\Phi_t = \hat{\Phi}_t - \Phi_{t+1} \tag{III.27}$$

The stability condition for the MSE is:

$$\mu < \frac{1}{\sigma_s^2 + \dfrac{\sigma_N^2}{4m}} \tag{III.28}$$

The full derivation of these expressions is available in [1].

3.6 Extension to Uniform Circular Arrays

3.6.1 Signal Propagation Model for UCA

A Uniform Circular Array (UCA) consists of M equally spaced antennas on a circle of radius R. For a narrowband plane wave arriving from azimuth θ and elevation ϕ, the steering vector is:

$$\mathbf{F}(\theta,\phi) = e^{-j\tau}\begin{bmatrix} e^{-jkR\sin(\phi)\cos(\theta-\phi_1)} \\ \vdots \\ e^{-jkR\sin(\phi)\cos(\theta-\phi_M)} \end{bmatrix} \tag{III.29}$$

where $\phi_m = 2\pi(m-1)/M$ is the angular position of the m-th sensor. The 2D tracking model is:

$$\mathbf{X}(t) = \mathbf{F}(\theta(t),\phi(t))\,s(t) + \mathbf{n}(t) \tag{III.30}$$

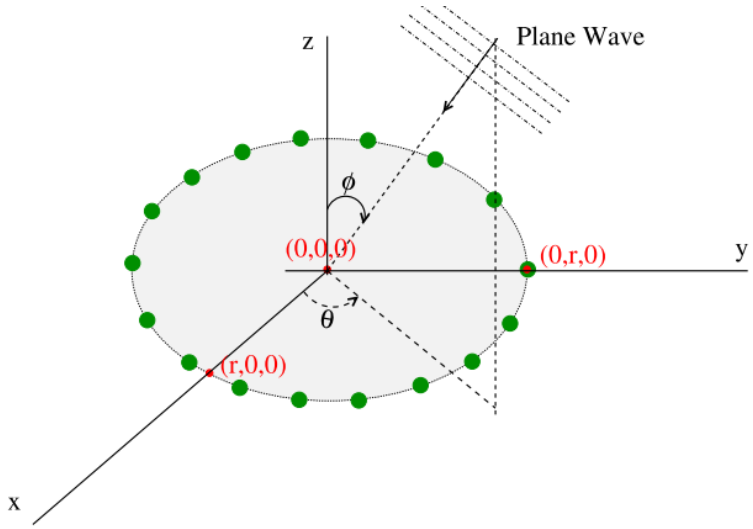


Figure 3.4: Uniform circular array (UCA) [28].

3.6.2 Adaptive GSC Algorithm for UCA

The UCA extension follows the same three-step structure as the ULA case, with the following differences.

What remains identical:

- The spatial alignment using the prior steering vector.
- The coherent averaging and noise subspace extraction.
- The LMS-based weight update of the auxiliary branch.
- The steering vector adaptation rule.

What changes due to circular geometry:

- The steering vector depends on two angles: azimuth θ and elevation ϕ.
- The projection onto the array manifold is a 2D optimization problem.
- The angle extraction simultaneously estimates both θ and ϕ.

The parameter update is formulated as:

$$(\hat{\theta}_t, \hat{\phi}_t) = \arg\min_{\theta,\phi} \left\| \tilde{\mathbf{A}}_t - \mathbf{F}(\theta, \phi) \right\|^2 \tag{III.31}$$

For the phase extraction, the instantaneous phase of the updated steering vector is $\boldsymbol{\psi}(t) = \angle\tilde{\mathbf{A}}(t)$. The phase model for UCA is:

$$\psi_m = -kR\sin(\phi)\cos(\theta - \phi_m) \tag{III.32}$$

This can be linearized as $\boldsymbol{\psi} = \mathbf{H}\mathbf{u}$, where $\mathbf{H} = [\cos(\phi_m), \sin(\phi_m)]$ and $\mathbf{u} = [u_x, u_y]^T$. The least-squares solution gives:

$$\mathbf{u} = -\frac{1}{kR}\mathbf{H}^{\dagger}\boldsymbol{\psi} \tag{III.33}$$

The DOA angles are then recovered as:

$$\hat{\theta} = \operatorname{atan2}(u_y, u_x), \quad \hat{\phi} = \sin^{-1}\left(\sqrt{u_x^2 + u_y^2}\right) \tag{III.34}$$

Algorithm 3 UCA GSC-LMS summary

Init: $\widehat{\mathbf{A}}_0 = \mathbf{F}(\hat{\theta}_0, \hat{\phi}_0)$, $\mathbf{W}_0^n = \mathbf{0}$, step sizes μ, η
for $t = 1, 2, \ldots, T$ **do**
  // — GSC signal estimation —
  $\mathbf{X}_t^s \leftarrow \operatorname{diag}\left(\widehat{\mathbf{A}}_{t-1}^H\right) \mathbf{X}_t$ — *spatial alignment*
  $\widehat{s}_t \leftarrow \frac{1}{M}\sum_{i=1}^{M} X_{i,t}^s \; - \; \mathbf{W}_t^{n\,H}\left(\mathbf{X}_t^s - \frac{1}{M}\sum X_{i,t}^s \cdot \mathbf{1}\right)$ — *beamformer output*
  $\mathbf{W}_{t+1}^n \leftarrow \mathbf{W}_t^n - \eta\, \widehat{s}_t^*\left(\mathbf{X}_t^s - \frac{1}{M}\sum X_{i,t}^s \cdot \mathbf{1}\right)$ — *LMS weight update*
  // — Steering vector identification —
  $\widetilde{\mathbf{A}}_t \leftarrow \widehat{\mathbf{A}}_{t-1} + \mu\left(\mathbf{X}_t - \widehat{\mathbf{A}}_{t-1}\widehat{s}_t\right)\widehat{s}_t^H$ — *LMS-like update*
  // — 2D manifold projection —
  $\boldsymbol{\psi}_t \leftarrow \angle\widetilde{\mathbf{A}}_t, \quad \mathbf{u} \leftarrow -\frac{1}{kR}\mathbf{H}^{\dagger}\boldsymbol{\psi}_t$ — *phase-based LS fit*
  $\hat{\theta}_t \leftarrow \operatorname{atan2}(u_y,\, u_x), \quad \hat{\phi}_t \leftarrow \arcsin\left(\sqrt{u_x^2 + u_y^2}\right)$ — *angle extraction*
  $\widehat{\mathbf{A}}_t \leftarrow \mathbf{F}(\hat{\theta}_t, \hat{\phi}_t)$ — *synchronize*
end for
return $\{(\hat{\theta}_t,\, \hat{\phi}_t)\}_{t=1}^T$

3.7 Conclusion

This chapter presents a robust adaptive beamforming and source-tracking approach based on an LMS-like algorithm and a Generalized Sidelobe Canceller architecture for signal extraction. The method was described for both ULA and UCA configurations.

The key points are:

- The steering vector is updated iteratively using an LMS-type rule, and the source signal is estimated via the GSC structure.
- The estimated steering vector is projected onto the array manifold to extract the electrical angle and the spatial position.
- The implicit blocking mechanism avoids explicit blocking-matrix construction and subspace decomposition, leading to a light per-snapshot complexity approximately linear with the number of antennas.
- Stability is governed by the step size μ, which must satisfy $\mu < 2/\sigma_s^2$ for the mean and $\mu < 1/(\sigma_s^2 + \sigma_N^2/4m)$ for the MSE. For the ULA configuration.
- The steady-state MSE depends on the SNR, the array geometry, and the target motion speed.

Chapter IV

Simulations and Performance Analysis

This chapter presents the simulation conducted to validate the two main algorithms developed in this work.

Both spectral and spatial algorithms presented in 1, 2 and 3 are evaluated under a wide range of simulated conditions using MATLAB-generated signals. Due to its high computational cost, the Monte Carlo approach is employed only for the spectral system performance assessment, where reliable estimates of P_d and P_{fa} are required. For the tracking algorithms, dynamic source trajectories are generated through time-varying steering vectors and immersed in additive white Gaussian noise for both ULA and UCA configurations.

4.1 CMAP-CFAR Spectrum Sensing Simulations

4.1.1 Simulation conditions and proof of concept

The first simulation scenario executed to evaluate the SCMAP-CFAR 1 consists of a narrowband signal at $f_0 = 433,150\,\text{MHz}$ that alternately appears and disappears at $\text{SNR} = 20\,\text{dB}$ with $\lambda = 0.95$ and $N_{ref} = 32$, $\text{K} = 7, 13$ to initially demonstrate its global behavior.

The FFT size is set to 1024 and the selected bandwidth is 2.4 MHz, covering the range $[433 - 1.2\,\text{MHz}; 433 + 1.2\,\text{MHz}]$. This specific band is chosen for consistency with the KrakenSDR hardware utilized in later stages of this work. This configuration achieves a frequency resolution of 2.34 kHz which is sufficient for detecting common practical signals. Consequently, a 1024-point FFT represents an optimal compromise between computational complexity and resolution.

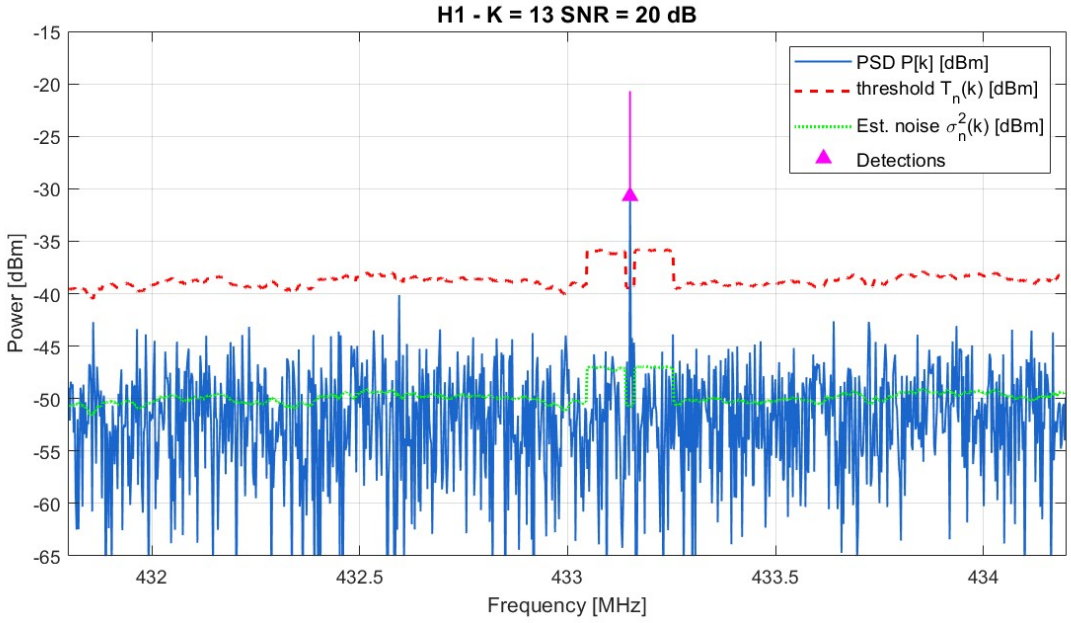


Figure 4.1: Proof of concept I: Example of a successful signal detection under the H_1 hypothesis.

The PSD in Figure 4.1 shows that the SCMAP-CFAR detector correctly identifies the SOI by dynamically raising the threshold $T_n(k)$ above the local noise floor estimate $\sigma_n^2(k)$.

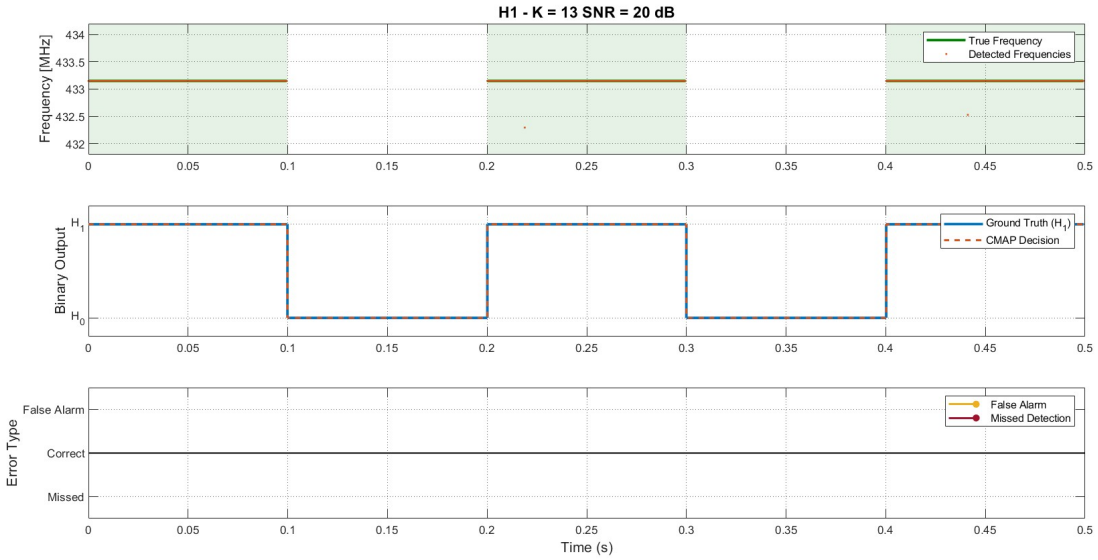


Figure 4.2: Proof of concept II: Temporal tracking and binary decision output matching the ground truth.

Figure 4.2 details the temporal tracking performance under the same configuration. The top and middle plots show that the estimated frequencies and the binary decision flag align with the ground truth transitions. As confirmed by the bottom panel, this threshold factor yields zero missed detections during H_1 intervals and zero false alarms during H_0 intervals.

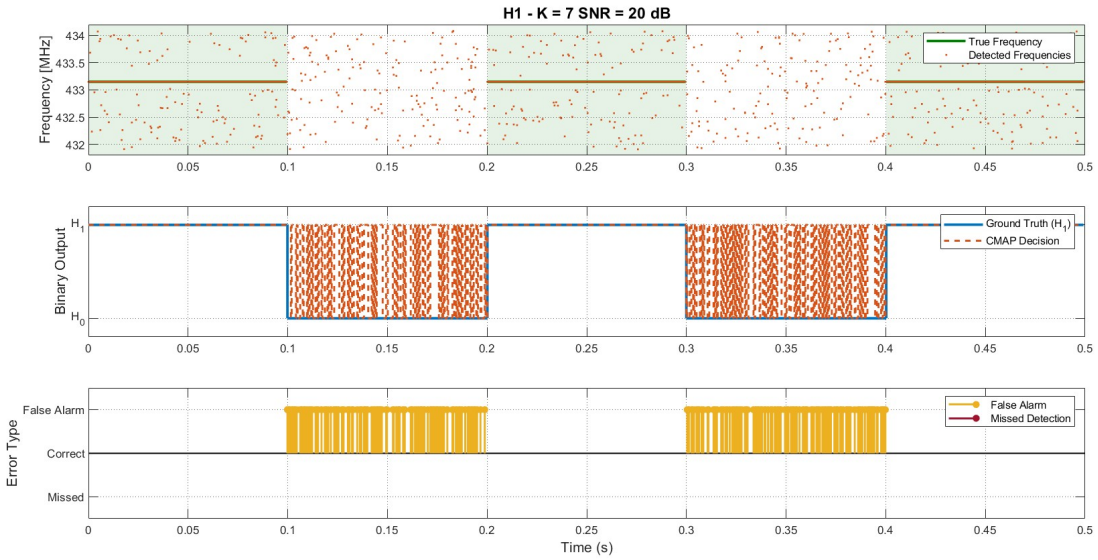


Figure 4.3: Threshold degradation effect: Binary decision contamination by false alarms due to an underestimated threshold factor.

Figure 4.3 illustrates the performance degradation when the threshold factor is set too low. While the signal remains successfully detected during the active H_1 periods, the inactive H_0 intervals are heavily contaminated by false alarms. Because the threshold is insufficient, noise fluctuations frequently cross the detection limit, resulting in rapid binary oscillations and an elevation of the false-alarm probability P_{fa}.

4.1.2 Detection Performance as a Function of SNR

The key performance indicator for a spectrum sensing algorithm are the probability of detection, the probability of false alarm wich are respectively defined as follow:

$$\mathrm{P}_d = P\left(\hat{H} = H_1 \mid H_1\right), \qquad P_{fa} = P\left(\hat{H} = H_1 \mid H_0\right).$$

The simulations are conducted by sweeping the SNR from -30 dB to $+25$ dB using 100 Monte Carlo trials for each configuration. Filter parameter λ is fixed at 0.95. The range from -30 dB to 5 dB represents the region where the detector's transition behaviour occurs and where precise tuning is critical. Beyond 5 dB, the signal-to-noise ratio is high enough that P_d deterministically reaches 1, making that region trivial for performance differentiation.

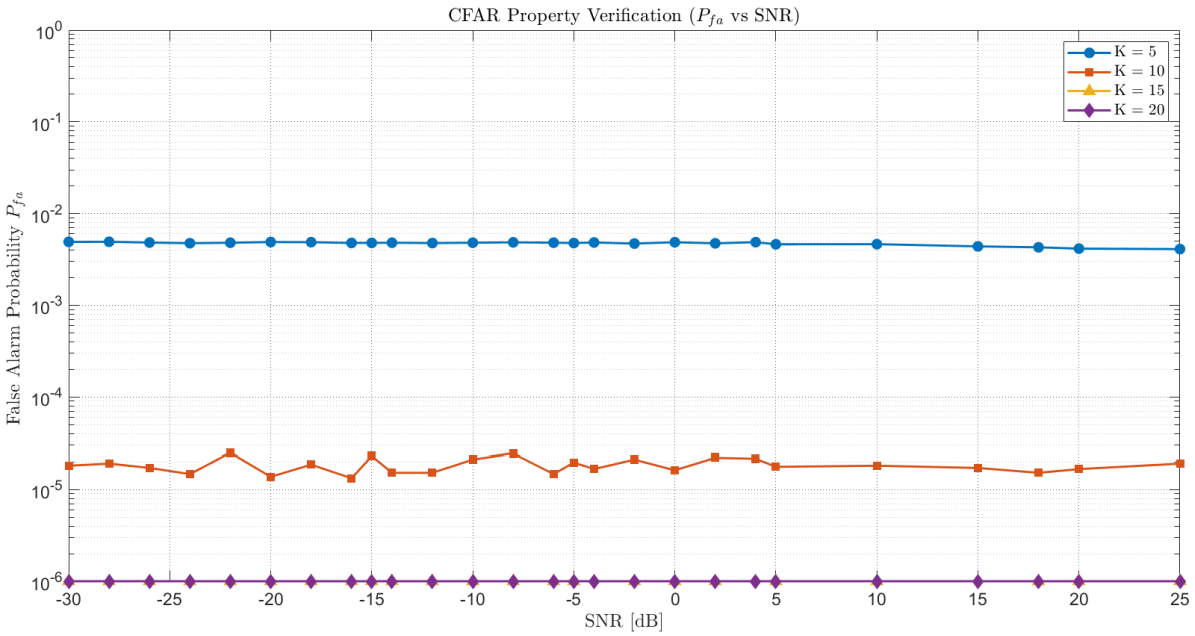


Figure 4.4: False-alarm probability P_{fa} as a function of SNR for different threshold values.

The first validation, consists in verifying the invariance of P_{fa} with respect to the noise power level. Figure 4.4 demonstrates this constant false-alarm rate property, as the curves remain completely flat across the entire SNR sweep for all evaluated values of K. Higher values of the threshold parameter K result in a lower false-alarm rate.

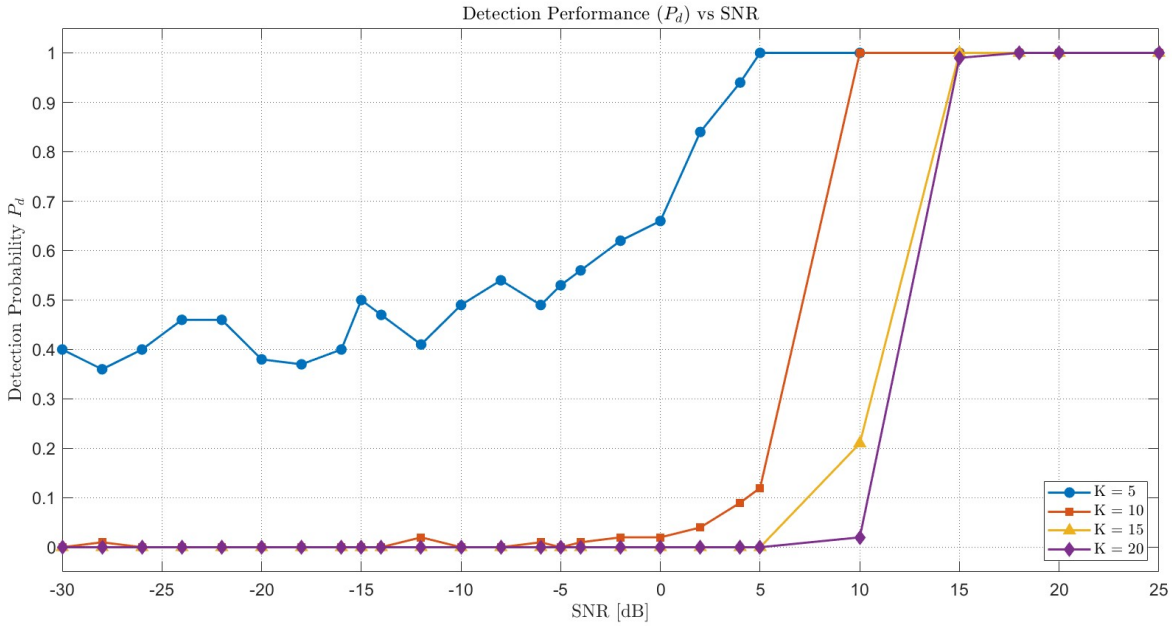


Figure 4.5: Detection probability P_d as a function of SNR for threshold values $K \in \{5, 10, 15, 20\}$.

Figure 4.5 illustrates the evolution of P_d across the SNR range. For $K = 5$, the low threshold permits a non-negligible detection probability (≈ 0.4) even at very low SNRs, though this comes at the expense of an elevated false-alarm rate. For higher thresholds ($K \in \{10, 15, 20\}$), P_d remains near 0 at low SNR and rises toward 1 crossing the 5 dB value.

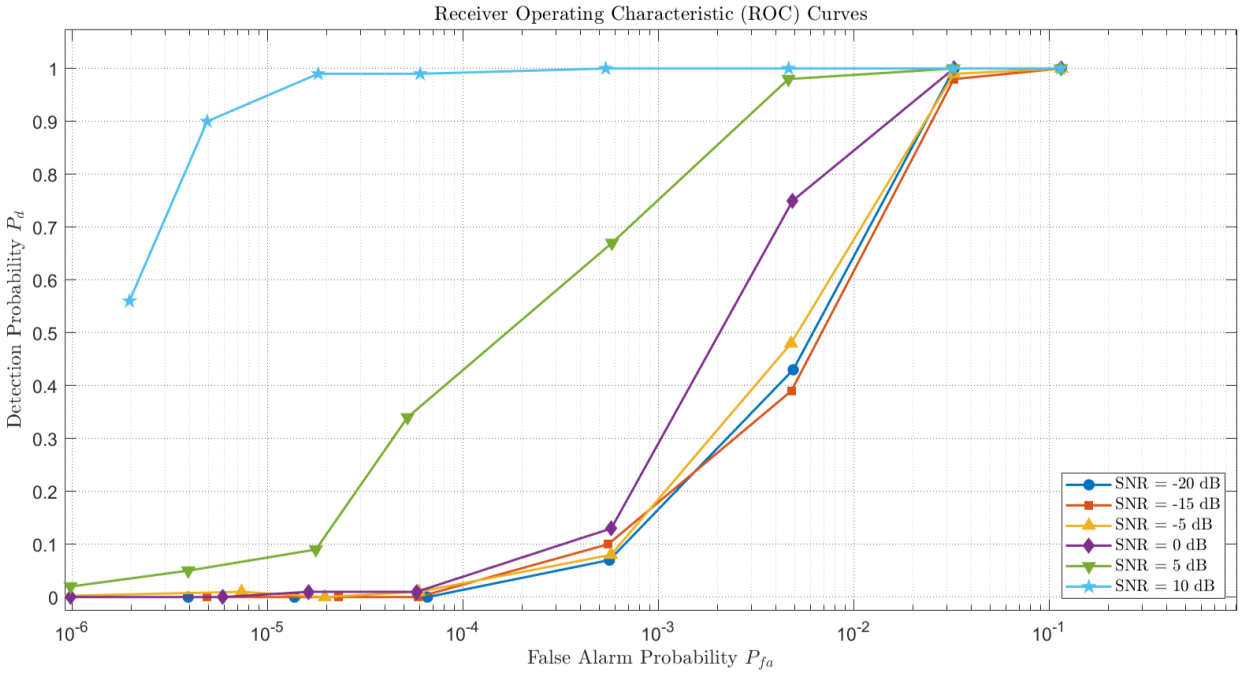


Figure 4.6: ROC curves P_d as a function of P_{fa} for different noise conditions.

From the figure 4.6 we clearly notice that as the SNR increases, The ROC[1] curves shift toward the upper-left corner, indicating better detection performance for any given false-alarm constraint. These curves serve as a practical reference for selecting the optimal threshold parameter K based on the operational requirements of the target application. From these graphics, an operating point is identified and used as a reference for the following analyses. The variations of P_d and P_{fa} are then used to determine the remaining parameter values that satisfy the system requirements.

4.1.3 Impact of the Detection Threshold K

The detection threshold factor K directly dictates the operational trade-off between the system's sensitivity and its selectivity.

$$P(n) > K\,\hat{\sigma}^2(n) \implies \hat{H} = H_1, \tag{IV.1}$$

where $P(n)$ represents the power of the cell under test (CUT) and $\hat{\sigma}^2(n)$ is the estimated noise power.

To evaluate how K affects the system under varying signal intensities, we variated both the detection probability (P_d, left axis) and the false alarm probability (P_{fa}, right axis) across ultra-low,and high SNR regimes, the only fixed parameters fixed here are $\lambda = 0.95$ and the number of reference cell 32.

Results reveal a dependency between the input SNR and the flexibility of the threshold configuration, highlighted by two distinct operational behaviors:

[1]ROC (Receiver Operating Characteristic) curves represent the trade-off between the probability of detection (P_d) and the probability of false alarm (P_{fa}) for a given detector.

- **Low SNR Regime:** At SNR $= -20\,\text{dB}$, only $K = 3$ satisfies both $P_{fa} \leq 0.05$ and $P_d = 1$. Threshold tuning therefore becomes highly critical, with practically no margin for error.

- **High SNR Regime:** On the other hand, when the signal energy dominates the environment, the dependency on a single critical point is relaxed over a wide range. The false alarm constraint can be reduced to (10^{-4}) and remains safely satisfied for a large interval of K.

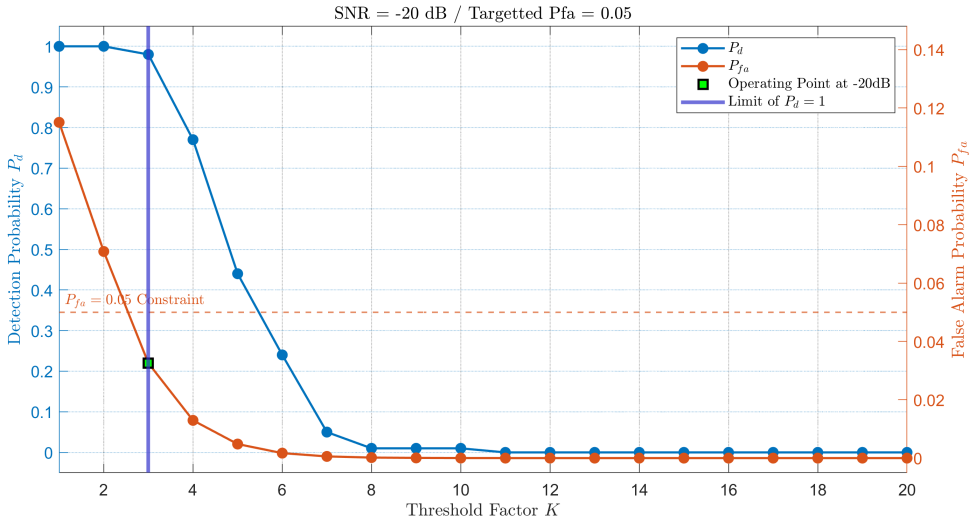


Figure 4.7: P_d and P_{fa} as a function of the threshold factor K for low SNR conditions ($P_{fa} = 0.05$, $\lambda = 0.95$).

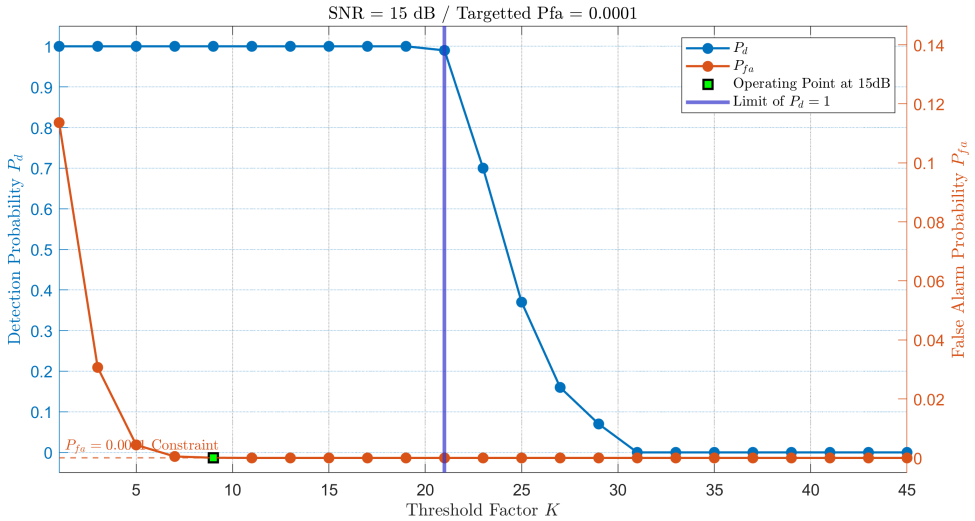


Figure 4.8: P_d and P_{fa} as a function of the threshold factor K for low and high SNR conditions ($P_{fa} = 0.0001$, $\lambda = 0.95$).

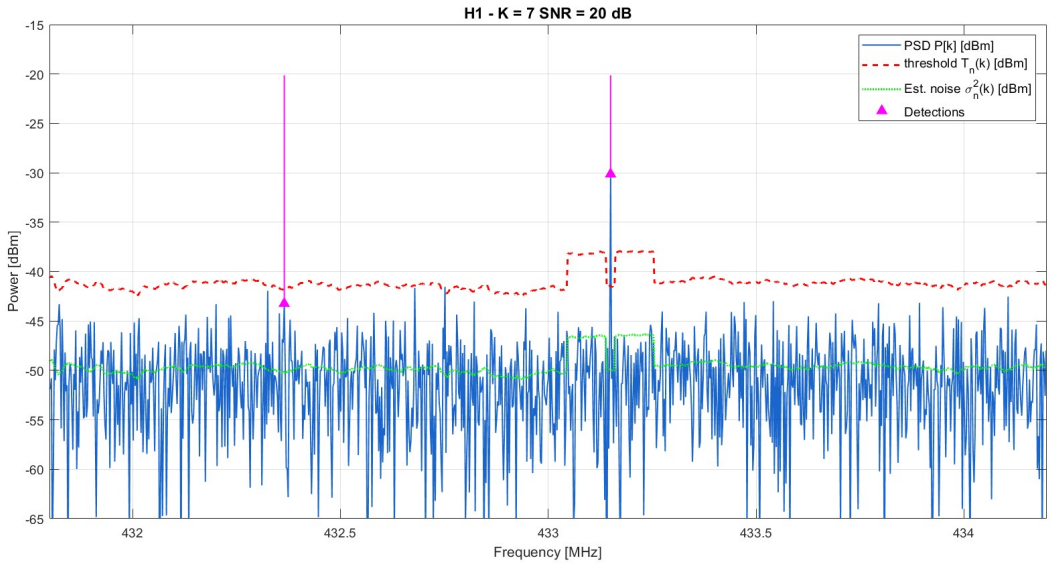


Figure 4.9: Spectral analysis illustrating a single isolated false alarm event across the 1024 FFT bins ($K = 7$, SNR $= 20\,\text{dB}$, $\lambda = 0.95$).

Only one false detection was observed among the 1024 frequency bins($P_{fa} \approx 9.77 \cdot 10^{-4}$.) This estimated P_{fa} agrees with the detection performance obtained in the previous experiments.

Upper Bounds of the system at high SNR : At high SNR, the threshold can be chosen in a wide range, but there is still a clear limit. This limit is shown by the blue vertical line in 4.8, which marks the maximum value of $P_d = 1$.

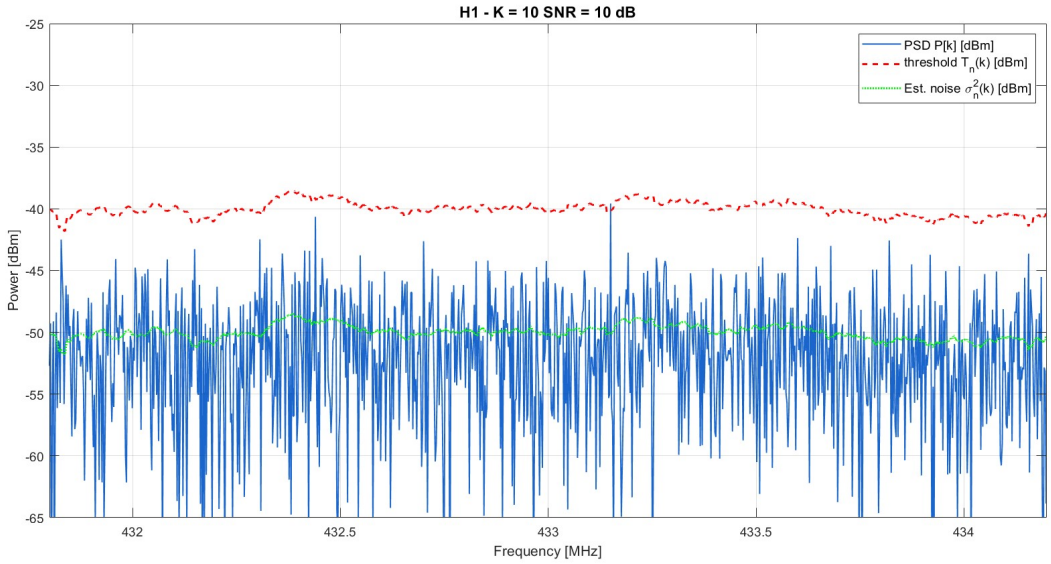


Figure 4.10: Effect of over-thresholding. Signal of interest is completely missed.

Exceeding this upper bound ($K > 10$ at $10\,\text{dB}$, or $K > 20$ at $15\,\text{dB}$) forces the system to scale past the peak power of the received signal as shown in the figure 4.10. Beyond this point, the target component is categorized as part of the background environment, causing a rapid decrease of P_d directly down to 0.

4.1.4 Robustness under Narrowband Interference and non Gaussian environment

1. Robustness to Non-Gaussian Impulsive Noise

In practice, RF receivers are exposed to impulsive interference caused by switching electronics, power lines, or co-channel bursts. Such environments are commonly modelled by a Bernoulli-Gaussian mixture [29], in which a sparse sequence of high-amplitude impulses is superimposed on a background Gaussian noise floor. Formally, the noise at sample n is:

$$b[n] = \underbrace{w[n]}_{\text{AWGN}} + \underbrace{u[n] \cdot g[n]}_{\text{impulsive}} \tag{IV.2}$$

where $w[n] \sim \mathcal{CN}(0, \sigma_w^2)$, $u[n] \sim \mathcal{B}(\lambda_p)$ is a Bernoulli random variable with impulse probability λ_p, and $g[n]$ is a zero-mean complex Gaussian impulse. In the simulation, $\lambda_p = 0.05$, meaning that 5% of samples carry a high-energy impulse.

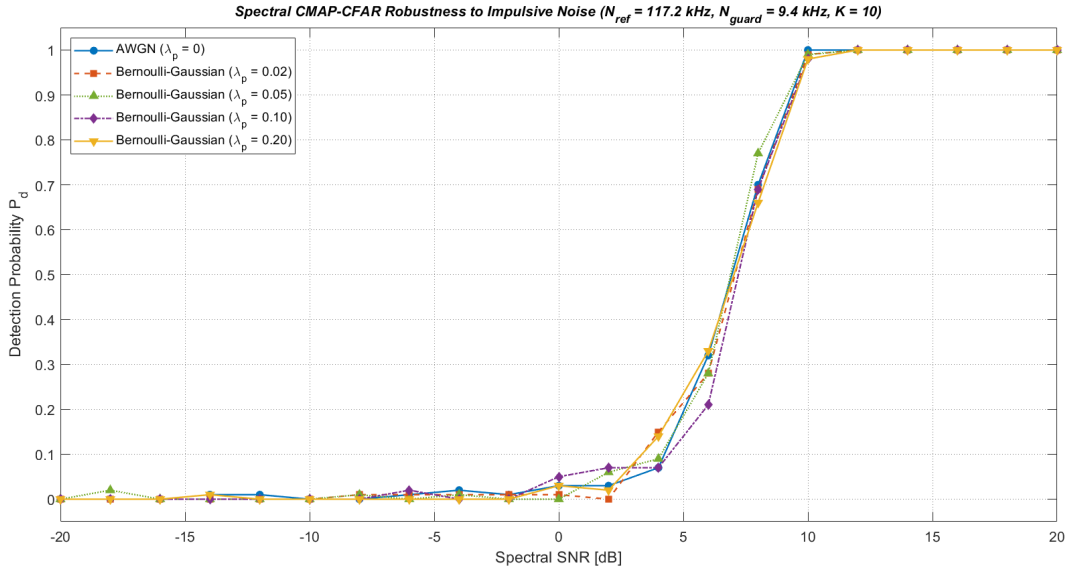


Figure 4.11: Detection probability versus SNR under AWGN and Bernoulli-Gaussian impulsive noise.

Figure 4.11 compares the detection probability under pure AWGN and under Bernoulli-Gaussian impulsive noise, The curves overlap over the entire SNR range, confirming that the SCMAP temporal filter effectively averages out sporadic impulses and that the threshold adaptation mechanism remains unaffected by non-Gaussian perturbations.

2. Interference and Masking effect Analysis

Figure 4.12(a) shows the effect of the frequency separation between the SOI and the interferer. Detection remains optimal when both signals overlap and is fully recovered once the interferer leaves the reference window. However, when the interferer falls inside the reference cells, the background estimate is corrupted and the target becomes masked.

Figure 4.12(b) illustrates the influence of the reference window size. Small windows can exclude the interferer and preserve detection performance, whereas larger windows are more

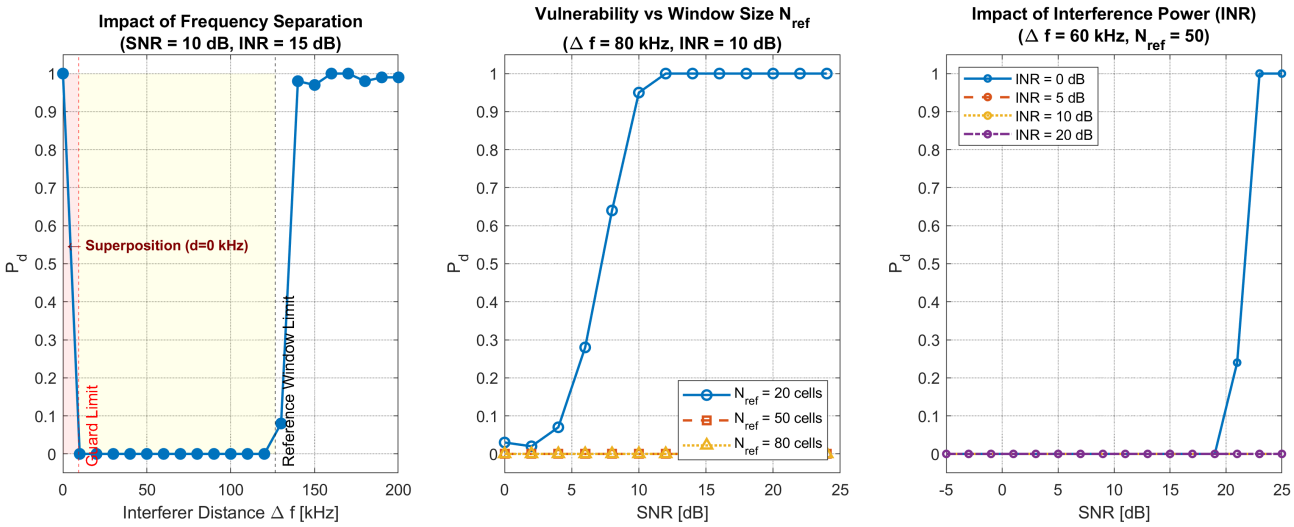


Figure 4.12: (a) effect of frequency separation (Δf), (b) effect of reference window size (N_{ref}), and (c) effect of interference power (INR), $\lambda = 0.95$.

likely to include the interfering signal, leading to target masking.

Figure 4.12(c) evaluates the impact of interference power. Without significant interference, detection improves with SNR. In contrast, when an interferer is present inside the reference window, even moderate INR values are sufficient to suppress detection. **These results confirm that the interferer location is more critical than its power level**.

Having characterized the detector's robustness to interference, we now examine the influence of the recursive filter parameter λ and justify our choice of 0.95 in the previous sections.

4.1.5 Influence of the Filter Parameter λ

The filter parameter λ control the memory depth of the noise level estimator. To evaluate its influence on the system, we performed a sweep from 0.05 to 1 under three distinct noise power (SNR = 20 dB, 10 dB, and 5 dB), while maintaining a fixed threshold factor.

The choice of λ has no impact when the signal-to-noise ratio is high.

But, when the signal intensity degrades, selecting an appropriate value for λ becomes critical. While P_{fa} maintains its constant property, the detection probability P_d begins to fluctuate, dropping as low as 0.4 under the 5 dB condition.

From the last two figures, we identify a region referred to as the *high-bias zone*, spanning $\lambda \in [0.6, 0.9]$, the reasons behind this phenomena are:

- **Excessive Memory Effect:** Large values of λ give a dominant weight to past observations. As a result, the clutter map evolves very slowly and retains information for a long period.

- **Biased Noise Estimation:** Under low-SNR conditions, the local noise floor may fluctuate significantly from one observation to another. Because the estimator reacts slowly, its output no longer reflects the current environment but remains influenced by past measurements. This produces a systematic estimation bias that degrades threshold adaptation and leads to the observed drops in P_d.

Therefore, to avoid this uncertain operating region and ensure stable detection performance, the filter parameter was fixed at $\lambda = 0.95$ in all SCMAP-CFAR simulations.

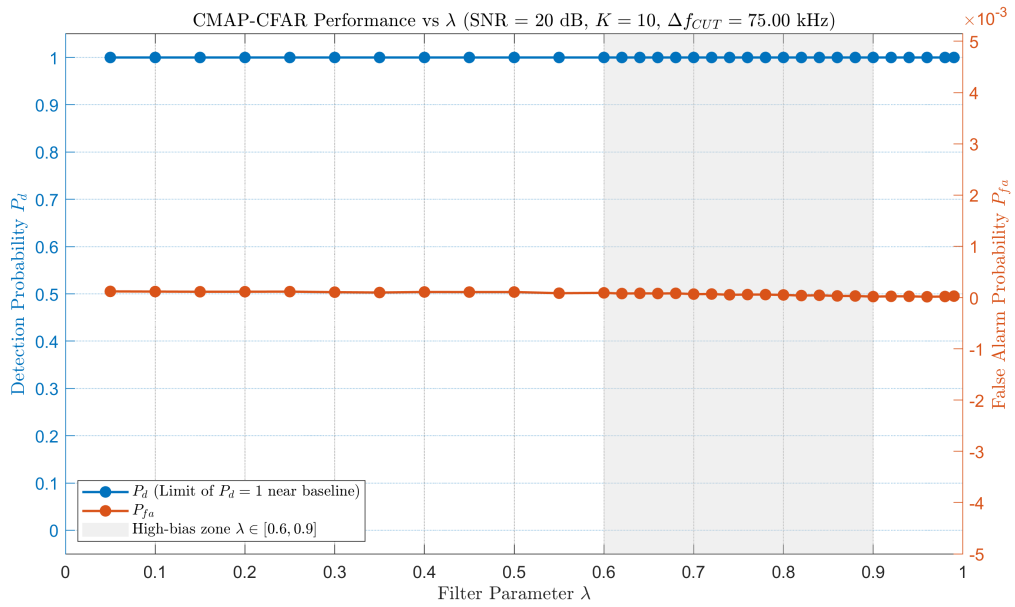


Figure 4.13: P_d and P_{fa} as a function of λ in a high-SNR regime (SNR $= 20$ dB, $K = 10$).

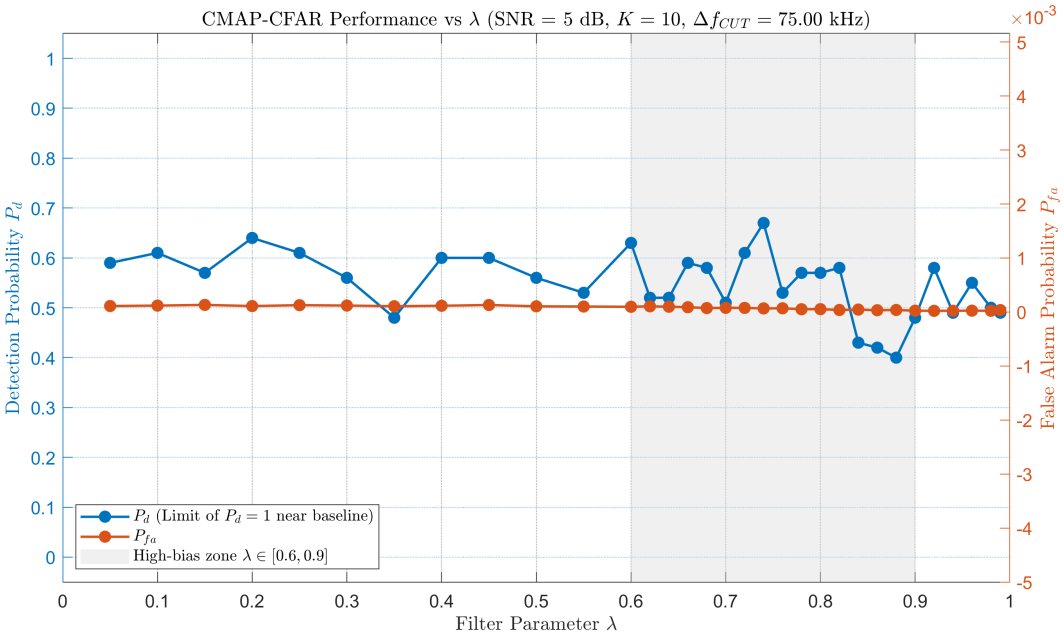


Figure 4.14: P_d and P_{fa} as a function of λ for ($SNR = 5$dB, $K = 10$).

Conclusion: The CFAR-enhanced energy detector provides robust performance in non-stationary environments. Its behavior is mainly governed by the reference-window configuration, the memory parameter λ, and the threshold factor K.

From the pervious analysis we deduce also that proper tuning of λ and K is particularly important under low-SNR conditions.

4.2 Source Tracking Simulations for Uniform Linear Arrays

This section delivers an evaluation of the adaptive tracking algorithm summarized in 2 using ULA.
To assess the algorithm's efficiency, two key performance indicators are monitored throughout the analysis: Convergence time and the mean-square-error.

4.2.1 Azimuth Tracking simulation

The tracking scenario evaluates a time-varying azimuth trajectory $\theta(t)$ under nominal simulation parameters.

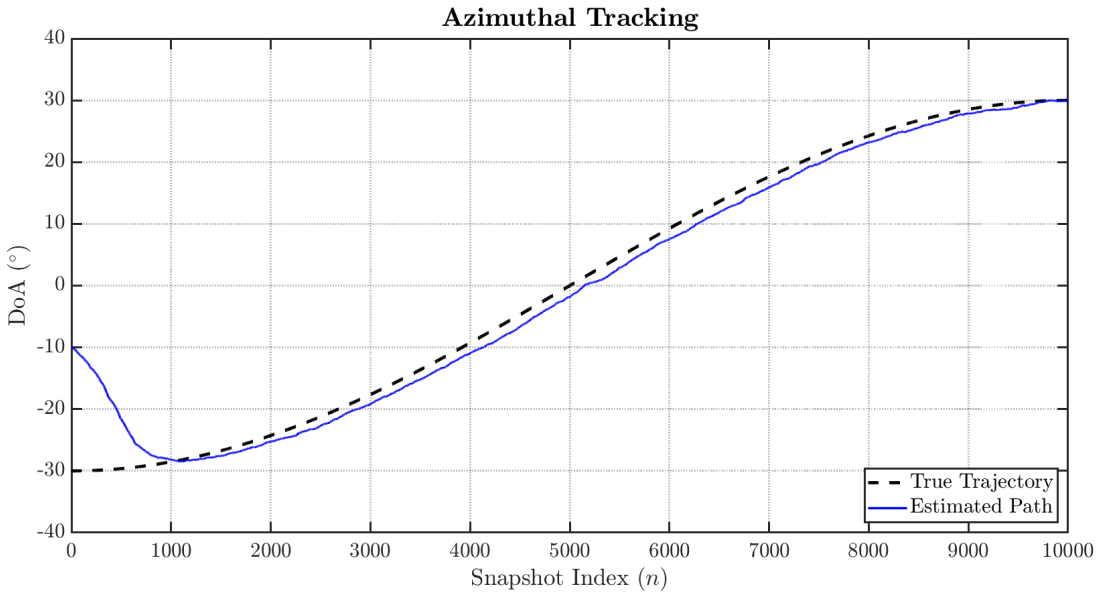


Figure 4.15: Initial demonstration of azimuth tracking execution on a 5-element ULA.

.

As shown by the tracking profile, the algorithm first undergoes a convergence phase before reaching a steady-state tracking regime. As discussed in Chapter 3, convergence mainly depends on the adaptation step size μ and the initialization angle, which must lie within the locking interval. A larger μ speeds up convergence but generally increases the steady-state error. The next section analyzes these trade-offs and the main performance limits of the proposed tracker.

4.2.2 Impact of the Adaptation Step Size μ

The adaptation step size μ is the main parameter governing the algorithm behaviour. It controls the trade-off between convergence speed, tracking responsiveness, and estimation accuracy. Small values provide smooth but slow adaptation, whereas large values improve responsiveness at the cost of increased fluctuations and potential instability. So, selecting μ requires finding a balance between tracking performance and stable operation.

To systematically map this behaviour under various environmental conditions, Figures 4.16, 4.17, and 4.18 illustrate the azimuthal tracking profiles for a grid of $\mu \in [0.0005, 0.5]$ across ultra-low (-10 dB), intermediate (0 dB), and high (10 dB) SNR regimes, respectively.

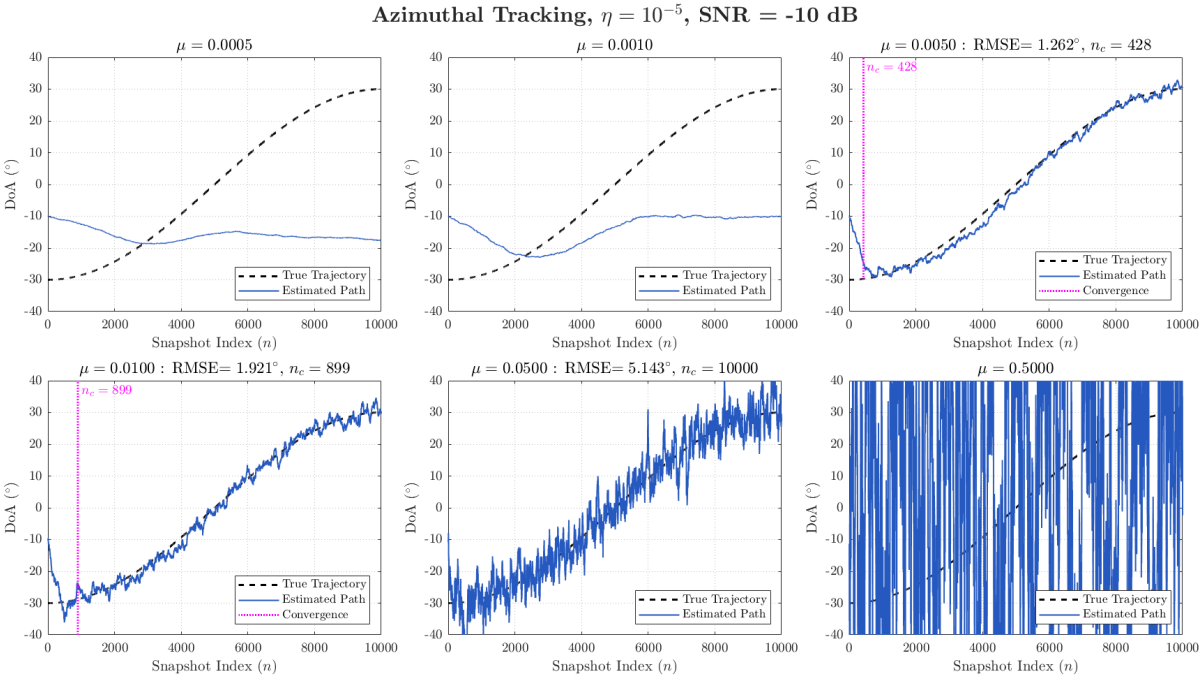


Figure 4.16: DoA tracking performance grid under severe noise conditions (SNR $= -10$ dB) for varying step sizes μ. The vertical magenta dotted line denotes the empirical convergence snapshot index n_c.

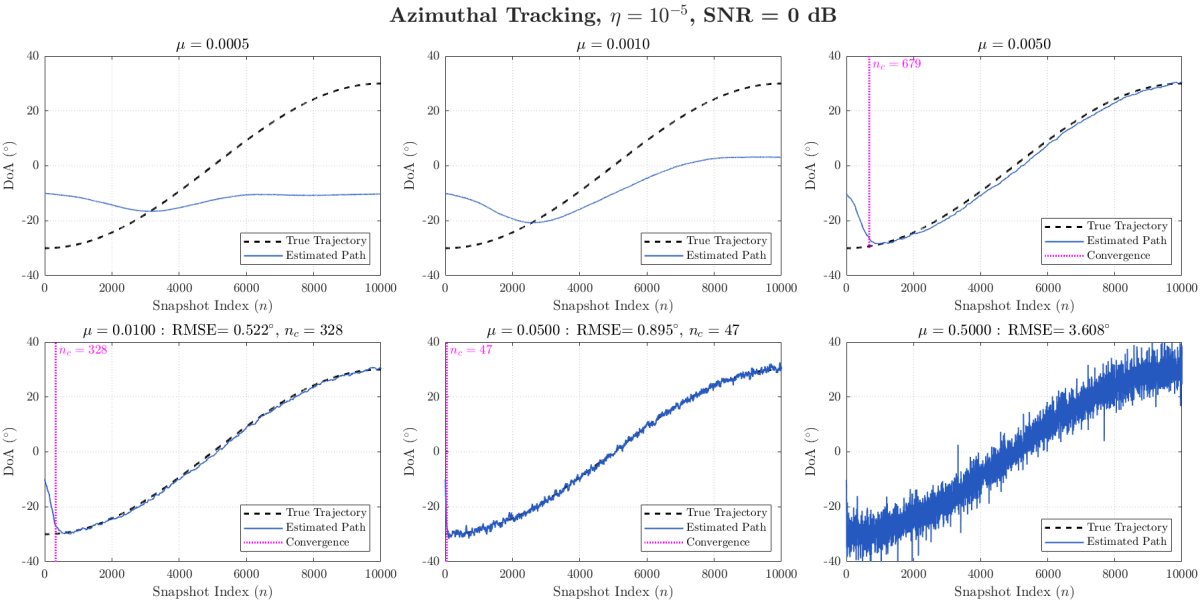


Figure 4.17: DoA tracking performance grid under intermediate noise conditions (SNR $= 0$ dB).

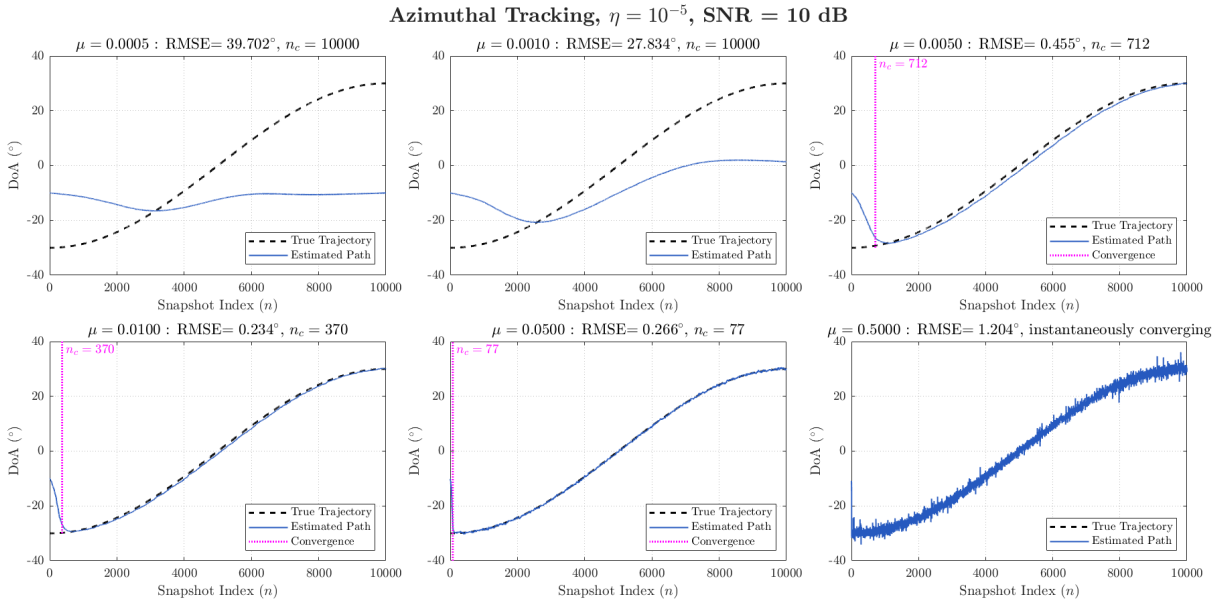
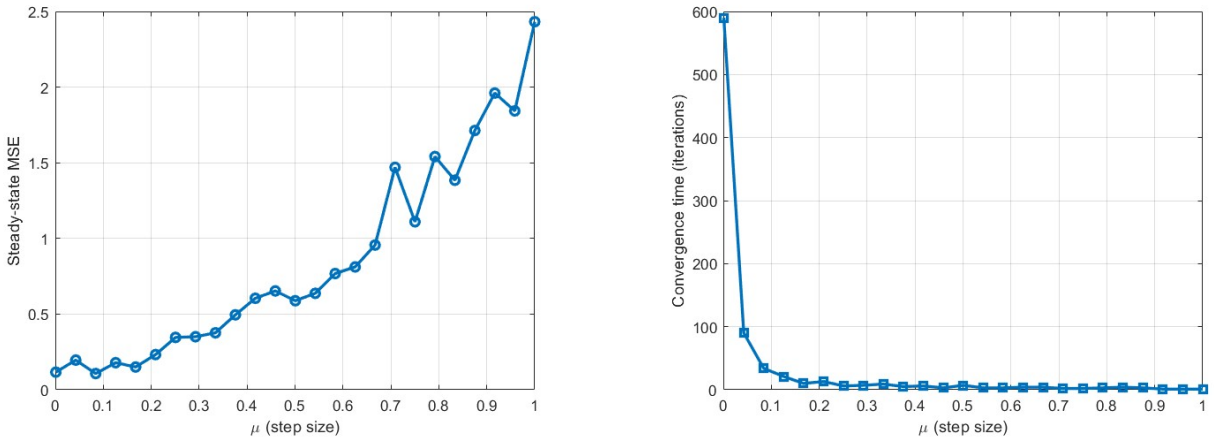


Figure 4.18: DoA tracking performance grid under high signal integrity (SNR = 10 dB).

At low SNR, only a limited range of μ provides reliable tracking, whereas higher SNR conditions allow a wider range of stable values. Figure 4.19 further shows that an operating point around $\mu \approx 0.05$ achieves a good balance between convergence time and steady-state error, consistent with the results of Gazor [1].



(a) Steady-state MSE vs μ.

(b) Convergence time vs μ.

Figure 4.19: Summary of the influence of the adaptation step-size μ at SNR = 10 dB.

4.2.3 Impact of the LMS Step Size η

While μ controls the source tracking process, η controls the adaptation of the LMS filter weights. Figure 4.20 shows that the system mainly depends on whether η remains below or above a stability limit.

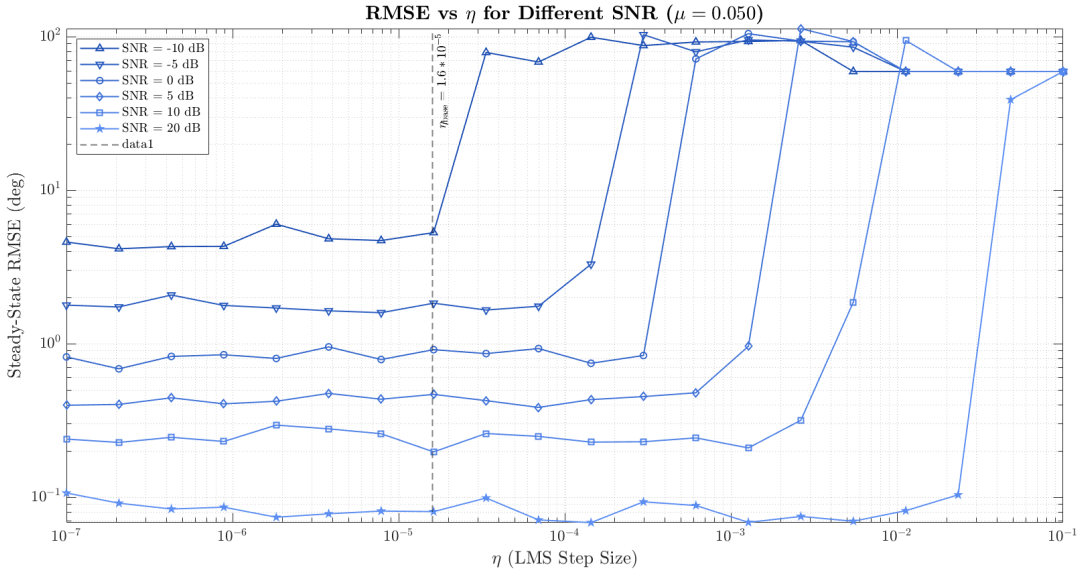


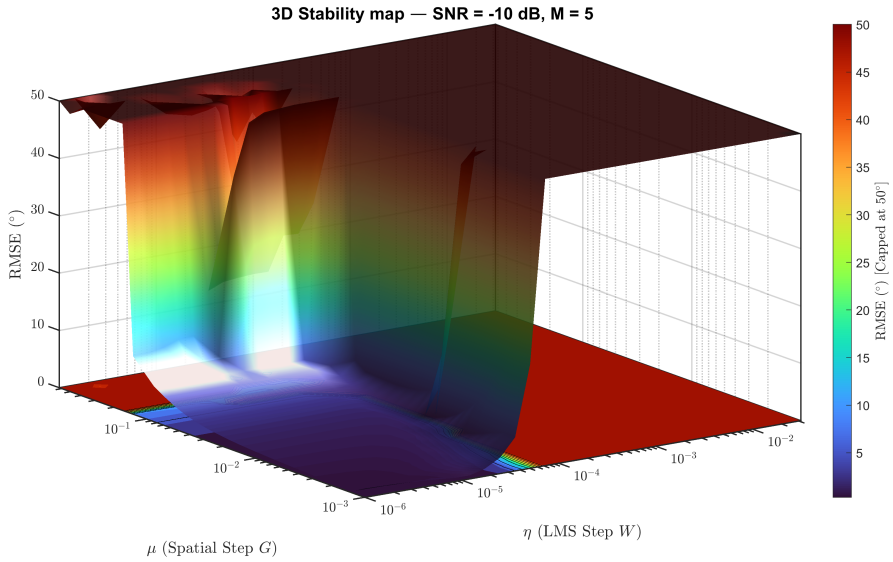
Figure 4.20: Steady-state RMSE as a function of the LMS step size η for various SNR levels ($\mu = 0.050$).

For $\eta \leq 10^{-5}$, the algorithm converges correctly and the tracking error remains stable for all tested SNR values. Increasing η beyond this limit causes the filter weights to fluctuate excessively, leading to divergence and an increase in RMSE.

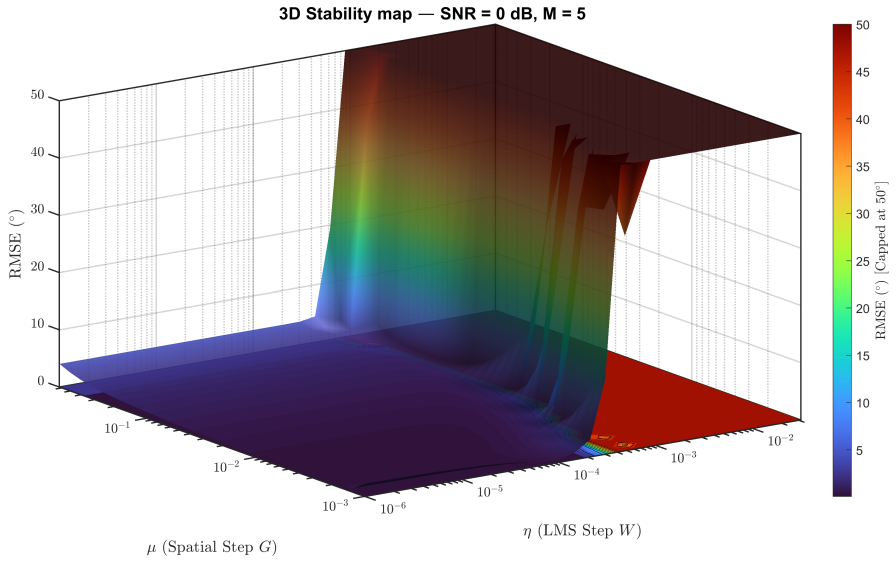
Higher SNR conditions tolerate slightly larger values of η, but the selected value $\eta = 10^{-5}$ provides stable operation for all considered scenarios. For this reason, it is retained as the default value throughout in this work.

4.2.4 Joint Parameter Optimisation: 3D Stability Landscape (μ, η)

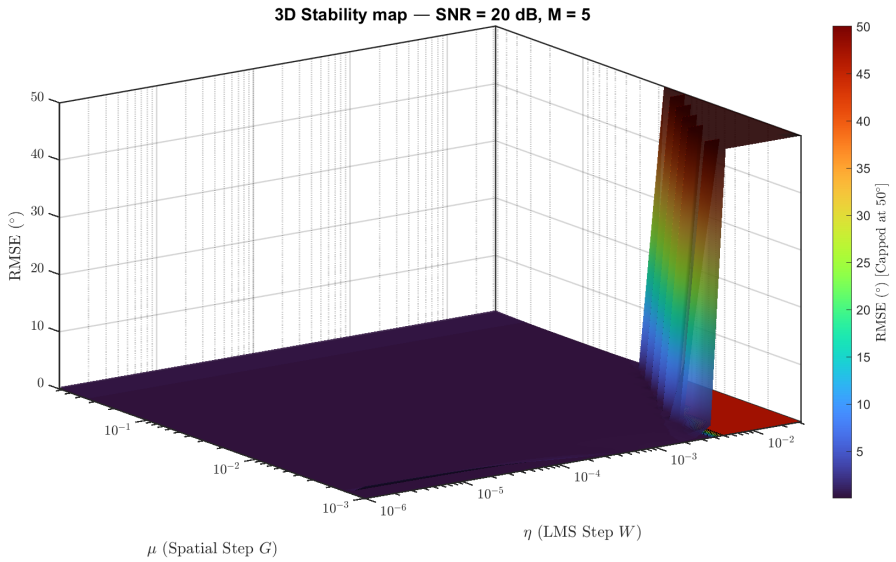
The individual effects of μ and η are not sufficient to characterize the tracker behavior. A joint parameter sweep is therefore performed to identify the stable operating region.



(a) SNR $= -10\,\mathrm{dB}$



(b) SNR = 0 dB



(c) SNR = 20 dB

Figure 4.20: 3D stability maps showing the steady-state RMSE as a function of μ and η under different SNR conditions.

While higher SNR improves overall system precision, large values of η consistently induce instability and elevated RMSE. As the SNR increases, this highly unstable region significantly narrows, thereby expanding the effective operational stability range of the algorithm.

4.2.5 Robustness to jamming

A jammer is an intentional interferer that transmits on the same frequency as the source of interest, with the deliberate intent of corrupting the receiver's spatial estimate. Testing against jamming is particularly relevant here because the GSC structure places nulls in the

directions of interferers through its auxiliary branch: evaluating this null-steering capability under controlled conditions directly validates the robustness claim of the algorithm.

In the simulations, the jammer transmits a sinusoidal signal at the same frequency as the target but from a different angular position. The target follows a sinusoidal trajectory centered around 0°, while the jammer is activated at snapshot $n = 3000$ from a fixed angle of 45°. At snapshot $n = 5000$, the jammer abruptly shifts to $-40°$, simulating a sudden change of interference direction. Two scenarios are considered: in the first, the jammer remains static at $-40°$ after the jump; in the second, it drifts continuously from $-40°$ onward, simulating a mobile interferer. The jammer is finally switched off at snapshot $n = 8000$, allowing observation of the tracker's recovery behaviour.

At moderate Jammer-to-Noise Ratio of 10 dB with a small step size $\mu = 0.005$, the GSC auxiliary branch successfully steers a null toward the jammer and the tracker maintains lock on the true target throughout all phases. At a higher Jammer-to-Noise Ratio of 12 dB with $\mu = 0.05$, the larger step size makes the algorithm overly reactive: the introduction of the jammer is sufficient to pull the estimate away from the true target direction, confirming that an aggressive adaptation gain compromises interference rejection in favour of tracking speed.

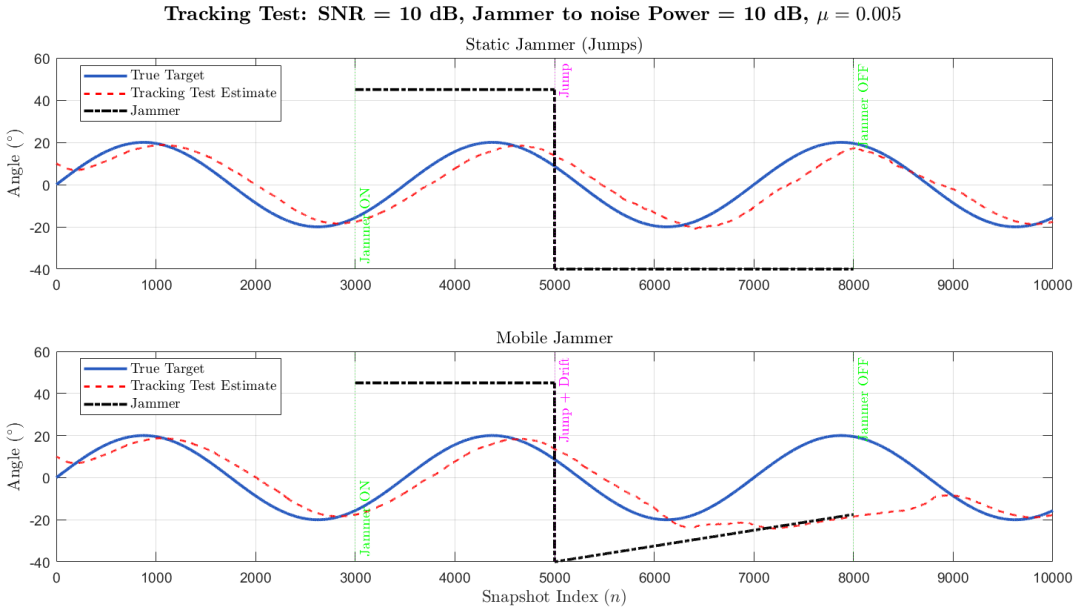


Figure 4.21: Static & mobile jammer with abrupt jumps (SNR = 10 dB, JNR = 10 dB)

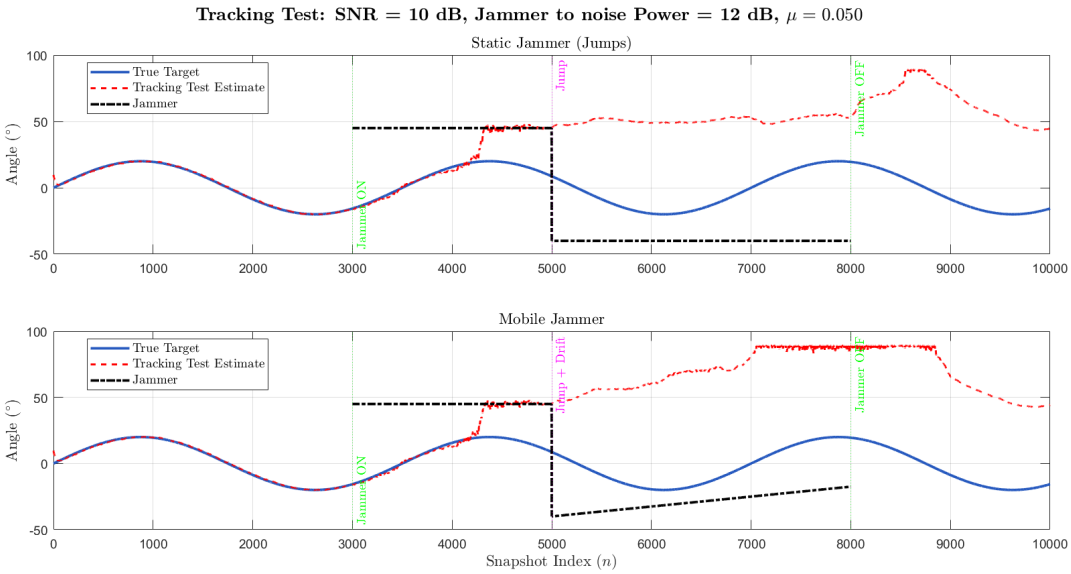
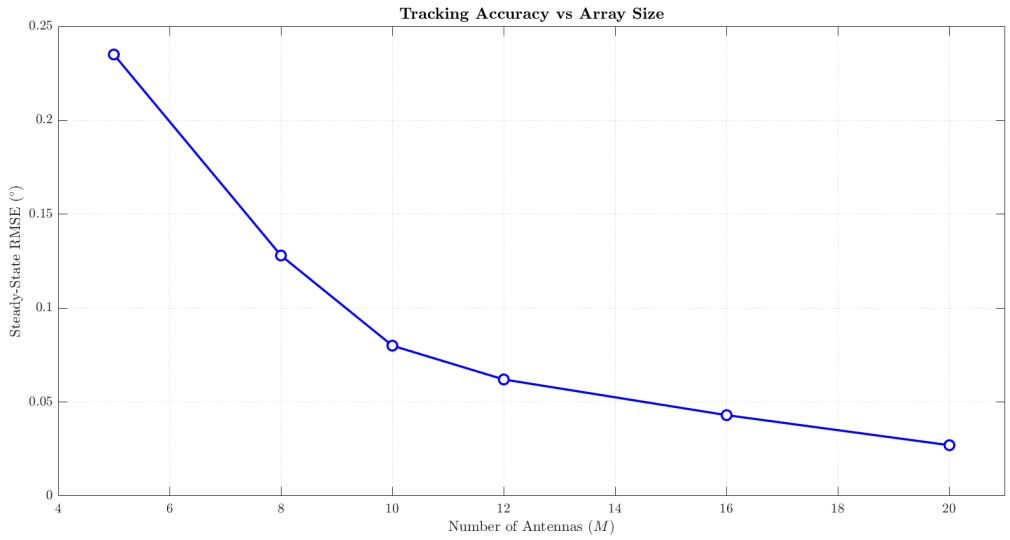


Figure 4.22: Static & mobile jamme with angular drift (SNR = 10 dB, JNR = 12 dB)

4.2.6 Influence of Array Size

Increasing the number of antennas M enhances spatial resolution, thereby reducing the steady-state RMSE. However, this precision is constrained by a narrower beamwidth, which strictly limits the permissible initial angular error for successful convergence. Outside this defined locking interval, the algorithm inevitably diverges.

The empirical trade-off between tracking accuracy and locking robustness is illustrated in the figures below.



Steady state RMSE vs Array size

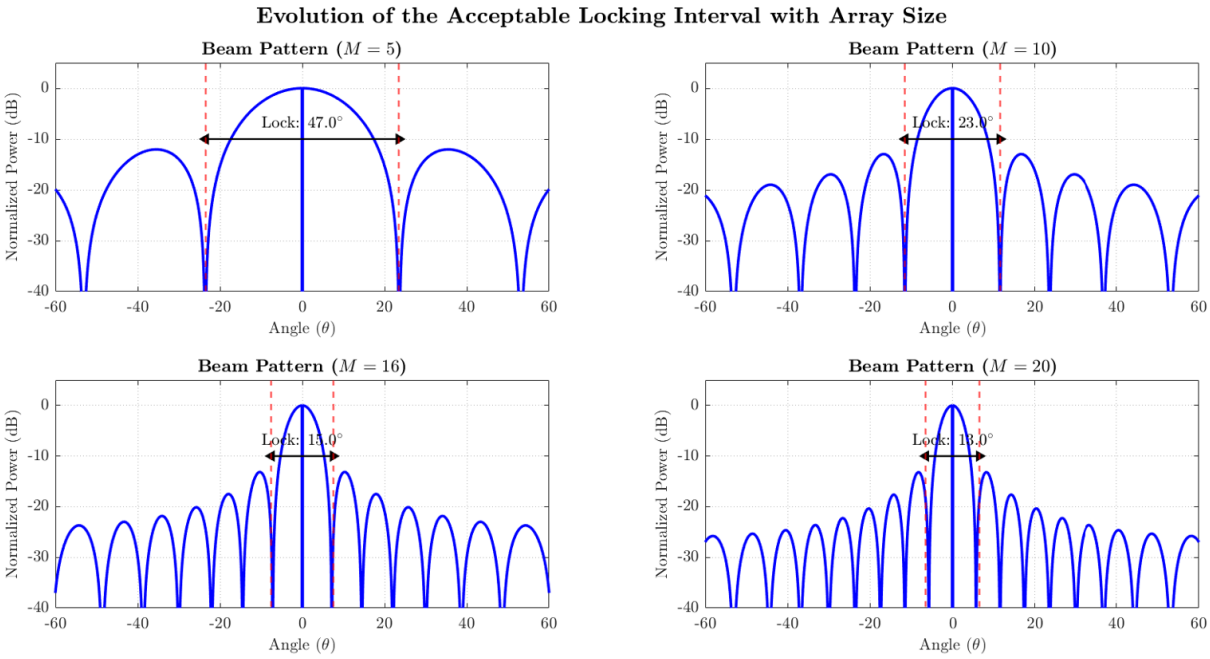


Figure 4.23: Evolution of the acceptable locking interval as a function of the number of array elements.

The ULA results establish the 1D tracking performance. The algorithm is now evaluated in its 2D extension using a Uniform Circular Array geometry.

4.3 RF Source Tracking Simulations for UCA

As described in Chapter 3, the algorithm has been extended to UCA geometry; simulation results confirm its validity.

To validate the 2D tracking capability, a trajectory was simulated under a Signal-to-Noise Ratio (SNR) of 10 dB. The algorithm demonstrates rapid and stable convergence toward the moving target.

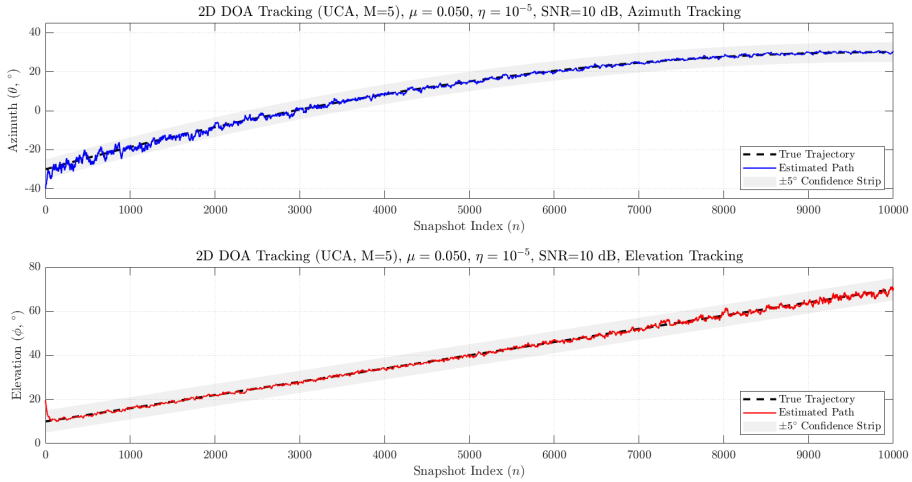


Figure 4.24: 2D Azimuth and Elevation tracking ($M = 5$, SNR = 10 dB).

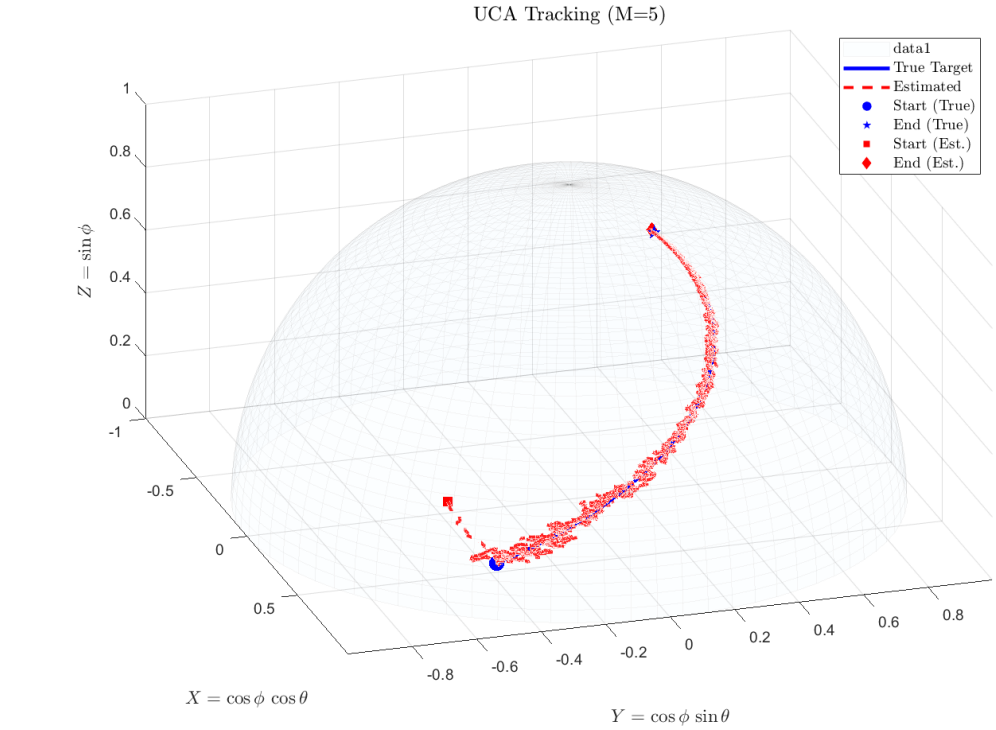


Figure 4.25: 3D trajectory tracking on the unit sphere ($M = 5$, SNR $= 10$ dB).

As illustrated in Figure 4.24 and Figure 4.25, the estimated path tracks the true trajectory accurately. The spatial beam locks onto the source and maintains tracking over time.

4.3.1 Impact of SNR degradation

To assess noise tolerance, the steady-state tracking error was evaluated against varying SNR levels for a fixed array size of $M = 5$, results are presented in Figure 4.26.

At SNR $= -5dB$, the steady-state RMSE remains below 6° for both azimuth and elevation

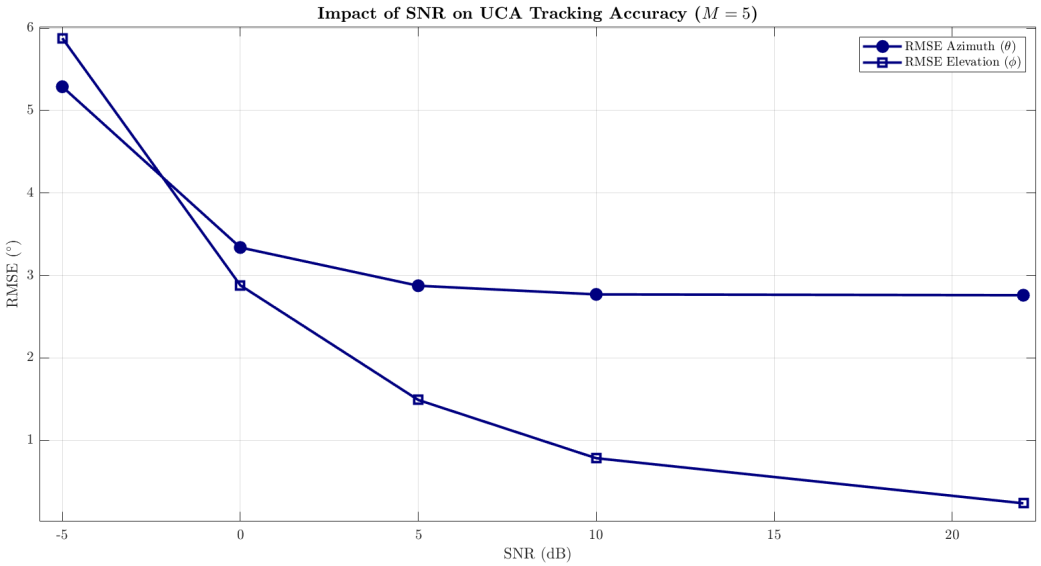


Figure 4.26: UCA 2D tracking RMSE θ, ϕ vs SNR

4.3.2 Influence of the Number of Antennas

The effect of the number of antennas M is evaluated on the UCA configuration. Increasing M reduces the estimation error, especially for the elevation angle and narrows the beamwidth in both direction, therefore reducing the tolerable initialisation error.

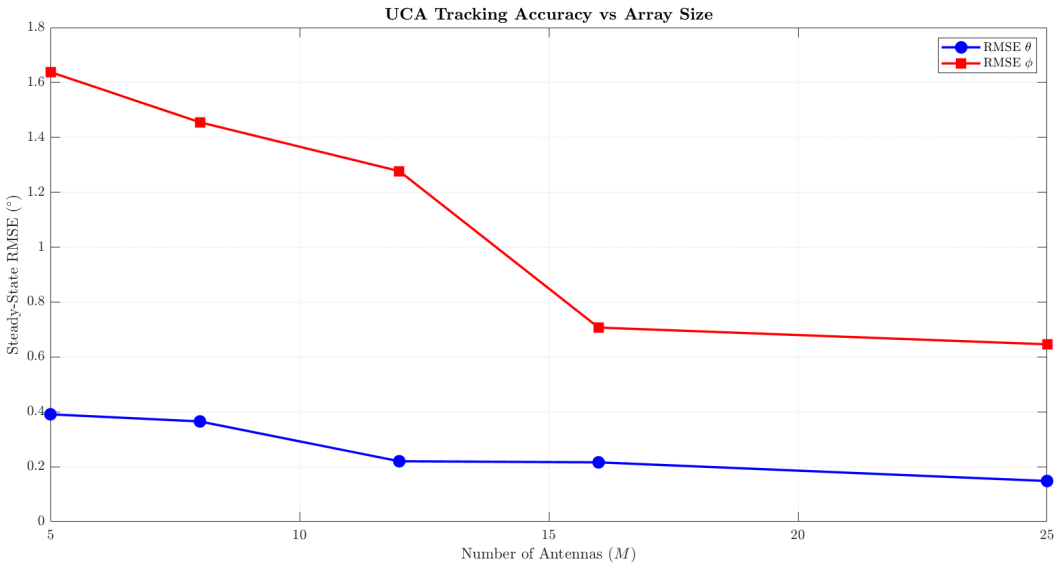


Figure 4.27: Effect of increasing the number of antenna in uniform circular arrays.

4.4 Conclusion

This chapter validates both subsystems through simulation. The SCMAP-CFAR detector maintains a stable false alarm rate across the tested SNR range, and its three governing parameters threshold factor K, memory coefficient λ, and reference window size Nref, have been individually characterized and jointly optimized. The adaptive beamformer converges reliably within the identified stable region $(\mu^*, \eta^*) = (0.05, 10^{-5})$, suppresses static and slow mobile jammers, and extends naturally to 2D tracking on a UCA geometry, with elevation accuracy as the primary remaining limitation. With the parameter space fully mapped and the expected performance bounds established, the next chapter moves to hardware deployment. The validated configurations are transferred directly into the GNU Radio implementation, where the system is evaluated against real RF signals captured by the KrakenSDR platform.

Chapter V

SDR Implementation and Experimental Results

This chapter presents the hardware deployment and experimental validation of the two subsystems studied in the previous chapters. An initial offline phase on captured IQ[1] data was used to prototype and validate the algorithms, before migrating to a fully real-time GNU Radio execution. All flowgraphs and programming paradigms involved are detailed in Appendix 3.3.

Each subsystem is validated independently, following the same modular evaluation strategy adopted in Chapter 4.

5.1 Hardware Platform

5.1.1 Transmitting *TX* Setup

All transmission tests are performed with a HackRF One software-defined radio connected to a VHF/UHF monopole antenna. The GNU Radio flowgraph controls three parameters: the carrier frequency f_c, the baseband signal frequency f_{bb}, and the RF gain. This same setup is used for all tests of both subsystems, providing a controlled and reproducible RF environment throughout the experiments.

5.1.2 Receiver *RX* Setup

The receiving platform is built around the KrakenSDR, a five-channel coherent software-defined radio. It consists of five RTL-SDR tuners that all share the same clock reference. Unlike the HackRF One, the KrakenSDR is a receive-only device, making it well suited for direction-finding and array signal processing applications such as passive radar. A key feature of the KrakenSDR is its built-in switchable noise source, which is briefly activated at startup to

[1] IQ (In-phase/Quadrature) modulation represents a complex-valued signal using two orthogonal components: an in-phase (I) component and a quadrature (Q) component shifted by 90°. Modulation combines these components onto a carrier, while demodulation recovers the complex baseband signal, preserving both amplitude and phase information.

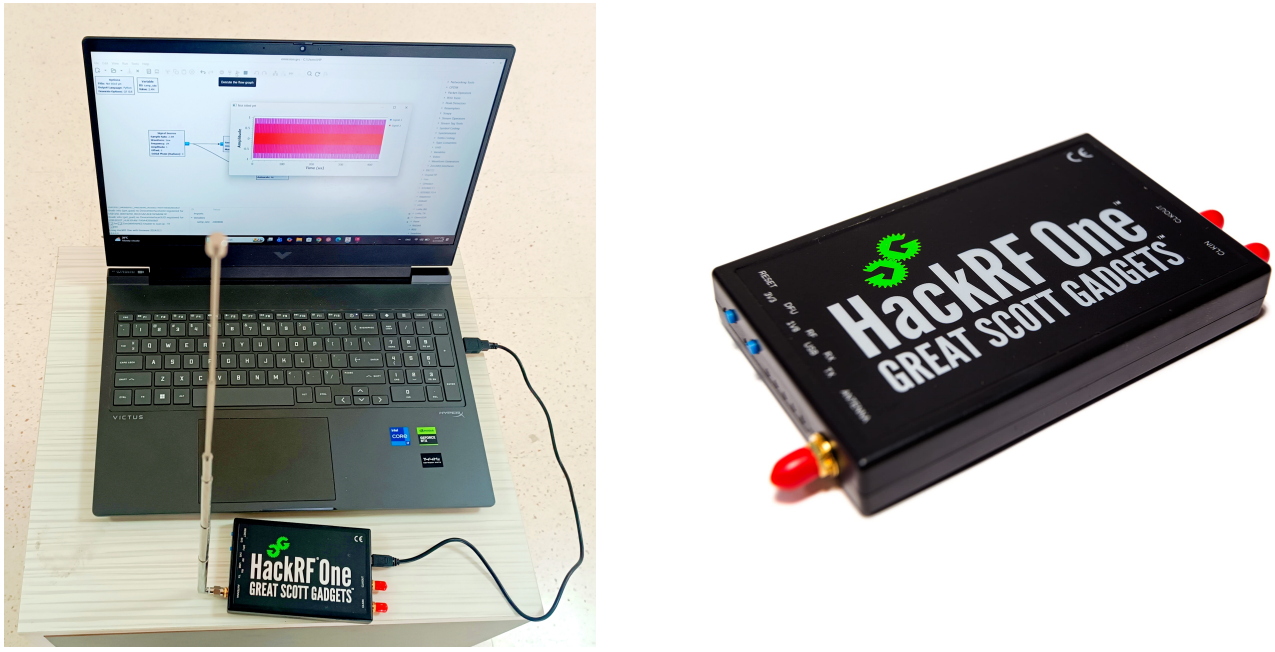


Figure 5.1: HackRF One transmission setup.

automatically calibrate the phase and gain across all five channels. [2], ensuring that all channels remain coherent at the sample level, which is a strict requirement for array signal processing.

5.2 IQ Acquisition

The DAQ firmware named Heimdall (Explainded in appendice 3.3) preprocessesthe five RTL-SDR[3] streams and exposes phase-coherent IQ Ssamples on `127.0.0.1:5000`. Frequency retuning commands are issued over the control port `127.0.0.1:5001`. Two acquisition modes are supported:

- **Real-time mode.** IQ frames flow directly into the GNU Radio OOT processing blocks, outputs are displayed on the Qt radar interface with no intermediate storage.
- **Offline mode.** A GNU Radio *File Sink* (preceded by a *Streams-to-Stream* multiplexer) serialises the five channels as `float32` interleaved pairs into a single `.dat` file for later processing.

[2]The noise source injects a common reference signal into all receivers; the measured phase offsets between channels are then corrected in software before any DoA processing begins.

[3]RTL-SDR (Realtek Software Defined Radio) is a low-cost software-defined radio receiver.



Figure 5.2: KrakenSDR with noise source activated during phase-calibration startup.

Table 5.1: IQ acquisition parameters.

Parameter	**Value**
Sample rate (hardware)	$F_s = 2.4\,\text{MHz}$
Sample rate (decimated)	$F_{s,\text{dec}} = 24\,\text{kHz}$
Decimation factor	$\div 100$ or $\div 10$
Channels	5 (coherent, float32 IQ)
Data format (offline)	Interleaved float32, `.dat`
Control interface	TCP `127.0.0.1:5001`
Data interface	Shared memory / TCP `127.0.0.1:5000`

5.3 DoA Signal Preprocessing

Before the adaptive tracking algorithm is run, three preprocessing steps are applied to the raw IQ frames: phase synchronisation enforced by the firmware, baseband FIR filtering, and a MUSIC initialisation that seeds the GSC-LMS tracker with a reliable starting angle. These steps are summarised in Figure 5.4.

5.3.1 Phase Synchronisation and Decimation

The Heimdall DAQ firmware enforces sample-level phase coherence across the five channels via a shared clock distributed on the KrakenSDR PCB, phase calibration is applied at startup and after each frequency retune. For real-time operation, the flowgraph requests frames at 24 kHz, Software processing provides them by transparent digital decimation ($\div 100$) without modifying the hardware oscillator.

5.3.2 Baseband FIR Filtering

After IQ down-conversion, a complex bandpass FIR filter isolates the signal of interest within the 2.4 MHz instantaneous bandwidth. The filter is built as a low-pass Hamming-windowed prototype (order 100, 3 dB cutoff at 1 kHz) frequency-shifted to the desired baseband offset f_c of the detected tone:

$$h_{\mathrm{BP}}[n] = h_{\mathrm{LP}}[n]\, e^{j2\pi f_c n/F_s}, \qquad n = 0, \ldots, 100. \tag{V.1}$$

The centre frequency f_c is updated at each frequency hop; all other parameters remain fixed. The equivalent GNU Radio block is the *Xlating FIR Filter*, which performs the frequency shift and low-pass filtering in a single polyphase stage.

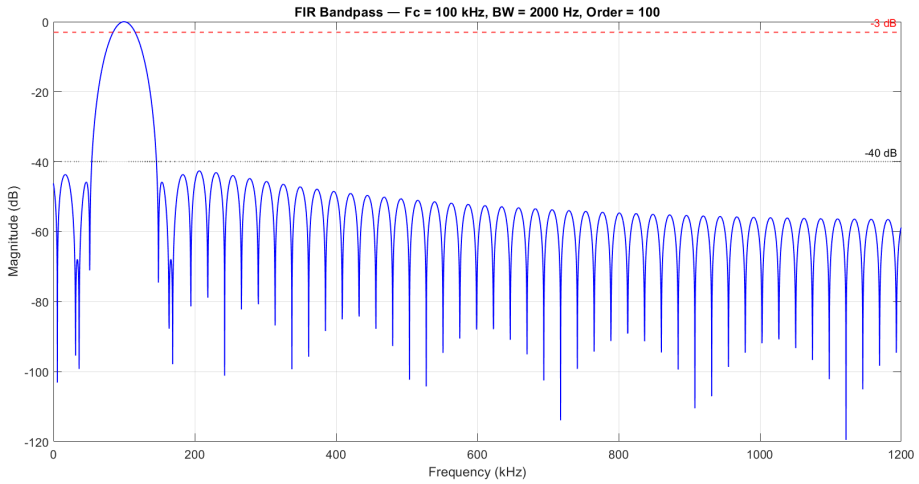


Figure 5.3: Frequency response of the complex bandpass FIR filter.

5.3.3 MUSIC Initialisation, SNR and Source-Number Estimation

Before each tracking we run a short MUSIC acquisition over 1024 snapshots provides the initial DoA estimate $\hat{\theta}_0$ that seeds the upstream blocks, guaranteeing that the starting point lies within the locking interval established in Chapter 4. At $F_s = 2.4$ MHz, 1024 snapshots represent less than 0.5 ms, the Eigen-based C++ MEX module solves the eigendecomposition in under 1 ms, making this step fully compatible with real-time operation.

SNR and source-number estimation: The sample covariance matrix $\hat{R}_{xx} = \frac{1}{N}\mathbf{X}\mathbf{X}^{\mathsf{H}}$ is formed from the N snapshots and eigendecomposed. The noise floor is estimated as the mean of the $M - \hat{\nu}$ smallest eigenvalues:

$$\hat{\sigma}_n^2 = \frac{1}{M - \hat{\nu}} \sum_{i=\hat{\nu}+1}^{M} \lambda_i, \tag{V.2}$$

and the estimated SNR is:

$$\widehat{\mathrm{SNR}}_{\mathrm{dB}} = 10 \log_{10}\!\left(\frac{\lambda_1 - \hat{\sigma}_n^2}{\hat{\sigma}_n^2} \right). \tag{V.3}$$

The number of active sources $\hat{\nu}$ is the count of eigenvalues exceeding the threshold $\gamma = 3\,\hat{\sigma}_n^2$.

MUSIC DoA initialisation: The noise subspace $\mathbf{E}_n$ (eigenvectors associated with the $M-\hat{\nu}$ smallest eigenvalues) is used to form:

$$P_{\text{MUSIC}}(\theta) = \frac{1}{\|\mathbf{E}_n^{\text{H}}\mathbf{a}(\theta)\|^2 + \varepsilon}, \quad \mathbf{a}(\theta) = \left[e^{jkd\,m\sin\theta}\right]_{m=0}^{M-1}. \tag{V.4}$$

The initial angle $\hat{\theta}_0$ is the global peak of P_{MUSIC} over $\theta \in [-90°, +90°]$.

The full preprocessing pipeline is illustrated in Figure 5.4, which shows the complete chain from raw array input to the adaptive tracker input.

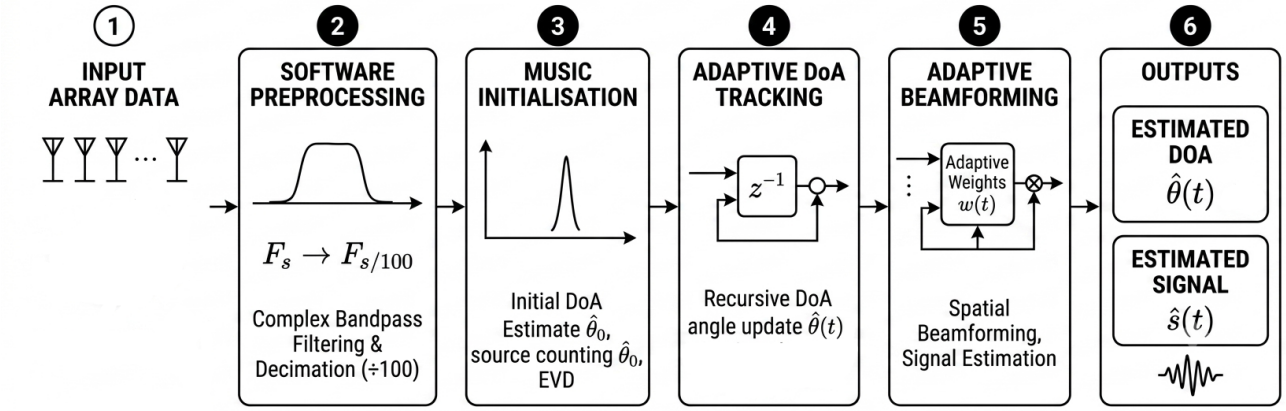


Figure 5.4: Proposed DoA tracking pipeline

5.4 Test Environment

All experiments are conducted indoors in a laboratory room. The transmitter is placed at measured positions relative to the broadside direction of the ULA. Three static positions are tested: broadside centre (0°), left (−25°), and right (+22°). Angles are measured geometrically and cross-validated by the MUSIC initialisation scan.

5.4.1 Testbench configuration

Antenna placement is critical for system performance. The inter-element spacing is set to $d = \lambda/2$ to avoid spatial aliasing. The KrakenSDR ships with five telescopic monopoles, which are vertically polarized and omnidirectional in the azimuthal plane. The same type of monopole was used for the emitter. The adjustable monopoles were set to maximum extension for 433 MHz tests and minimum extension for 1 GHz tests, no significant sensitivity difference was observed between the two length settings.
In practice, a minimum distance between the transmitter and the receiver is required to ensure operation in the far-field region and to satisfy the plane-wave approximation. In our experiments, a separation distance of 7 m was used to ensure that this condition is satisfied.

Table 5.2: Array geometry per frequency band.

Parameter	**433 MHz**	**1 GHz**
Inter-element spacing d	33 cm ($\lambda/2$)	15 cm ($\lambda/2$)
element length	max extension	min extension

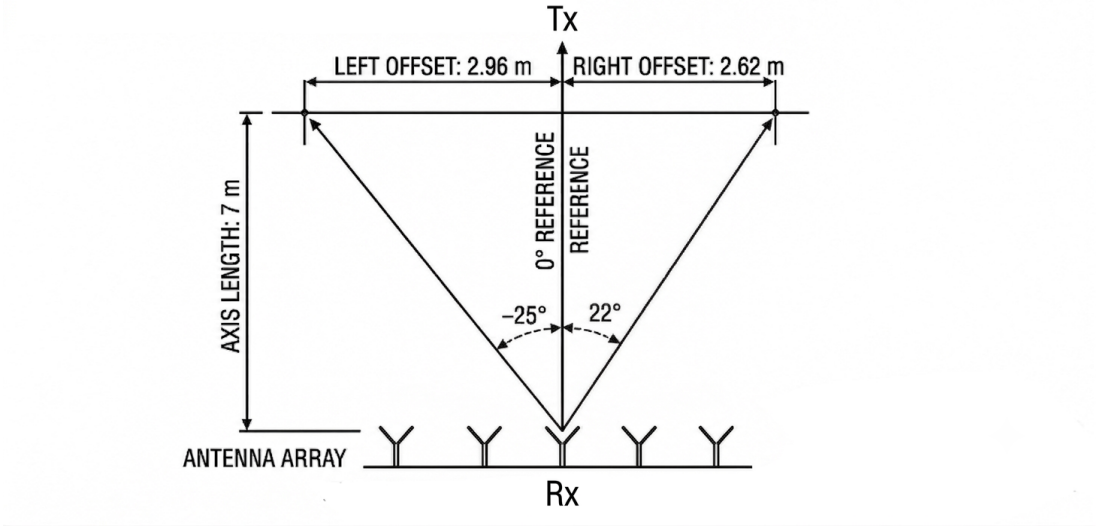


Figure 5.5: representative top view of the test bench



(a) Left position ($\theta = -25°$). (b) Right position ($\theta = +22°$).

Figure 5.6: Transmitter positions at 433 MHz $d = 33$ cm.

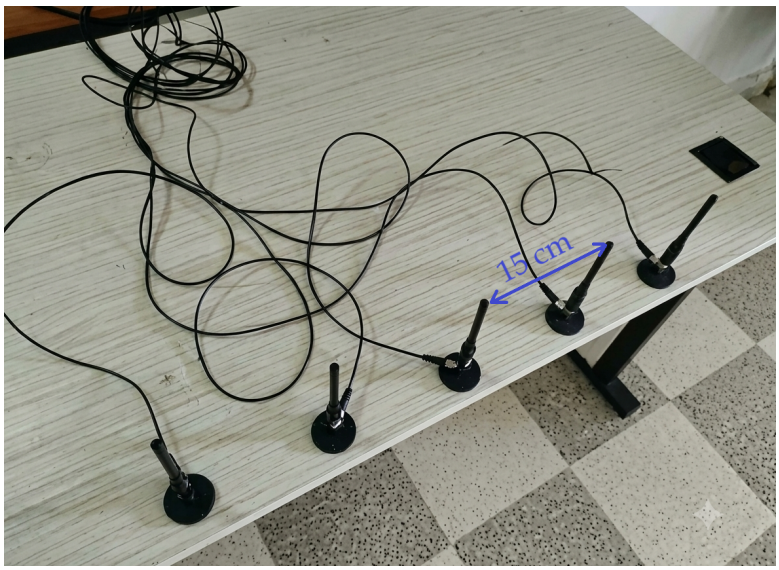


Figure 5.7: Array reconfigured for 1 GHz tests, $d = 15$ cm.

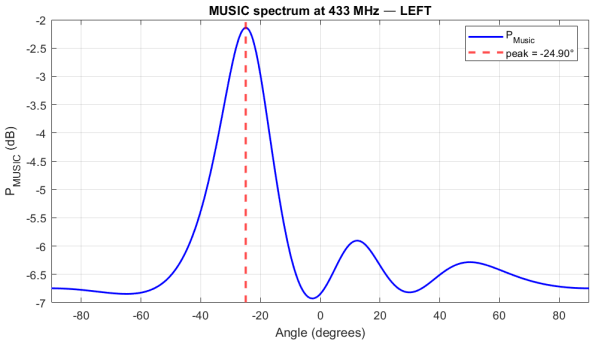


Figure 5.8: UCA configured for 433 MHz , diameter = 33 cm, angular separation 72°

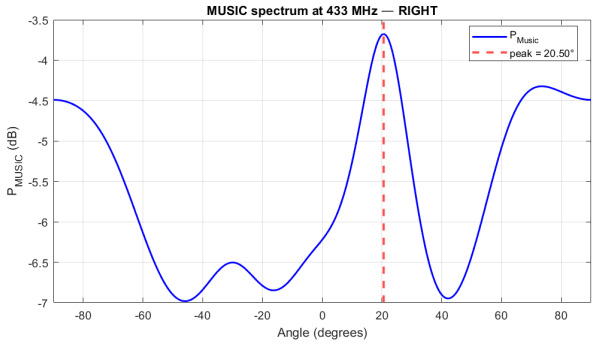
5.5 DoA Tracking Results

5.5.1 MUSIC Localisation for landmark setting

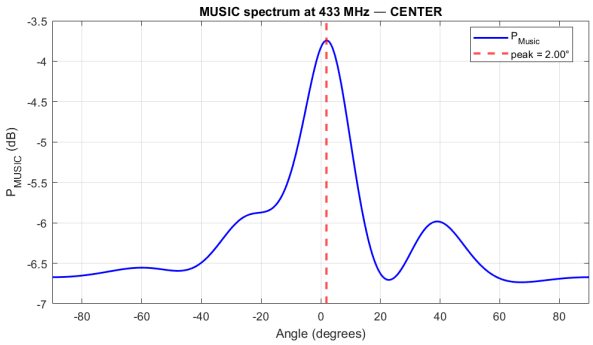
Figures 5.9 and 5.10 show the localisation results obtained at 433 MHz and 1 GHz for the left and right and the center. In all cases the dominant peak is well resolved and within 2° of the ground truth.



(a) Left $\hat{\theta}_{est} = -25°$ (true: $-25°$).

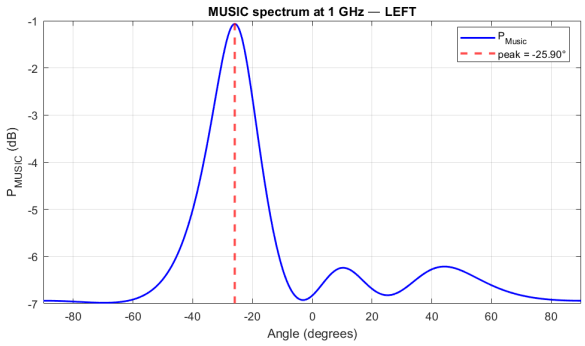


(b) Right $\hat{\theta}_{est} = 20.5°$ (true: $+22°$).

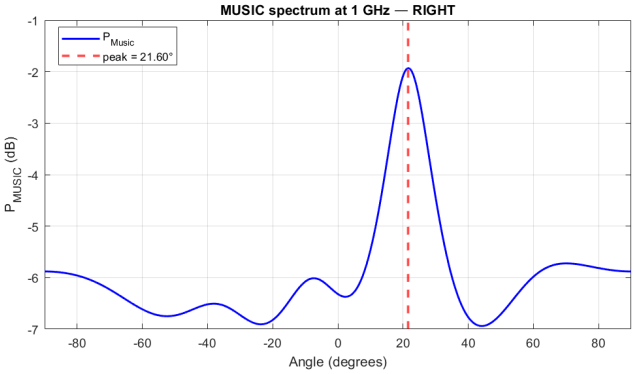


(c) Center $\hat{\theta}_{est} = 2°$ (true: $0°$).

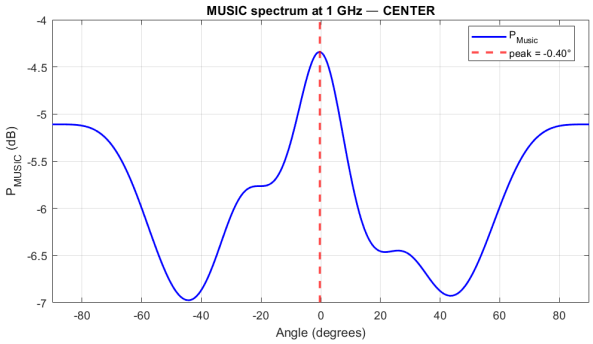
Figure 5.9: MUSIC Localisation at 433 MHz



(a) Left $\hat{\theta}_{est} = $ -25.90°.



(b) Right $\hat{\theta}_{est} = 21.60°$.



(c) Center $\hat{\theta}_{est} = 0.40°$.

Figure 5.10: MUSIC Localisation at 1 GHz.

Table 5.3: Initial DoA localisation results.

Freq.	**Position**	**True θ**	$\hat{\theta}_{est}$	**Error**
433 MHz	Centre	0°	2.0°	2.0°
433 MHz	Left	−25°	−24.9°	0.1°
433 MHz	Right	+22°	+20.5°	1.5°
1 GHz	Centre	0°	+0.4°	0.4°
1 GHz	Left	−25°	−25.9°	0.9°
1 GHz	Right	+22°	+21.6°	0.4°

All estimates fall within 2° of the ground truth across both frequency bands, confirming that MUSIC provides a reliable initialisation point for the adaptive GSC-LMS tracker, systematically placing the starting angle within the locking interval identified in Chapter 3 and 4.

5.5.2 Tracking Test for 433 MHz

Scenario: The emitter is initially located at −25° and transmits a 433MHz signal with a 500kHz complex baseband bandwidth, corresponding to an RF signal centered at 433.5MHz. The signal lies within the ±1.2MHz acquisition band. During the experiment, the transmitting platform (laptop and HackRF mounted on a rolling table) moves at constant speed from −25°(Point B) to approximately 22°(Point A).

The recorded dataset is processed offline with $\mu = 5 \cdot 10^{-4}$, $\eta = 10^{-6}$, and a decimation(/10).

Inter-Channel Cross-Correlation

As explained in 5.3, small time differences between antennas are tolerated since the Heimdall firmware synchronizes the samples before processing. To verify this synchronization, the cross-correlation between the five channels was computed, leading to the results shown in Figure 5.11 and Table 5.4. are estimated from the normalized cross-correlation between each pair of channels:

$$\hat{\tau}_{ij} = \arg\max_{\tau} \left| \sum_{n} x_i(n)\, x_j^*(n-\tau) \right|$$

The lag $\hat{\tau}_{ij}$ at which the cross-correlation magnitude peaks gives the relative time delay between channels i and j in samples. This value is then converted to nanoseconds using $\hat{\tau}_{ij}[\text{ns}] = \hat{\tau}_{ij}[\text{samples}]/(F_s \cdot 10^9)$. The estimation is performed on a 98 304-sample window (≈ 40 ms).

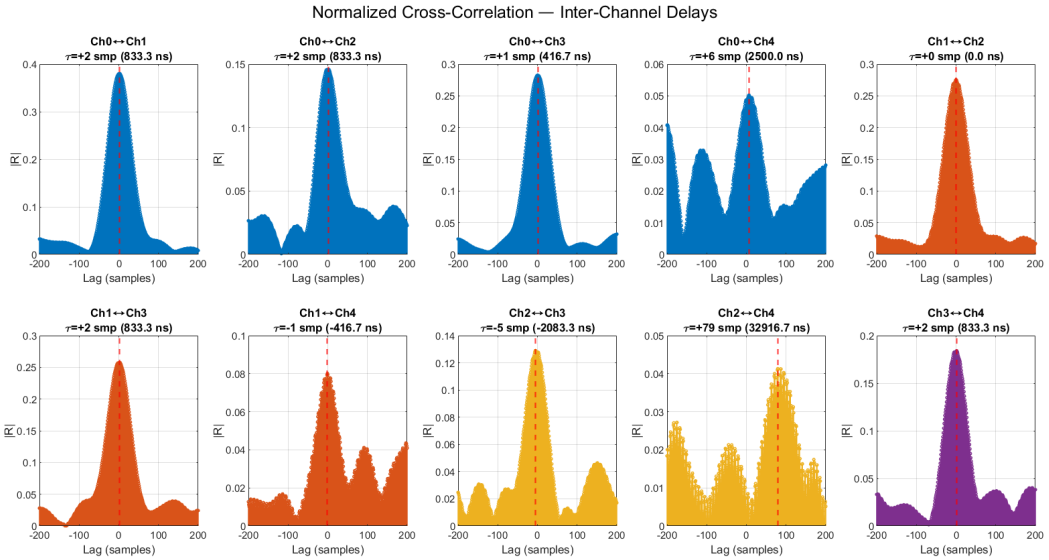


Figure 5.11: Normalized cross-correlation between all channel pairs of the 5-element ULA at 433 MHz.

Table 5.4: Estimated inter-channel delay matrix (nanoseconds).

	Ch0	**Ch1**	**Ch2**	**Ch3**	**Ch4**
Ch0	0.0	+833.3	+833.3	+416.7	+2500.0
Ch1	−833.3	0.0	0.0	+833.3	−416.7
Ch2	−833.3	0.0	0.0	−2083.3	+32916.7
Ch3	−416.7	−833.3	+2083.3	0.0	+833.3
Ch4	−2500.0	+416.7	−32916.7	−833.3	0.0

The results of table 5.4 show good synchronization across all channels, with delays generally below $\pm 1, \mu$s. Only the Ch2↔Ch4 pair exhibits a larger delay of about 33μs. This may be caused by multipath reflections or a software bottleneck.[4]

This small mismatch has no noticeable impact on the beamforming performance.

Spectrum Analysis

Figure 5.12 shows the PSD of the received signal before and after filtering, with the useful component around +500 kHz and an estimated SNR of 32.4 dB.

After FIR filtering, the signal is better isolated and out-of-band noise is strongly reduced, increasing the SNR to 50.5 dB. Spurious spectral components due to multipath and SDR imperfections are also significantly attenuated.

[4]The KrakenSDR uses five RTL-SDR receivers connected through a shared USB 2.0 hub. Under heavy processing load, some channels may experience buffering or scheduling delays, especially when operating close to the USB bandwidth limit.

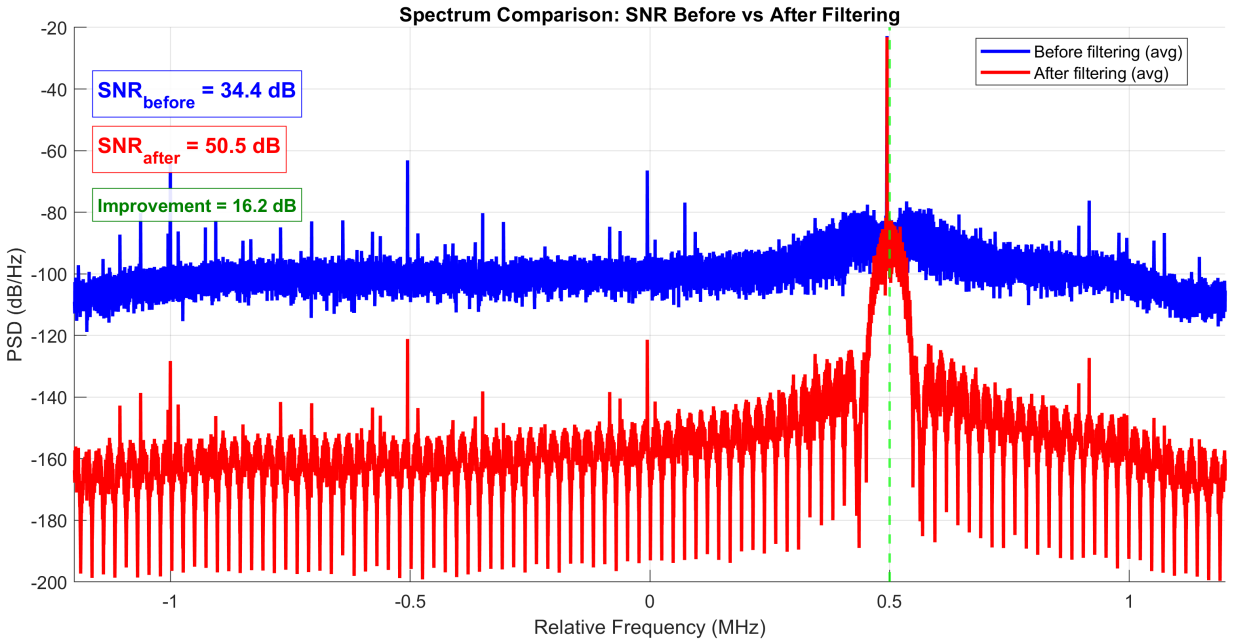


Figure 5.12: Comparison of the spectra before and after filtering.

DoA Tracking Results

Figure 5.13 shows the estimated DoA trajectory. The tracker follows the emitter continuously from $-25°$ to approximately $22°$, demonstrating stable convergence with several oscillations around the main path.

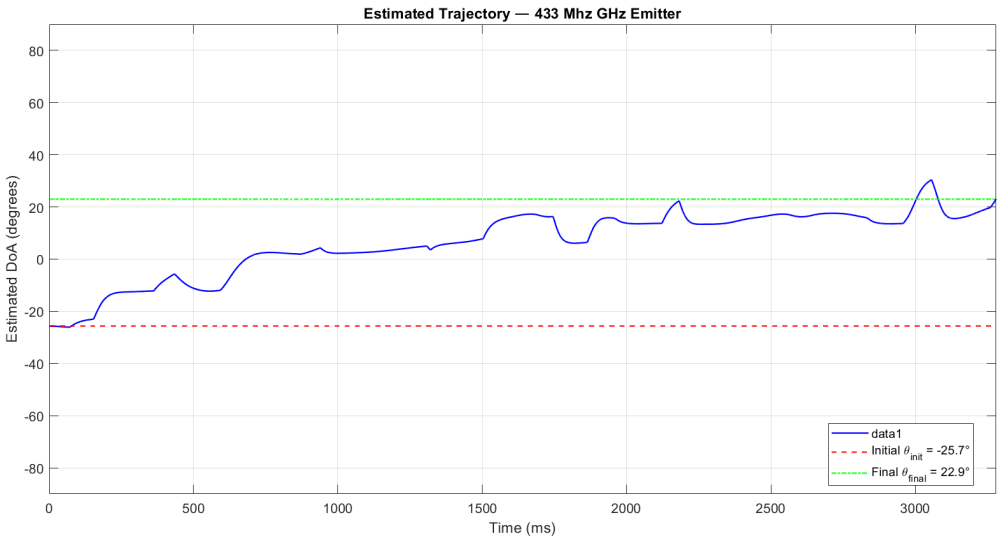


Figure 5.13: Estimated DoA trajectory during the 433 GHz experiment.

Beam Pattern Evolution

Figure 5.14 compares the initial beam steered toward the MUSIC initialization angle and the final beam obtained after convergence. The final main lobe points towards $22.9°$ giving an error of $0.9°$.

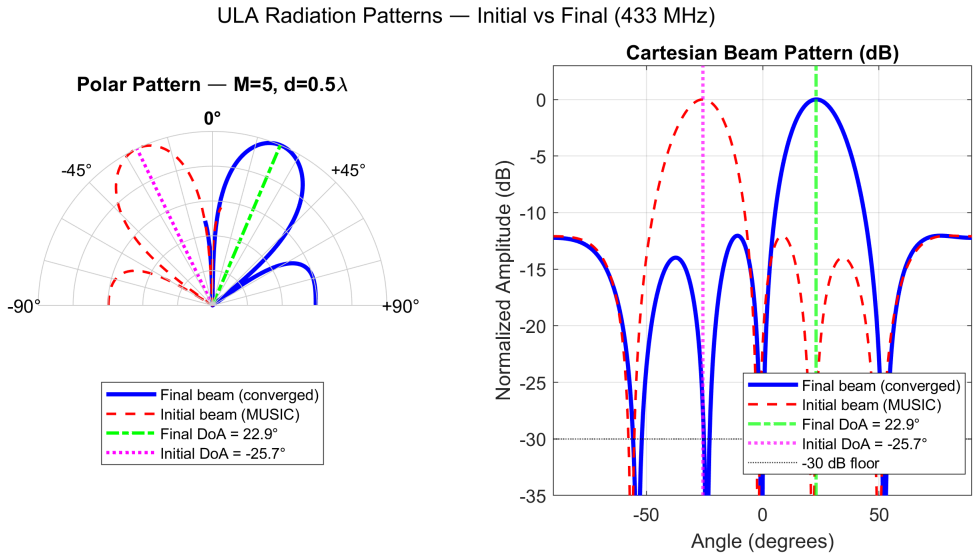


Figure 5.14: Initial and final beam patterns at 433 Ghz

5.5.3 Tracking Test for 1 GHz

Scenario: The emitter is initially located at 22° and transmits a 1 GHz signal with a 200 kHz complex baseband signal, corresponding to an RF signal centered at 1.0002 GHz. During the experiment, the transmitting platform moves at constant linear speed along a half-azimuthal arc from 22° to approximately −25°, then returns to the initial position.

Inter-Channel Cross-Correlation

Similarly to the previous test, we compute the cross correlation between all the channel, to verify the time integrity of the saved data.

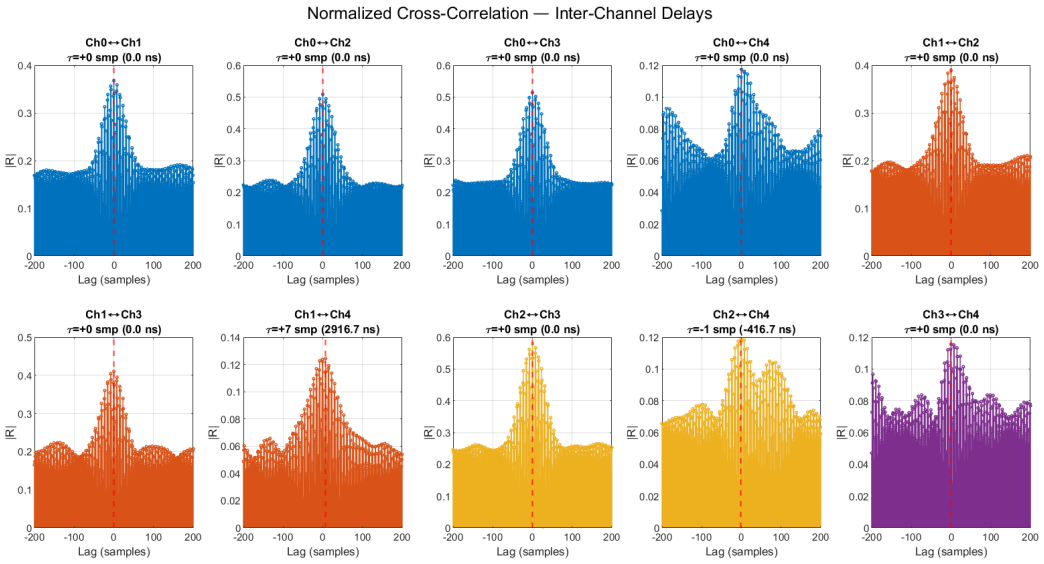


Figure 5.15: Normalized cross-correlation between all channel pairs of the 5-element ULA at 1 GHz.

Table 5.5: Estimated inter-channel delay matrix (nanoseconds) at 1 GHz.

	Ch0	**Ch1**	**Ch2**	**Ch3**	**Ch4**
Ch0	0.0	0.0	0.0	0.0	0.0
Ch1	0.0	0.0	0.0	0.0	+2916.7
Ch2	0.0	0.0	0.0	0.0	−416.7
Ch3	0.0	0.0	0.0	0.0	0.0
Ch4	0.0	−2916.7	+416.7	0.0	0.0

Spectrum Analysis

Figure 5.16 shows the PSD of the received signal before filtering, with the useful component around +500 kHz and an estimated SNR of 33.3 dB.

After FIR filtering, the signal is better isolated and out-of-band noise is strongly reduced, increasing the SNR to 45.9 dB.

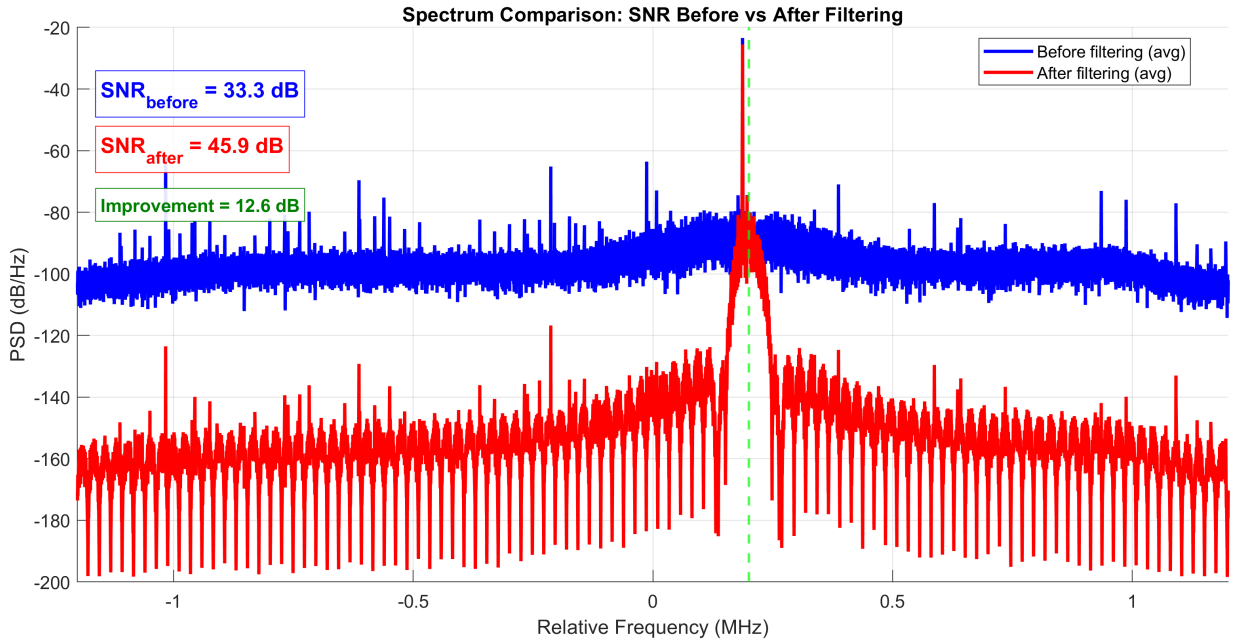


Figure 5.16: Comparison of the spectra before and after filtering at 1 GHz.

DoA Tracking Results

Figure 5.17 shows the estimated DoA trajectory.

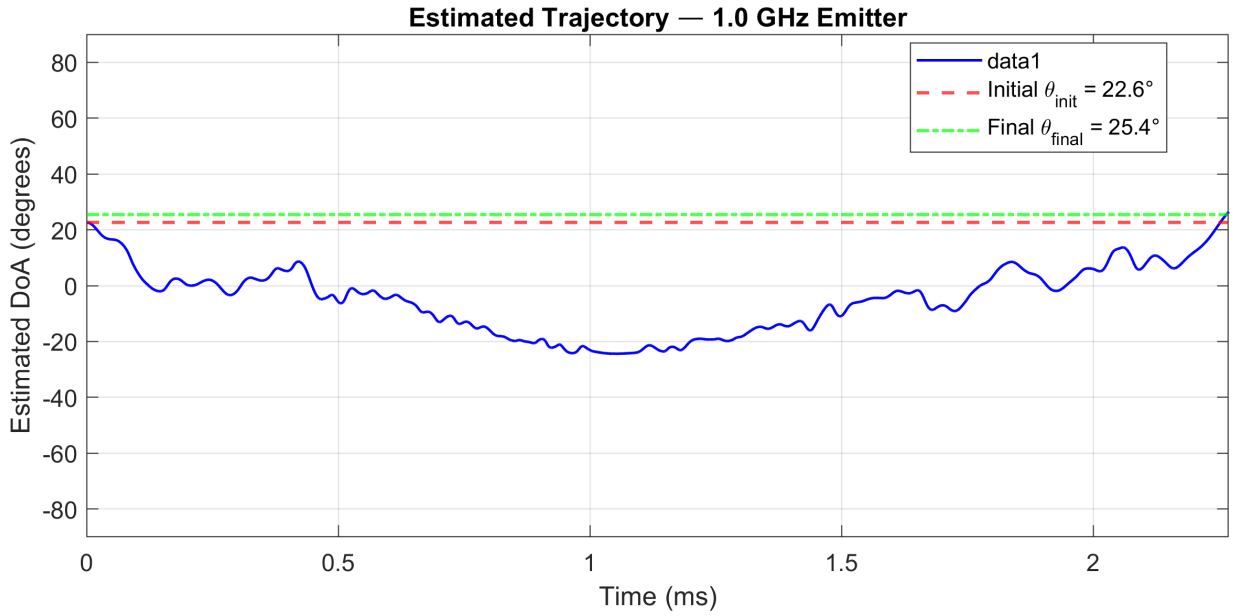


Figure 5.17: Estimated DoA trajectory during the 1 GHz experiment.

Figure 5.17 shows that the tracker follows the emitter along the half-azimuthal arc from 23° (point A) to −25° (point B), then tracks the return path. The algorithm correctly captures the overall trajectory , but noticeable oscillations appears, a consequence of increased multipath sensitivity at 1 GHz.

To mitigate these oscillations without modifying the core algorithm, a two-state Kalman filter is cascaded after the GSC-LMS output. The filter operates in the electrical angle domain $u = \sin\theta$, rather than directly on θ. This choice avoids introducing the nonlinearity of the arcsin function inside the feedback loop and ensures that the constant-speed model $\dot{u} \approx$ const remains physically consistent with the uniform linear displacement of the transmitting platform.

Kalman Post-Filter

The state vector $\mathbf{x}_k = [u_k,\, \dot{u}_k]^\top$ tracks $u = \sin\theta$ and its rate of change. Under a constant-speed assumption, the prediction step is:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}\,\hat{\mathbf{x}}_{k-1}, \qquad \mathbf{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \tag{V.5}$$

where $\Delta t = 1/F_s$. The measurement update fuses the raw GSC-LMS estimate u_{meas} with the predicted state:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(u_{\text{meas}} - \hat{x}^{(0)}_{k|k-1}\right), \qquad \mathbf{K}_k = \frac{\mathbf{P}_{k|k-1}\,\mathbf{h}^\top}{S_k}, \tag{V.6}$$

where $\mathbf{h} = [1,\, 0]^\top$, $S_k = \mathbf{h}^\top \mathbf{P}_{k|k-1}\mathbf{h} + R$, and R is the measurement noise variance. The estimated angle is finally recovered as:

$$\hat{\theta}_k = \arcsin\!\left(\hat{x}^{(0)}_k\right). \tag{V.7}$$

The process noise covariance $\mathbf{Q} = \text{diag}(Q_{00},\, Q_{11})$ is set to $Q_{00} = 10^{-3}$ to allow gradual DoA

changes, while $Q_{11} = 10^{-4}$ encodes the quasi-constant angular velocity of the rolling platform. The measurement noise $R = 4.5 \cdot 10^{-4}$ reflects confidence in the original algorithm output when it is consistent with the motion model. The parameters were calibrated empirically:

Table 5.6: Kalman filter parameters at 1 GHz tracking.

Parameter	**Value**
Q_{00}	10^{-3}
Q_{11}	10^{-4}
R	$4.5 \cdot 10^{-4}$
$P_{00} = P_{11}$	0.1

Figure 5.18 and 5.19 shows the raw GSC-LMS output against the Kalman-enhanced estimate . The filter substantially reduces transient oscillations while preserving the ability to follow the full movement.

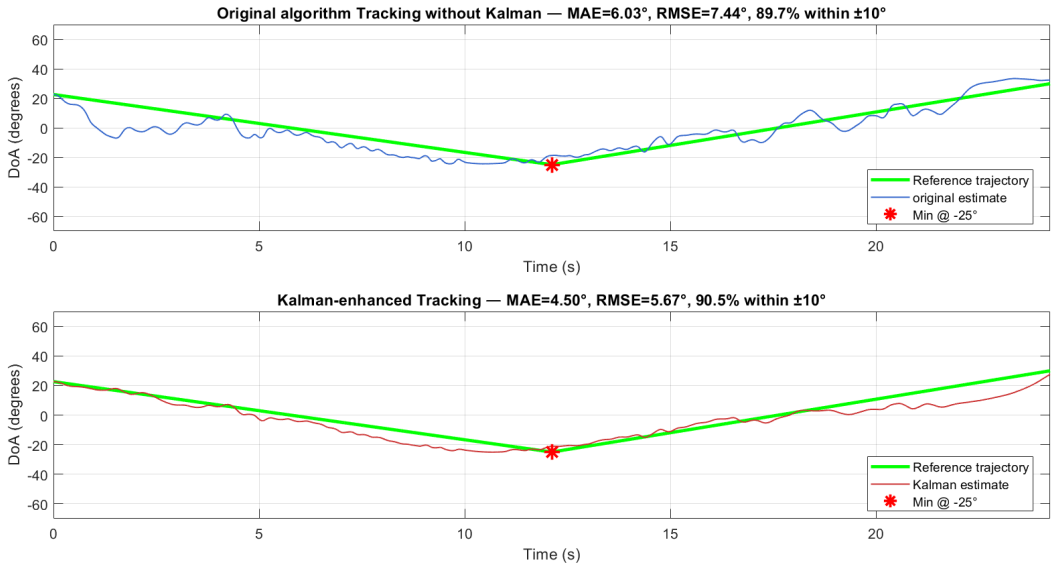


Figure 5.18: DoA tracking at 1 GHz comparisons between the original technique and the kalman enhanced version.

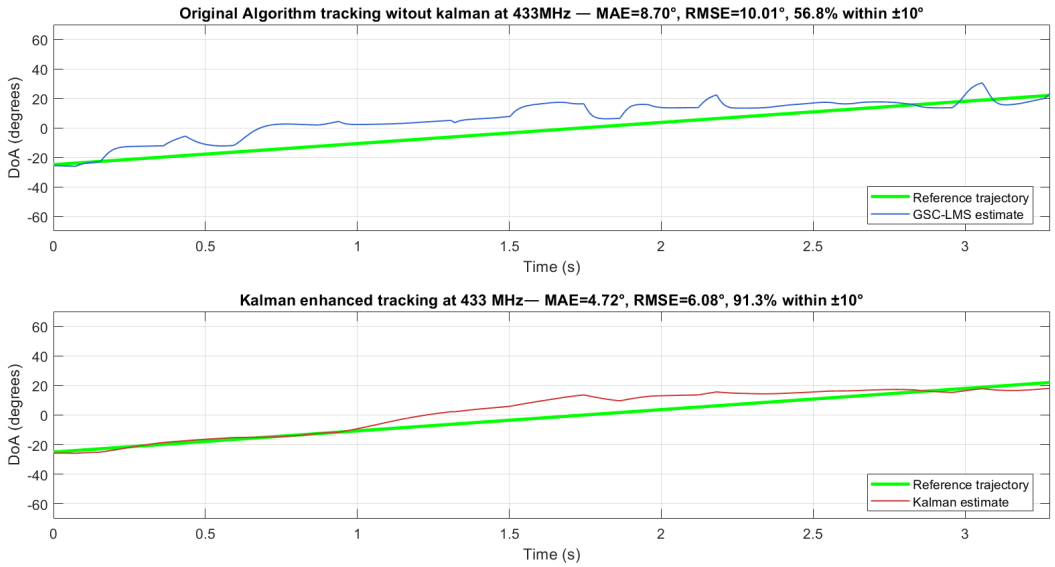


Figure 5.19: Tracking at 433 MHz with and without kalman post-filtering.

The two metrics reported in Table 5.7 are defined as follows. The MAE (Mean Absolute Error) is the average of the absolute deviation between the estimated DoA and the reference trajectory; it gives a direct measure of the tracker's angular bias in degrees. The Within ±10° criterion is the percentage of time instants for which the estimation error remains bounded within a ±10° window around the reference; it quantifies the temporal reliability of the tracker regardless of occasional large deviations.

Table 5.7: Tracking performance comparison: Original GSC-LMS vs Kalman-enhanced, at 433 MHz and 1 GHz.

	433 MHz		**1 GHz**	
Metric	**Original**	**Kalman**	**Original**	**Kalman**
MAE (°)	8.70	4.72	6.03	4.50
RMSE (°)	10.01	6.08	7.44	5.67
Within ±10° (%)	58.6	91.3	89.7	90.5

At 1 GHz, the Kalman post-filter reduces the MAE from 6.03° to 4.50° and the RMSE from 7.44° to 5.67°. At 433 MHz, the improvement is more pronounced: the MAE drops from 8.70° to 4.72° and the RMSE from 10.01° to 6.08°, with a clear smoothing of the trajectory. These results confirm that the chosen Kalman parameters (Q_{00}, Q_{11}, R) provides a reasonable balance between responsiveness and smoothing, without any modification to the core algorithm.

5.5.4 Results Summary

.

The system tracks the angular position of the RF source across all tested scenarios, though with different error levels depending on the operating frequency. The offline experiments con-

ducted on saved data confirmed the effectiveness of the algorithm under controlled conditions. However, in the recorded dataset, the indoor multipath environment introduces signal replicas that corrupt the coherence assumption underlying the signal extraction stage described in Chapter 2.3: the algorithm occasionally minimises the main branch output rather than the residual, causing transient divergences Kalman post-filter effectively attenuates these oscillations, confirming its value as a lightweight complement to the core algorithm. *"Indoor radio waves bounce around like light in a room full of mirrors, creating optical illusions (ghost signals) that confuse the algorithm, and that monopole antennas are physically blind to anything happening directly above them".*

5.6 Spectrum Sensing Implementation

The SCMAP-CFAR detector is implemented as a GNU Radio embedded Python block operating on the raw IQ stream. It does not require the FIR preprocessing or MUSIC initialisation described in Section 5.3, which are specific to the DoA chain. Two deployment configurations are considered: a single-band mode targeting a known sub-band within the 2.4 MHz instantaneous bandwidth, and a five-hop sweep mode covering a total band of 12 MHz.

5.6.1 Single-Band Mode and Subsystem Fusion

In the single-band configuration, the spectrum sensing block receives a 1024-point FFT vector, estimates the adaptive threshold bin by bin using the clutter map, and identifies the dominant spectral peak. When a detection is confirmed, the block issues a PMT[5] message carrying the detected baseband offset f_{off} to a downstream *Frequency Xlating FIR Filter*, which automatically recentres on the emitter frequency. The absolute frequency is reconstructed as:

$$f_{\text{abs}} = f_0 + f_{\text{off}}, \qquad f_{\text{off}} = \left(k^* - \frac{N}{2}\right)\frac{F_s}{N}, \tag{V.8}$$

where k^* is the detected bin index and $N = 1024$. This frequency resolution is $F_s/N = 2.34\,\text{kHz}$ per bin, consistent with the simulation parameters of Chapter 4.

In this configuration, the SCMAP output is directly compatible with the DoA preprocessing chain: the detected frequency seeds the FIR filter centre frequency f_c (Equation V.1), and the filtered IQ stream feeds the GSC-LMS tracker. **This constitutes the first successful partial fusion of the two subsystems**, operating jointly on a single sub-band without any frequency retuning overhead. Figure 5.20 shows the real-time Qt interface during a 1 GHz test, where the SCMAP block has locked onto the emitter and the DoA radar display reports $\hat{\theta} = -26.97°$.

[5]Polymorphic Type is the native message-passing format used by GNU Radio for inter-block communication. A PMT message is a typed key–value pair that travels asynchronously between blocks.



Figure 5.20: Real-time Qt DoA interface during the single-band fusion test at 1 GHz for RF signal and 1 MHz for baseband signal. The SCMAP block detected the emitter at $f_{\text{abs}} = 1000.984\,\text{MHz}$ and commanded the Xlating FIR filter accordingly. The system reports $\hat{\theta} = -26.97°$.

5.6.2 Real-Time Tracking Test

The offline validation confirmed algorithmic correctness. The system was then deployed in real time using the GNU Radio flowgraph shown in Appendix 3.3.

The IQ stream from the KrakenSDR passes through a chain of custom DSP blocks. An initial Python implementation was used for rapid prototyping but introduced a non-negligible processing latency: only one sample out of every 512 was effectively processed, causing the adaptive tracker to diverge. The blocks were rewritten as C++ OOT[6] modules, which eliminated the latency and enabled stable sample-by-sampl processing.

Several tests were conducted. The system performed well overall, with two observed limitations: a systematic angular bias of approximately 4° and high sensitivity to the initial locking angle, consistent with the locking interval analysis of Chapter 4.

Figures 5.21 and 5.22 show the Qt DoA interface during a joint SCMAP-CFAR and GSC-LMS real-time test. The emitter transmits a 1 MHz baseband signal modulated onto a 1 GHz carrier, falling within the instantaneous 2.4 MHz acquisition band.

[6]Out-Of-Tree modules are custom GNU Radio blocks developed outside the standard GNU Radio source tree

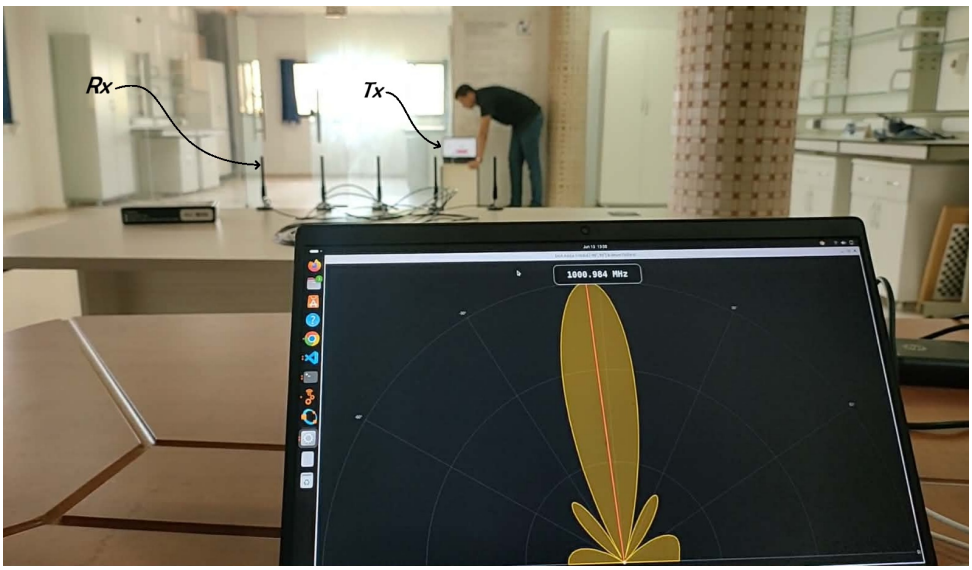


Figure 5.21: Real-time Qt DoA interface initialisation at broadside ($\hat{\theta} = 0°$). The SCMAP block has detected the emitter at $f_{\text{abs}} = 1000 + 1 = 1001\,\text{MHz}$.



Figure 5.22: Real-time Qt DoA interface successful tracking after the transmitter is moved to the left. The beam follows the source direction. The displayed baseband frequency (1 MHz modulate onto 1 GHz) confirms that both subsystems are operating jointly in real time.

5.6.3 Five-Hop Sweep Mode

When the target frequency is unknown, the system must scan a wider band. The KrakenSDR instantaneous bandwidth is fixed at 2.4 MHz, covering a 12 MHz scan band therefore requires five contiguous frequency hops. Each hop is commanded by a TCP retuning message sent to the Heimdall controller on port 5001 wich tune the LO frequency to the desired frequency.

Schematic 5.23 highlights the keys parts of this approach

The hardware constraints of the KrakenSDR impose two important limitations. First, the PLL requires approximately 1 seconds to stabilise after each retune, during which the received

IQ frames are invalid. Second, because Heimdall is single-threaded, the frequency command is processed in the main acquisition thread, which means the IQ queue must be flushed after each hop to eliminate residual samples from the previous centre frequency. These constraints motivated a dedicated CFAR-master architecture, described below.

5.6.4 Spectral CFAR-Master Architecture

An initial implementation used a separate Control Unit block to count detection messages and decide when to update parameters. This approach suffered from race conditions: detection messages arrive in bursts and the counter overflowed, causing the TRACK state to never be reached. The architecture was redesigned with the spectrum sensing block acting as master, handling all settling, analysis, and decision logic internally.

The flowgraph 4.2 comprises four main blocks whose roles are summarised in Table 5.8.

Figure 5.23: Multi-band or Wideband spectrum sensing pipeline

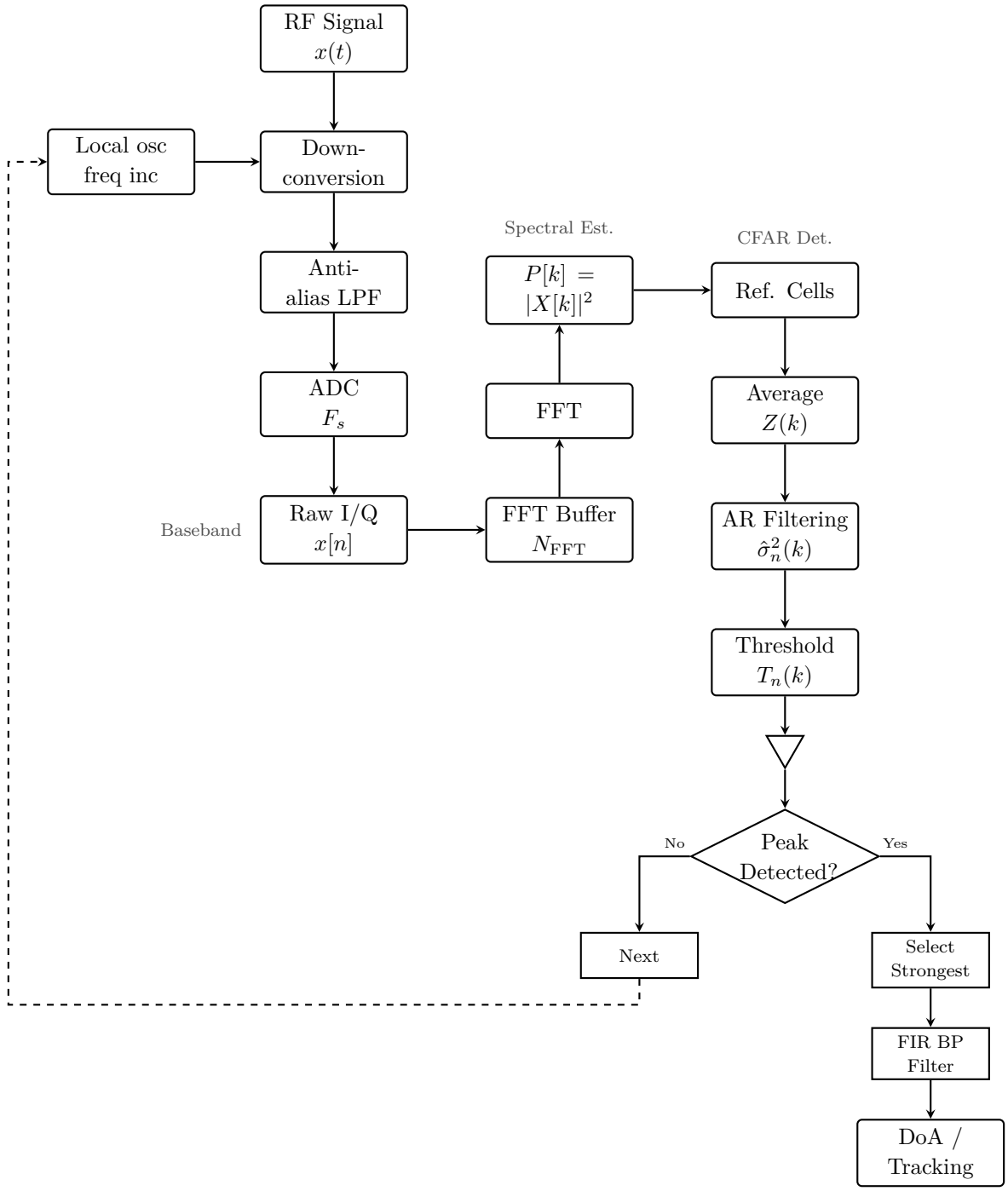


Table 5.8: GNU Radio embedded Python blocks .

Block	**Role**	**Type**
`cmap_cfar_spectrum.py`	CFAR master: settling, analysis, decision	Master
`control_unit.py`	Frequency sequencer, SOI locking	Slave
`krakensdr_source.py`	IQ streaming, frequency control via TCP	Source
`freq_display.py`	Qt interface: f_{abs}, LED status	Display

5.6.5 Scan State Machine

The CFAR master block operates through three sequential states. Table 5.9 lists the key parameters.

Table 5.9: Frequency sweep parameters.

Parameter	Value	Description
`Nfft`	1024	FFT size
`samp_rate`	2.4 MHz	Instantaneous bandwidth
Total scan band	12 MHz	5 hops 2.4 MHz
`settling_cpi`	24 CPI	Frames discarded after retune
`cpi_count`	100 CPI	Analysis frames per hop
`soi_threshold`	25	Consecutive detections to lock
K	10	CFAR threshold factor
λ	0.95	Clutter map memory coefficient

1. **Settling phase:** After each TCP retune command, the first 24 CPI [7] frames are discarded. This window covers the PLL stabilisation time ($\approx$ 1 s at 24 kHz) and eliminates calibration artefacts. The IQ queue is also flushed to remove residual samples from the previous centre frequency. No detection decision is made during this phase.

2. **Scanning phase.** The block accumulates 100 FFT frames and applies the CMAP-CFAR algorithm to each. The DC bin (bin 512, offset 0 Hz) is excluded from the search to prevent locking on the RTL-SDR DC offset. A detection event is counted each time the dominant peak exceeds the adaptive threshold.

3. **Decision and locking phase.** During the scanning process, each sub-band is observed over 100 consecutive CPI. For every scan, the CFAR detector reports whether a Signal of Interest has been detected. The master block then computes the detection ratio, defined as the fraction of CPI in which a detection occurred. This ratio is compared with the parameter `soi_threshold`, **which represents the minimum proportion of positive detections required to declare the presence of a valid signal in the current sub-band.** If the detection ratio exceeds this threshold, a `result=1` message is sent to the Control Unit, which switches to TRACK mode and freezes the frequency sweep. Otherwise, a `result=0` message instructs the Control Unit to continue scanning by tuning to the next centre frequency.
   On the locked or tracking mode the operator can also force a manual frequency change at any time via the *Skip* button in the Qt interface, which sends an asynchronous PMT command to the Control Unit.

[7] A *Coherent Processing Interval* (CPI) is the block of samples over which a single spectral estimate is computed; in our system, 1 CPI = 1024 samples given that each block accumulates 100 consecutive FFT frames sampled at a raw hardware rate of 2.4 MHz

The PMT message protocol between blocks is given in Table 5.10.

Table 5.10: Inter-block PMT message protocol.

Message	Direction	Format
`cmd_freq`	Control Unit → CFAR + KrakenSDR	`cons("freq", double)`
`detection`	CFAR → Control Unit	`cons("detect", bool)`
`detection_info`	CFAR → Qt display	`cons("peak", cons(`f_{abs}`, ...))`
`ui_status`	Control Unit → Qt display	`cons("locked"/"searching", double)`

5.7 Multi-band scanning Results

Figure 5.24 shows the sweep system locked onto a 1 GHz (Parameters used are mentioned upper in table 5.9) emitter in a low-SNR indoor environment (no peak is visible in the spectrum). The system correctly identifies the spectral activity (bin $\approx$ 780, offset $\approx$ +0.4 MHz) and reports a locked state at $f_{\text{abs}} = 1000.406$ MHz which corresponds to the ground truth, as we emit base band signal a 400 Khz modulated to 1Ghz. The binary integration counter shows 78 detections out of 100 accumulated frames, exceeding the locking threshold.

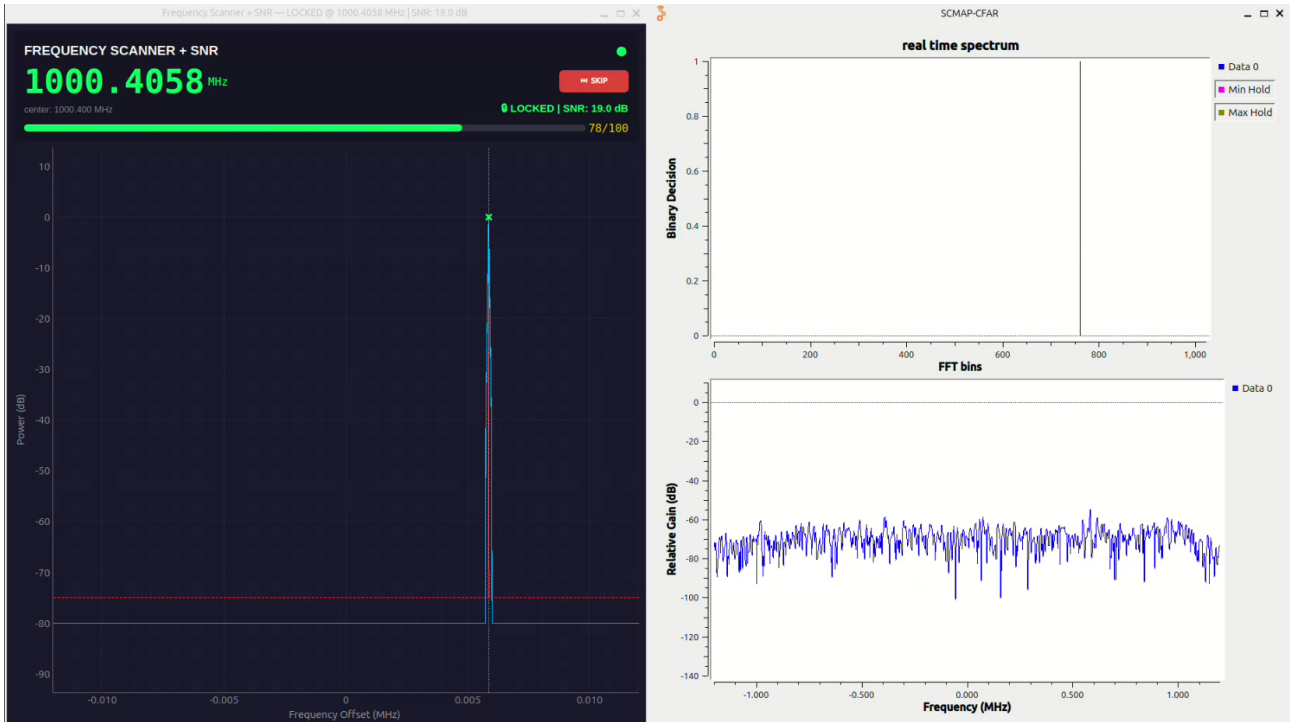


Figure 5.24: Sweep system locked at $f_{\text{abs}} = 1000.406$ MHz.

The results confirm that the proposed architecture resolves the race-condition issues of the initial Control-Unit-master design. The 24-CPI settling window reliably covers the KrakenSDR PLL stabilisation time, and the binary integration over 100 frames suppresses isolated noise threshold crossings without introducing false locks.

Note on UCA tracking implementation

Despite the encouraging results obtained in the linear configuration, the real-time extension of the algorithm to full 3-D source tracking yielded inconclusive outcomes. Preliminary tests conducted in an indoor environment revealed significant inconsistencies: both azimuth and elevation estimates were highly erratic and unreliable across all trials. Two compounding factors were identified as the primary causes of this degradation.

First, the indoor propagation environment introduces severe multipath and reflection artefacts, rendering elevation estimation particularly sensitive even under ideal array conditions.

Second, the monopole antennas used at Rx plateform exhibit a purely azimuthal radiation pattern: their far-field diagram provides no meaningful phase diversity in the elevation dimension, making any estimation ill-conditioned regardless of the algorithm employed. Given these hardware and environmental constraints, and in the interest of delivering a validated, reproducible system within the scope of this work, the decision was made to focus the experimental validation on the ULA-based azimuthal tracking subsystem and the SCMAP-CFAR spectrum sensing chain.

5.8 Conclusion

This chapter valides both subsystems on real hardware. The spectrum sensing system runs in real time within the GNU Radio framework, producing frequency and detection outputs consistent with the simulation results of Chapter 4. The GSC-LMS tracker converges on all static test angles at both 433 MHz and 1 GHz. The MUSIC-based initialisation proves reliable for seeding the adaptive filter within the locking interval, and the Eigen-based C++ implementation confirms its suitability for real-time deployment. The key enabling step for real-time operation was the migration from Python embedded blocks to C++ modules. Fusion of the two subsystems into a single GNU Radio flowgraph remains as a natural continuation of this work.

General Conclusion

This project addresses the problem of radio-frequency source tracking by combining adaptive beamforming and FFT-based spectrum sensing with CMAP-CFAR detection. Two complementary subsystems were studied, implemented, and validated on a KrakenSDR five-channel coherent receiver platform.

The first subsystem performs Direction of Arrival estimation and tracking via the GSC-LMS algorithm, which iteratively updates the steering vector and extracts the source signal through an implicit sidelobe canceller structure. The second subsystem performs frequency-domain signal detection via a CMAP-CFAR chain, which maintains an adaptive clutter map and applies a calibrated threshold to identify occupied spectral bins. Experimental results confirmed the feasibility and complementarity of both subsystems under real indoor propagation conditions, including multipath and moderate SNR degradation.

The system developed in this work finds natural applications in spectrum monitoring, cognitive radio, electronic warfare with targeted jamming, wireless communications infrastructure, and passive surveillance systems, where joint spatial and spectral awareness of RF emitters is required.

Regarding future developments, a low-level optimisation of the processing chain and a migration toward dedicated hardware platforms such as RFSoC or specialised DSP accelerators would significantly improve throughput and real-time capability. Extending the array geometry from the current ULA to a planar or circular configuration would further enable full 2D azimuth-elevation tracking. Finally, the acquisition pipeline already in place provides a straightforward path to building labelled RF datasets, opening the door to deep-learning-based DoA estimation as a longer-term research direction.

References

[1] Saeed Gazor. "Optimisation d'une antenne acoustique pour la prise de sons". ENST 94 E 021, soutenue le 2 novembre 1994. Thèse de doctorat. Paris, France: Ecole Nationale Supérieure des Télécommunications (Télécom Paris), 1994.

[3] Jeffrey Foutz, Andreas Spanias, and Mahesh K. Banavar. *Narrowband Direction of Arrival Estimation for Antenna Arrays*. Morgan & Claypool Publishers, 2008.

[4] Rodrigo Brandt et al. "A Study of Algorithms for Detecting the Direction of Arrival (DOA) of RF Signals". In: *Proceedings (conference unspecified)*. Universidade Federal do Rio Grande do Sul (UFRGS) and Universidade Federal de Santa Catarina (UFSC). Porto Alegre and Florianópolis, Brazil, Unknown.

[5] Mohammed Amine Ihedrane and Seddik Bri. "Direction of Arrival Using Uniform Circular Array Based on 2-D MUSIC Algorithm". In: *Indonesian Journal of Electrical Engineering and Computer Science* 12.1 (Oct. 2018), pp. 30–37. DOI: 10.11591/ijeecs.v12.i1.pp30-37.

[6] B. Widrow et al. "Adaptive Antenna Systems". In: *Proceedings of the IEEE* 55.12 (1967), pp. 2143–2159. DOI: 10.1109/PROC.1967.6092.

[7] Simon Haykin. *Adaptive Filter Theory*. 5th. International Edition. Upper Saddle River, NJ: Pearson, 2014.

[8] Will Tidd et al. "A RF Source Localization and Tracking System". In: *IEEE Military Communications Conference (MILCOM)*. 2010, pp. 858–863. DOI: 10.1109/MILCOM.2010.5680188.

[9] R. E. Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Transactions of the ASME–Journal of Basic Engineering* 82.1 (1960), pp. 35–45. DOI: 10.1115/1.3662552.

[10] M. Sanjeev Arulampalam et al. "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking". In: *IEEE Transactions on Signal Processing* 50.2 (Feb. 2002), pp. 174–188.

[11] T. Yücek and H. Arslan. "A survey of spectrum sensing algorithms for cognitive radio applications". In: *IEEE Communications Surveys & Tutorials* 11.1 (2009), pp. 116–130.

[12] Dayan Adionel Guimaraes. "Spectrum Sensing: A Tutorial". In: *Journal of Communication and Information Systems* 37.1 (2022), p. 10.

[13] R. Tandra and A. Sahai. "SNR walls for signal detection". In: *IEEE Journal of Selected Topics in Signal Processing* 2.1 (2008), pp. 4–17.

[14] Abdel Hamid Mbouombouo Mboungam, Yongfeng Zhi, and Cedric Karel Fonzeu Monguen. "Clutter Map Constant False Alarm Rate Mixed with the Gabor Transform for Target Detection via Monte Carlo Simulation". In: *Applied Sciences* (2024).

[15] Vladimir G. Galushko. "Analysis of the CA CFAR Algorithm as Applied to Detection of Stationary Gaussian Signals Against a Normal Noise Background". In: *Proceedings of the International Conference*. Institute of Radio Astronomy, National Academy of Sciences of Ukraine. Kharkiv, Ukraine, June 2016.

[16] H. M. Finn and R. S. Johnson. "Adaptive detection mode with threshold control as a function of spatially sampled clutter level estimates". In: *RCA Review* 29 (1968), pp. 414–464.

[17] H. Rohling. "Radar CFAR thresholding in clutter and multiple target situations". In: *IEEE Transactions on Aerospace and Electronic Systems* AES-19.4 (1983), pp. 608–621.

[18] Simon J. Julier and Jeffrey K. Uhlmann. "Unscented Filtering and Nonlinear Estimation". In: *Proceedings of the IEEE* 92.3 (2004), pp. 401–422.

[19] Henrique M. T. Menegaz et al. "A Systematization of the Unscented Kalman Filter Theory". In: *IEEE Transactions on Automatic Control* 60.10 (Oct. 2015), pp. 2583–2598.

[20] Shengguo Ge, Kuo Li, and Siti Nurulain Binti Mohd Rum. "Deep Learning Approach in DOA Estimation: A Systematic Literature Review". In: *Wireless Communications and Mobile Computing* 2022 (2022), pp. 1–18. DOI: 10.1155/2022/1032108.

[21] Hongmei Zhang, Xiwei Sun, and Mingjie Chen. "Azimuth Tracking for Underwater Moving Target Based on MUSIC and Function Approximation". In: *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE. Changchun, China, Aug. 2018, pp. 1–6.

[22] Y. Xiong, Y. Zhang, and L. Yang. "Deep Learning Based Target Tracking and Localization in Wireless Sensor Networks". In: *IEEE Access* 9 (2021), pp. 11845–11856. DOI: 10.1109/ACCESS.2021.3051234.

[23] H. E. Bouchelaghem et al. "Adaptive Clutter-Map CFAR Detection in Distributed Sensor Networks". In: *International Journal of Electronics and Communications (AEÜ)* (2016). Received: July 10, 2015; Accepted: June 29, 2016.

[24] Ramon Nitzberg. "Clutter Map CFAR Analysis". In: *General Electric Company* (1986). Technical Report.

[25] Abdellatif Rouabah et al. "Improving the Detection Quality of the Clutter Map-Constant False Alarm Rate Detector in a Non-Homogeneous Environment". In: *Research Article - Electrical Engineering* (2025). Received: July 4, 2024; Accepted: April 14, 2025.

[26] Mostafa Y. Rihan, Zaki B. Nossair, and Roaa I. Mubarak. "An improved CFAR algorithm for multiple environmental conditions". In: *Signal, Image and Video Processing* 18.4 (2024), pp. 3383–3393. DOI: 10.1007/s11760-024-03001-x.

[27] Louise H. Crockett, David Northcote, and Robert W. Stewart. *Software Defined Radio with Zynq UltraScale+ RFSoC*. Ed. by Louise H. Crockett, David Northcote, and Robert W. Stewart. First. Available online at https://www.rfsocbook.com. Strathclyde Academic Media, 2023. ISBN: 9781739588601.

[28] Kaushalya Adhikari and Ridhab Al Kinani. “Transformation of a circular array to a linear array with a Vandermonde structure”. In: *Digital Signal Processing* 140 (2023), p. 104098.

[29] Abdeldjalil Aïssa-El-Bey, Karim Abed-Meraim, et al. “Impulsive noise modeling and robust receiver design”. In: *EURASIP Journal on Advances in Signal Processing* 2020.1 (2021), pp. 1–32. DOI: 10.1186/s13638-020-01868-1.

[30] HeIMDALL DAQ Project. *HeIMDALL DAQ Firmware*. Version 1.0.20201130. Technical documentation for the HeIMDALL DAQ firmware, 52 pages. 2020.

[32] MathWorks. *MATLAB*. https://fr.mathworks.com/products/matlab.html. Accessed: June 7, 2026. 2026.

Webography

[2] CRFS. *Guide to RF Geolocation*. Accessed: 2026-05-25. 2024. URL: https://www.crfs.com/guide-to-rf-geolocation.

[31] GNU Radio Project. *About GNU Radio*. URL: https://www.gnuradio.org/about/ (visited on 06/06/2026).

Appendices

Appendix A

Kraken Hardware Specifications

1.1 KrakenSDR

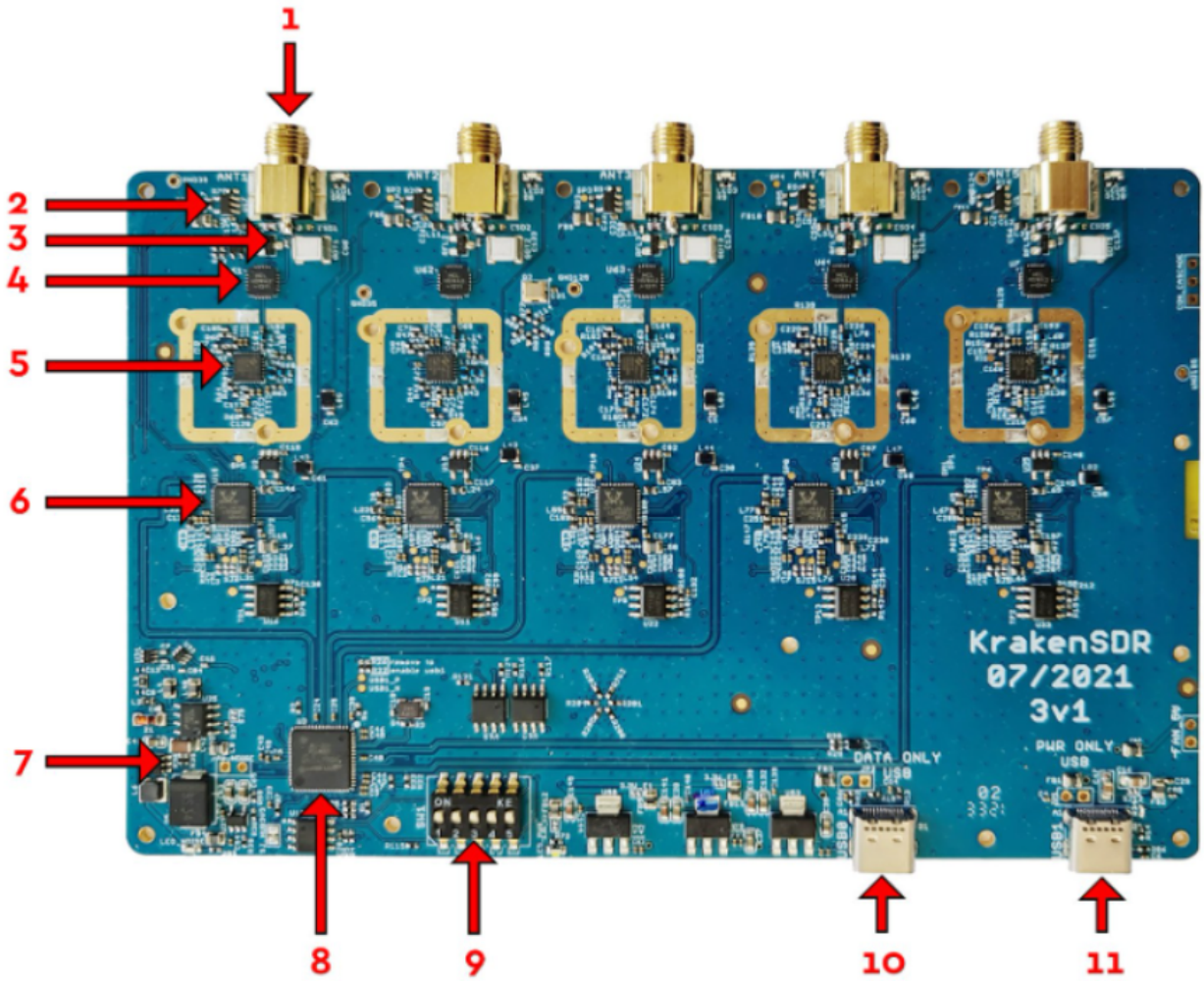


Figure 1.1: KrakenSDR PCB component overview. (1) SMA antenna inputs, (2) Bias Tee, (3) ESD protection, (4) Noise calibration switches, (5) R820T2 tuner, (6) RTL2832U ADC, (7) Noise source, (8) USB Hub, (9) Individual tuner on/off DIP switches, (10) USB TypeC DATA, (11) USB TypeC PWR.

Table 1.1: KrakenSDR technical specifications.

Parameter	**Value**
Dimensions	177112.325.86 mm
Weight	670 g
Power consumption	5 V, 2.2 A (11 W) typical
Radio tuner	5 R820T2
ADC	5 RTL2832U, 8bit
Frequency range	24 MHz 1766 MHz
Max channel BW	2.56 MHz
RX channels	5 (phasecoherent)
Oscillator stability	1 PPM

Appendix B

Heimdall DAQ Firmware

Integrating the KrakenSDR into our processing chain required a detailed study of the Heimdall DAQ firmware. The official documentation [30] served as the primary reference, complemented by functional reverse engineering of the modules directly involved in IQ transport and hardware control. The objective was to understand the acquisition pipeline, the control interfaces, and the data flow in order to adapt the GNU Radio OOT `KrakenSDR Source` block to our architecture.

2.0.1 Global Architecture

Figure 2.1 shows the internal structure of the firmware. The pipeline is organized as a chain of independent processing modules, each running on a dedicated thread. Three types of links are used:

- **IQ data path** (solid lines): frames pass through sharedmemory pingpong buffers between consecutive modules. Each buffer alternates between two slots (A/B); when one slot is written, a FIFO notification unblocks the downstream module.
- **Control FIFOs** (dashed): slowspeed interprocess commands (gain, frequency, squelch threshold) are exchanged via Linux named pipes (`_data_control/rec_control_fifo`, etc.).
- **Ethernet** (dotted): remote configuration (port 5001) and IQ data export (port 5000) use TCP.

The data path from the antenna to the output goes through the following stages: the RF front-end handles amplification and analog-to-digital conversion, then the IQ samples travel over USB into the `Realtek DAQ`, followed by `Sync`, `Rebuffer`, `Decimator`, `Squelch`, and `Delay Synchronizer`, before reaching the `IQ Server`. In parallel, the `Hardware Controller` runs on a separate thread: it carries no IQ samples, but instead reads IQ frame headers and injects commands into the data path through FIFOs.

This separation between the IQ path and the control path was the key architectural insight for our integration: modifying the frequency requires only a TCP message to port 5001, without interrupting the IQ stream, only cleaning it form the dummy frames that flows at the calibration stage.

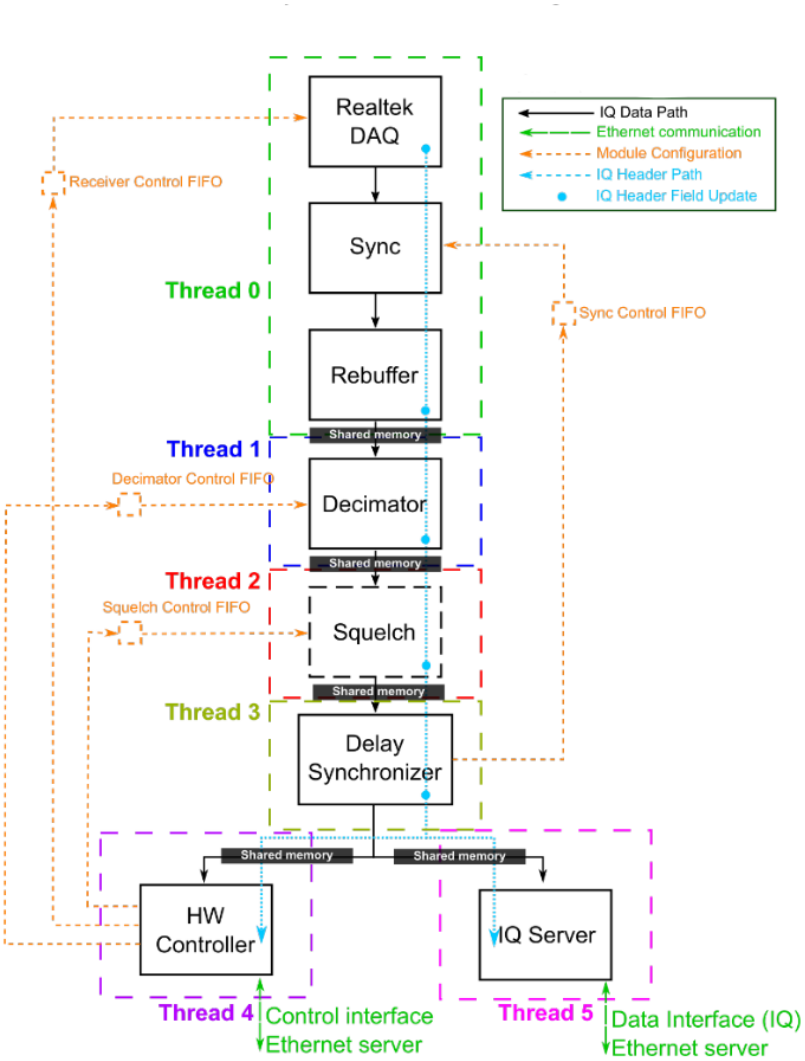


Figure 2.1: Heimdall DAQ firmware block diagram.

2.0.2 Principal modules used for Integration

Hardware Controller

The Hardware Controller (Thread 4) exposes a TCP control server on port 5001. It accepts 128byte messages whose first four bytes encode the command word. The relevant command for our work is:

`FREQ` carries the new centre frequency as a `uint64` value in Hz.

The controller injects the frequency change into the `rec_control_fifo`, which the `Realtek DAQ` module reads on its next iteration. Because Heimdall is singlethreaded within each module, the command is processed synchronously in the main acquisition loop there is no interrupt mechanism. After a retune, several dummy frames are emitted while the PLL stabilises ($\approx 3\,\mathrm{s}$); these must be discarded by the downstream chain.

Understanding this mechanism allowed us to implement the frequency sweep in our OOT block: each hop sends a single `FREQ` command over TCP and then waits for `settling_cpi` frames before resuming detection.

IQ Server

The IQ Server (Thread 5) reads processed frames from the last sharedmemory buffer (output of the Delay Synchronizer) and streams them over TCP on port 5000. It uses internal double-buffering: one buffer is being filled from shared memory while the other is being transmitted over Ethernet, so the two operations run in parallel.

Each frame carries a structured header containing synchronisation flags (`sample_sync_flag`, `iq_sync_flag`), the frame type (normal / calibration / dummy), and metadata. Our `KrakenSDR Source` block reads this header to detect and discard calibration and dummy frames automatically.

The IQ payload is formatted as `complex float32`, M`cpi_size` samples per frame, with all five channels interleaved. This is the format consumed directly by our GNU Radio chain after a `StreamstoStream` demultiplexing step.

2.0.3 Role of `daq_chain_config.ini`

All firmware parameters are set at startup via the `daq_chain_config.ini` file. Two parameters had a critical impact on our acquisition stability:

Table 2.1: Key `daq_chain_config.ini` parameters adjusted during integration.

Parameter	Default	adapted value	Effect
`daq_buffer_size`	$256Ko$	$512Ko$	Eliminates USB buffer overflow
`cpi_size`	2^{18}	1024	Matches our FFT size
`decimation_ratio`	1	100	Reduces output rate to 24 kHz
`cal_track_mode`	0	2	Burst calibration between hops

`daq_buffer_size`. This parameter sets the number of IQ samples read from each RTLSDR USB buffer in one call. If set too small, the `Realtek DAQ` module must poll the USB buffers too frequently; under CPU load this causes sample loss. If set too large, downstream modules cannot drain the buffer in time. We observed persistent framedrop warnings (`shmemIface: Dropping frame`) with the default value. Increasing `daq_buffer_size` to 2^{21} resolved all overflow conditions in our setup.

`cpi_size`. This parameter defines the number of samples per output frame. The Rebuffer module adapts the block size from `daq_buffer_size` to `cpi_size decimation_ratio`. Setting `cpi_size = 1024` aligns the Heimdall output frames directly with our 1024point FFT blocks, eliminating any additional rebuffering in GNU Radio.

`cal_track_mode = 2`. In burst tracking mode, Heimdall periodically injects calibration frames to verify IQ synchrony. After each TCP frequency update, the first frames are calibration or dummy type. Our settling logic discards exactly these frames by checking the `frame_type` field in the IQ header before passing data to the CFAR block.

2.0.4 OOT Block Modifications

The stock `KrakenSDR Source` block provided by the KrakenRF project reads IQ frames from port 5000 and outputs them as a fixed frequency stream. It offers no mechanism for dynamic frequency retuning from within a GNU Radio flowgraph.

Our modifications added three capabilities:

1. **PMT message input.** A message port named `cmd_freq` was added. When a `cons("freq", double)` PMT message arrives, the block opens a TCP connection to port 5001 and sends the `FREQ` command with the new centre frequency encoded as `uint64`. The connection is closed immediately after the command is acknowledged, consistent with the singleclient constraint of the Heimdall control server.

2. **Frametype filtering.** The block reads the IQ frame header on each iteration. Dummy and calibration frames are silently dropped and replaced by zerofilled output buffers, so that downstream blocks always receive validlength vectors without stalling.

3. **Settling counter.** After each retune command, an internal counter suppresses output for `settling_cpi` frames (= 15 in our configuration), covering the PLL stabilisation window identified in the firmware documentation.

These modifications made the `KrakenSDR Source` block a firstclass participant in the GNU Radio message bus, enabling the CFAR master block to orchestrate the full fivehop frequency sweep via PMT messages without any external scripting or manual intervention.

Appendix C

Software Tools

3.1 GNU Radio

GNU Radio is a free, opensource signalprocessing toolkit for building softwaredefined radio systems. It can be used either with external RF hardware or in pure simulation mode. In this work, it was used to design the realtime acquisition and processing flowgraphs for both the SCMAPCFAR sensing chain and the DoA tracking subsystem [31].



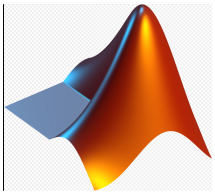
3.2 PyQt

PyQt is a Python binding of the Qt framework dedicated to graphical user interface development. In this work, it was used to implement the graphical interfaces presented in Chapter 5, providing realtime monitoring, visualization, and user interaction during the experiments.



3.3 MATLAB

MATLAB is a numerical computing environment used for algorithm development, data analysis, and visualization [32]. In this work, it served as an offline prototyping and validation platform before realtime deployment in GNU Radio.



Appendix D

GNU Radio Flowgraphs

This appendix gathers the main GNU Radio flowgraphs used during the experimental implementation of the proposed SDR platform.we exposes first the HackRF emission flowgraph, the the wideband frequency sweep mode, the monoband detection and adaptive tracking mode, and the offline IQ recording chain used for postprocessing and validation.

4.1 HackRF Transmission Flowgraph

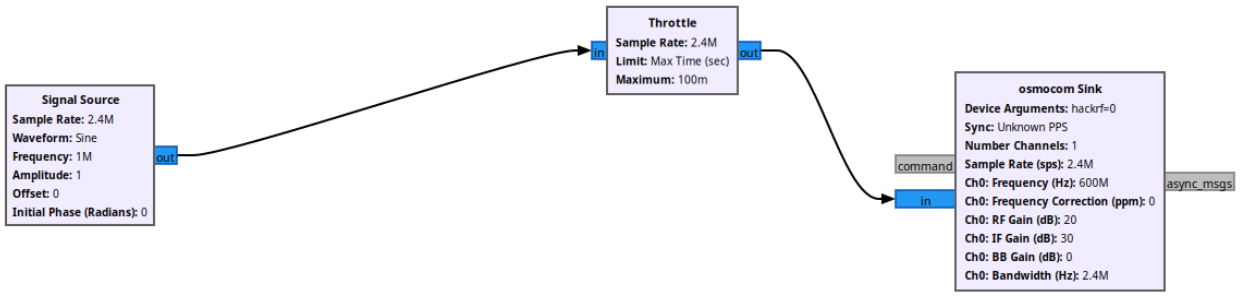


Figure 4.1: GNU Radio flowgraph for RF signal transmission via HackRF One. Blocks: Signal Source → Throttle → Osmocom Sink.

4.2 Frequency Sweep Flowgraph

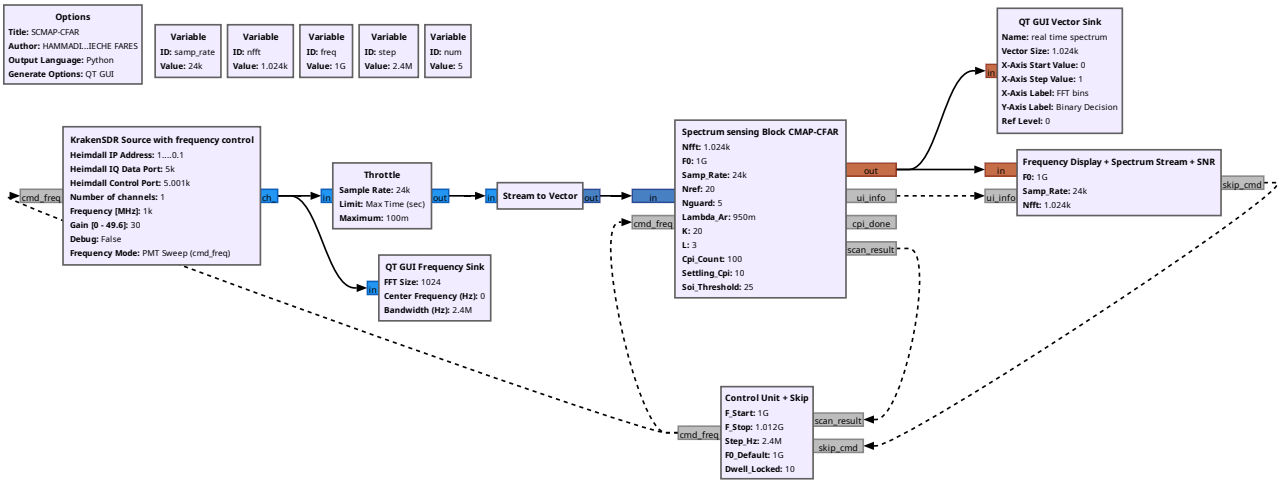


Figure 4.2: GNU Radio flowgraph used for wideband frequency sweep and spectrum sensing.

This flowgraph was used to scan a predefined frequency range and detect active RF emissions in real time. It implements the sweep mode of the sensing system and provides the input data required by the detection stage before switching to a narrower operating band.

4.3 MonoBand Detection and Adaptive Tracking Flowgraph

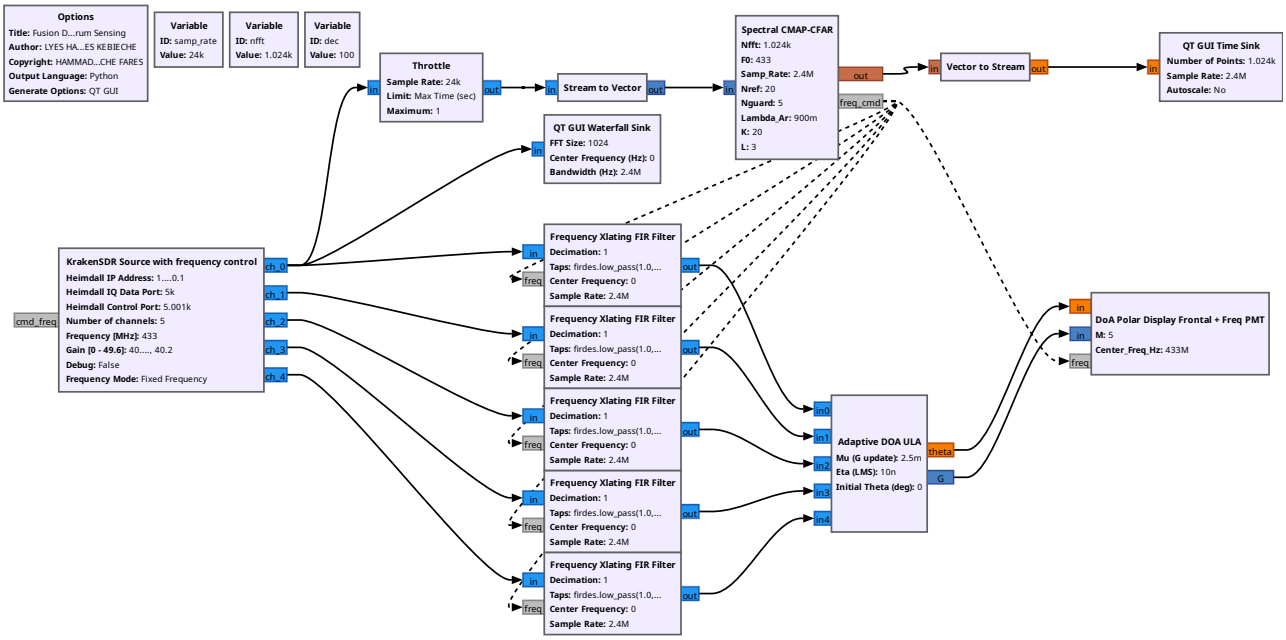


Figure 4.3: GNU Radio flowgraph used for monoband detection and adaptive DoA tracking.

This flowgraph corresponds to the realtime operating mode used after signal detection. It combines the sensing stage with the adaptive tracking block in order to monitor a selected band

and estimate the source direction continuously.

Frequency Xlating FIR Filter

This block combines in a single pass the three operations that would otherwise require separate stages: frequency translation, low-pass filtering, and decimation. Conceptually, it is equivalent to first mixing the input signal with a complex exponential to shift the band of interest to baseband, then applying a FIR low-pass filter to reject out-of-band components, and finally downsampling by the decimation factor. The analogy is that of a radio tuner that simultaneously tunes to a station, cuts adjacent channels, and reduces the data rate in one atomic operation.

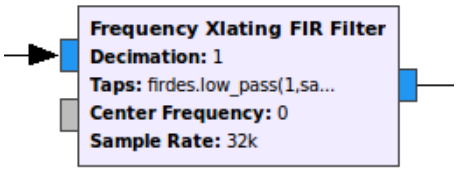


Figure 4.4: Frequency Xlating FIR Filter block in GNU Radio.

4.4 Offline IQ Recording Flowgraph

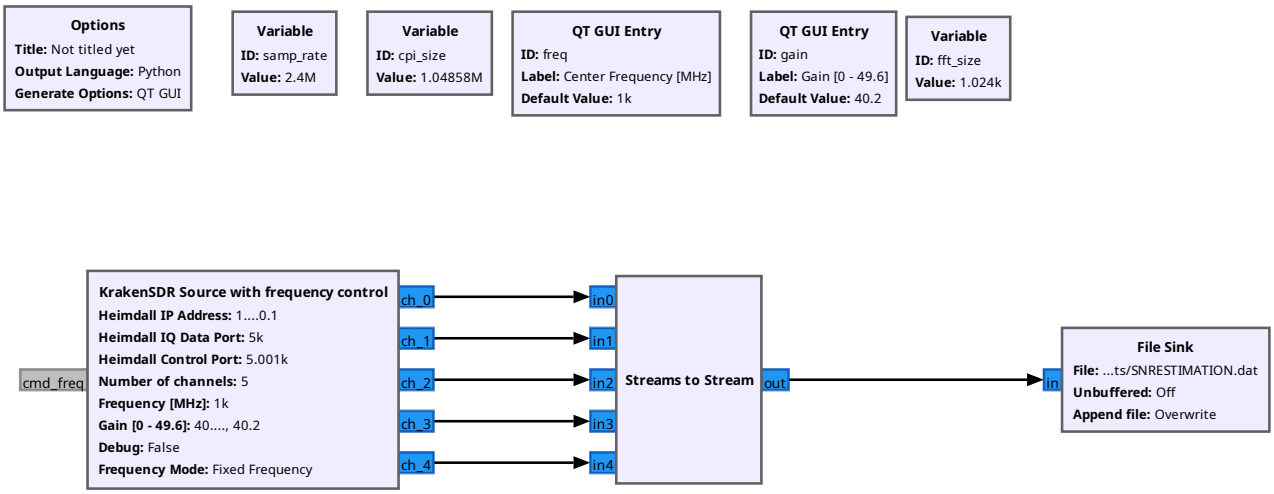


Figure 4.5: GNU Radio flowgraph used for multichannel IQ data recording.

This flowgraph was used to record raw multichannel IQ streams for offline analysis. It was mainly employed to build test datasets, validate the implemented algorithms, and replay real measurements outside the realtime acquisition chain.