

Diploma Project

Network Traffic Analysis on remotely operated drones using machine learning

Andreas Charalambous



University of Cyprus
**Department of Computer
Science**

Department of Computer Science
May 2026

UNIVERSITY OF CYPRUS

Faculty of Pure and Applied Sciences

Department of Computer Science

Network Traffic Analysis on remotely operated drones using machine learning

Andreas Charalambous

Advisor:

Dr Panagiotis Kolios

The Thesis was submitted in partial fulfillment of the requirements for obtaining the Computer Science degree of the Department of Computer Science of the University of Cyprus

May 2026

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Panagiotis Kolios, for his invaluable guidance, continuous support, and insightful direction throughout the course of this thesis. His expertise and encouragement were instrumental in shaping this research and bringing it to fruition. I am also deeply grateful to the members of the Research Center "Koios" Nikolas Soulis, George Pettemerides, and Yannis Grigoriou for their collaboration, technical support, and the stimulating research environment they helped foster. Their contributions and willingness to share their knowledge greatly enriched this work. I would also like to extend my appreciation to the University for providing the resources and academic foundation that made this research possible.

Finally, I wish to thank my family and friends for their unwavering support, patience, and encouragement throughout this journey. Their belief in me has been a constant source of motivation.

ABSTRACT

Unmanned Aerial Vehicles (UAVs) have become ubiquitous in commercial, recreational, and industrial applications. As drone adoption grows, so do concerns regarding security, privacy, and unauthorized surveillance. Modern consumer drones, such as those manufactured by DJI, employ robust encryption protocols, including AES-256, to protect their communication links, making conventional packet inspection infeasible for monitoring and classification purposes. This thesis presents a novel approach to drone behavioral classification through passive Radio Frequency (RF) physical layer analysis. Using a USRP B210 Software Defined Radio, we capture raw In-phase/Quadrature (IQ) samples from the 5.8 GHz communication band of DJI drones operating the OcuSync protocol, and extract 46 engineered features spanning energy, amplitude, phase, spectral, and temporal domains. A Random Forest classifier trained on these features achieves 96% precision in distinguishing three operational states — no drone present, drone idle, and drone with active camera movement — without ever decrypting the underlying data. Additionally, we investigate the security posture of DJI drones through RF jamming experiments, GPS spoofing attacks, and replay attack analysis, revealing critical asymmetries in how OcuSync protects video down-link versus control up-link channels. Real flight experiments validate the approach under field conditions while exposing the practical limitations of RSSI-based distance estimation in weak-signal regimes. Our results demonstrate that encrypted drone communications remain vulnerable to side-channel analysis at the physical layer, enabling passive surveillance detection and behavioral inference purely from RF traffic patterns.

Keywords: UAV, Drone Security, RF Traffic Analysis, Machine Learning, Software Defined Radio, IQ Samples, OcuSync, Physical Layer Analysis, GPS Spoofing, RF Jamming

TABLE OF CONTENTS

ABSTRACT	iii
TABLE OF CONTENTS	iv
LIST OF TABLES	vi
LIST OF FIGURES	vii
1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement	2
1.3 Research Objectives	2
1.4 Core Hypothesis	3
1.5 Contributions	3
1.6 Document Organization	4
2 Literature Review	5
2.1 UAV Communication Protocols	5
2.2 Software Defined Radio in Security Research	6
2.3 RF Traffic Analysis	7
2.4 Machine Learning for RF Classification	8
2.5 Drone Detection and Classification Research	9
2.6 RF Jamming and Anti-Jamming Techniques	10
2.7 GPS Spoofing Attacks on UAVs	10
2.8 Replay Attacks on Wireless Systems	11
2.9 Summary and Research Gap	11
3 System Architecture	13
3.1 Hardware Setup	13
3.2 Software Architecture	15
3.3 Data Collection Methodology	16

3.4	Feature Engineering Framework	18
3.5	Machine Learning Pipeline	22
4	Experimental Results	24
4.1	Simulation Mode Baseline	24
4.2	Stationary Laboratory Classification (M300 RTK)	27
4.3	Camera Movement Detection (Binary Classification)	39
4.4	RF Jamming Experiments	41
4.5	GPS Spoofing Experiments	46
4.6	Replay Attack Analysis	49
4.7	Real Flight Validation	51
4.8	Real Flight with Camera (April 6, 2026)	56
4.9	Improved Real Flight — April 16, 2026	67
4.10	Wideband OcuSync FHSS Spectrogram	88
4.11	Summary of Experimental Results	97
5	Conclusion	98
5.1	Summary of Key Findings	98
5.2	Research Contributions	100
5.3	Limitations	101
5.4	Future Work	101
	BIBLIOGRAPHY	103

LIST OF TABLES

3.1	System Specifications	13
3.2	Effect of fingerprint bin size on 50 μ s bursts	14
3.3	Drone operational states and corresponding datasets	18
3.4	V1 Energy/Activity Features	19
3.5	V3 Inter-Window Features	21
3.6	Model Comparison — Three-Class Lab Classification (M300 RTK, 70/15/15 stratified split, 675 test samples	22
4.1	Key Lab Results (March 3, 2026)	27
4.2	Comparison of visualization axes between Shi et al. and this work	34
4.3	Confusion matrix for camera movement classification	40
4.4	System behavior during RF jamming	42
4.5	OcuSync 2.0 Data-Link Channel Map (5.8 GHz Band)	46
4.6	Successful GPS Spoofing Configuration	48
4.7	Comparison between traffic analysis and replay attack approaches	51
4.8	Camera Flight Dataset Summary	56
4.9	Parameter changes between April 6 and April 16 experiments	67
4.10	April 16, 2026 Flight Dataset Summary	68
4.11	Fingerprint Metrics Across All Experiments	74
4.12	Parameter Change Impact Analysis	77
4.13	Comparison of frequency hopping coverage between default and custom bands	77
4.14	Cross-Flight Comparison — All Three Real Flights	78
4.15	Effect of hop duration and signal strength on feature interpretation	87
4.16	Narrowband vs. Wideband Capture Comparison	89
4.17	Narrowband Fingerprint vs. Wideband STFT Spectrogram	95
4.18	Wideband Spectrogram Development — Problems and Solutions	96
4.19	Summary of All Experimental Results	97

LIST OF FIGURES

3.1	RF signal processing and attack pipeline	15
4.1	Confusion matrix	24
4.2	Feature Importance	25
4.3	Feature Comparison	26
4.4	Summary Report	26
4.5	V1 Size comparison	28
4.6	V2 Amplitude Phase Features	28
4.7	Fingerprint Analysis (M300 RTK, Lab, Stationary)	30
4.8	TDD Power Analysis	32
4.9	TDD Histograms	33
4.10	Fingerprint Heatmap	34
4.11	Spectrograms of various drones from the Shi et al paper	35
4.12	OcySync RF Spectrogram of Matrice 300 RTK	36
4.13	Before and after the Denoising	37
4.14	Ocusync Spectrogram	38
4.15	IQ All features	40
4.16	IQ Size Comparison	41
4.17	Power Distance Correlation	52
4.18	Power Loss Point Approximation	52
4.19	Fingerprint Analysis (M300 RTK, Real Flight, March 26, Mobile)	54
4.20	Power over Time	57
4.21	Camera Noise Reprocessed	58
4.22	Distance Estimation	59
4.23	Fingerprint Analysis (M300 RTK, Real Flight, April 16, Mobile)	60
4.24	Fingerprint Analysis vs Baseline (M300 RTK, Real Flight, March 26, Mobile)	61
4.25	Phase Fingerprint	62
4.26	Multi Flight Overlay	63
4.27	Comparison Boxplots	64
4.28	Landing Detection	65

4.29 Previous Flight	66
4.30 Summary Table	68
4.31 Power Comparison	69
4.32 Timeseries	70
4.33 Power CDF	70
4.34 Fingerprint	71
4.35 Comprehensive RF Fingerprint Analysis — April 16 Flights (gain=70 dB, 2 MHz, noise-floor=-45 dB)	73
4.36 Distance Metrics	75
4.37 Detection Tracking	81
4.38 Packet Size Dictance Proxy	84
4.39 Active Energy Distance Proxy	86
4.40 Spectrogram Amplitude	89
4.41 Spectrogram Amplitude	90
4.42 Spectrogram	94

1 Introduction

1.1 Motivation

The proliferation of Unmanned Aerial Vehicles (UAVs), commonly known as drones, has transformed numerous sectors, from aerial photography and agricultural monitoring to infrastructure inspection and emergency response. The global drone market has experienced rapid growth, with consumer and commercial UAV shipments expanding dramatically over the past decade. However, this widespread adoption has simultaneously introduced significant security and privacy challenges that demand rigorous academic and practical investigation.

Modern consumer drones communicate with their ground controllers through sophisticated digital radio links. DJI, the dominant manufacturer controlling approximately 70% of the global consumer drone market, employs its proprietary OcuSync communication protocol across its product lines, including the Mavic, Mini, and Matrice series. OcuSync implements AES-256 encryption, unique per-session cryptographic keys, sequence numbering, and timestamp validation, rendering traditional network traffic inspection — such as deep packet inspection or protocol-level analysis — entirely ineffective against these platforms.

Despite these strong cryptographic protections, a fundamental question arises: "Can the operational behavior of an encrypted drone be inferred without decrypting its communications?" The answer, as this thesis demonstrates, is affirmative. Even when the content of transmissions is entirely opaque, the physical characteristics of the radio signal — its power, bandwidth, temporal patterns, and phase behavior — carry information about the drone's operational state. This observation represents a side-channel vulnerability inherent to any wireless communication system, regardless of its encryption strength.

The motivation for this research is threefold. First, from a security perspective, understanding what information encrypted drone communications leak through side channels is essential for developing comprehensive threat models. Second, from a surveillance detection point of view, the ability to passively detect and classify drone behavior has applications in law enforcement, critical infrastructure protection, and privacy preservation. Third, from an academic perspective, the intersection of Software Defined Radio, signal processing, and machine learning applied to drone communications represents a fertile and underexplored research domain that contributes to the broader fields of RF fingerprinting, traffic analysis, and adversarial machine learning.

1.2 Problem Statement

The central problem addressed by this thesis is the classification of encrypted drone communication states through passive physical layer RF analysis. Specifically, we investigate whether machine learning algorithms, trained on features extracted from raw IQ samples captured by a Software Defined Radio, can reliably distinguish between different drone operational states — including the presence or absence of a drone, idle versus active transmission modes, and behavioral variations such as camera movement — without any access to the encrypted payload data.

Beyond classification, this thesis examines the security posture of DJI drone communication systems through three categories of active attacks:

- **RF Jamming:** Investigating the vulnerability of OcuSync’s Time-Division Duplex (TDD) architecture to broadband noise jamming, and characterizing the asymmetric resilience between video down-link and control up-link channels.
- **GPS Spoofing:** Evaluating the susceptibility of DJI drones and consumer GPS devices to fabricated GPS L1 signals generated through a USRP SDR, and analyzing the anti-spoofing mechanisms employed by modern navigation systems.
- **Replay Attacks:** Assessing whether captured raw RF signals can be replayed to control or impersonate a DJI drone, and understanding the cryptographic and protocol-level defenses that prevent such attacks.

1.3 Research Objectives

This thesis pursues the following research objectives:

- Design and implement a passive RF capture and feature extraction pipeline using a USRP B210 Software Defined Radio, capable of extracting meaningful physical layer features from encrypted drone communications in real time.
- Engineer a comprehensive feature set from raw IQ samples that captures energy, amplitude, phase, spectral, and temporal characteristics of drone RF signals without requiring decryption.
- Train and evaluate machine learning classifiers (Random Forest, XGBoost) on the extracted features to accurately distinguish between drone operational states, achieving classification accuracy suitable for practical deployment.
- Characterize the OcuSync TDD structure through RF analysis, demonstrating the ability to observe time-division patterns, burst timing, and channel utilization from passive

observations alone.

- Evaluate the security posture of DJI drone communication systems against jamming, GPS spoofing, and replay attacks, identifying vulnerabilities and characterizing defensive mechanisms.
- Validate the approach through real flight experiments, assessing the practical limitations of RF-based drone analysis under field conditions, including the effects of distance, signal attenuation, and noise floor limitations.

1.4 Core Hypothesis

The central hypothesis of this thesis states:

Even with fully encrypted drone communications, behavioral classification is possible through RF physical layer analysis. Passive traffic analysis of encrypted communication metadata at the physical layer enables reliable inference of drone operational states, highlighting a side-channel vulnerability independent of payload encryption.

This hypothesis rests on the observation that encryption protects the content of communication but not its physical characteristics. As an illustrative analogy: encryption scrambles the letter inside a sealed envelope, but an observer can still measure the envelope’s weight, size, and frequency of delivery to infer the nature of the correspondence. Similarly, we measure the power, bandwidth, phase behavior, and temporal patterns of the radio signal envelope to classify drone behavior without ever ”opening” the encrypted payload.

1.5 Contributions

This thesis makes the following contributions to the field:

- A 46-feature RF characterization framework for drone traffic analysis, organized into three progressive layers: energy/activity features (V1), amplitude/phase statistics and fingerprints (V2), and inter-window temporal features (V3). This framework enables comprehensive physical layer characterization from raw IQ samples.
- Empirical demonstration that a Random Forest classifier achieves 96% accuracy in three-class drone state classification and 95.2% accuracy in binary camera-movement detection, using only physical layer features from encrypted OcuSync communications.
- Characterization of OcuSync’s asymmetric TDD resilience, showing through jamming experiments that video downlink is significantly more vulnerable to RF jamming than

control uplink, due to fundamentally different bandwidth requirements and error correction capabilities.

- Systematic evaluation of GPS spoofing against consumer devices and DJI drones, documenting the multiple anti-spoofing mechanisms employed by modern navigation systems and identifying the conditions under which spoofing succeeds.
- Real flight validation including path-loss analysis, distance estimation limits, and the identification of a distance-blind region where RSSI-based tracking becomes unreliable due to noise floor proximity.

1.6 Document Organization

This thesis is organized into five chapters:

Chapter 1 (Introduction): presents the motivation, problem statement, research objectives, core hypothesis, and contributions of this work.

Chapter 2 (Literature Review): surveys the relevant academic literature on UAV communication protocols, Software Defined Radio, RF traffic analysis, machine learning for signal classification, and drone security research.

Chapter 3 (System Architecture): describes the hardware setup, software pipeline, data collection methodology, feature engineering framework, and machine learning pipeline used throughout the experimental work.

Chapter 4 (Experimental Results): presents the findings from stationary laboratory experiments, camera movement detection, RF jamming, GPS spoofing, replay attacks, and real flight validation, with detailed analysis of each experiment's outcomes and implications.

Chapter 5 (Conclusion): summarizes the key findings, discusses the limitations of the study, and outlines directions for future research.

2 Literature Review

2.1 UAV Communication Protocols

2.1.1 DJI OcuSync

DJI's OcuSync protocol family, introduced with the Mavic Pro in 2016, represents the state of the art in consumer UAV communication. The protocol has evolved through several generations:

- OcuSync 1.0: First generation, supporting dual-band operation on 2.4 GHz and 5.8 GHz with automatic frequency selection.
- OcuSync 2.0: Enhanced version used in the Mavic 2 series and DJI Mini, featuring improved range (up to 8 km), higher video resolution support, and adaptive frequency hopping.
- OcuSync Enterprise (O3 Enterprise): Industrial variant used in the DJI Matrice 300 RTK, featuring enhanced security, longer range, and higher transmit power (up to 26 dBm EIRP).

All OcuSync variants share key architectural features relevant to this research [1, 2]:

1. Dual-Band Operation: The protocol can operate simultaneously on 2.4 GHz (2.400–2.4835 GHz) and 5.8 GHz (5.725–5.850 GHz). In dual-band mode, video and control data can be distributed across both bands for redundancy and performance.

2. Time-Division Duplexing (TDD): OcuSync uses TDD rather than Frequency-Division Duplexing (FDD). The drone and controller share the same frequency channel but transmit in alternating time slots. Experimental observations indicate a TDD frame period of approximately 10 ms, with approximately 85% allocated to video downlink, 10% to control uplink, and 5% to telemetry.

3. OFDM Modulation: The protocol employs Orthogonal Frequency-Division Multiplexing (OFDM), distributing data across multiple subcarriers within approximately 10 MHz channels.

4. Adaptive Frequency Hopping (AFH): OcuSync monitors the RF spectrum and dynamically switches between channels to avoid interference. In the 5.8 GHz band, research has identified approximately 10 discrete channels between 5.7515 GHz and 5.8515 GHz, each approximately 10 MHz wide. The drone scans all channels and selects the one with the lowest measured interference (RSSI), with hops occurring approximately every 600 ms [3, 4].

5. AES-256 Encryption: All data — including video, telemetry, and control commands — is encrypted using AES-256. A random, unique session key is generated each time a drone is

powered on, and the controller and drone share paired cryptographic keys established during the binding process [2].

2.1.2 WiFi-Based Drone Communication

Lower-cost consumer drones, such as early DJI models and many hobby-grade UAVs, utilize standard IEEE 802.11 (WiFi) for communication. These systems operate on the 2.4 GHz ISM band and are significantly more susceptible to traffic analysis, as standard network inspection tools can capture and analyze WiFi frames. However, this thesis focuses on OcuSync-equipped drones, which present a more challenging and security-relevant analysis target due to their proprietary, encrypted protocol stack.

2.1.3 Other Proprietary Protocols

DJI's earlier Lightbridge protocol and protocols used by other manufacturers (such as Skydio's proprietary communication system) share similar principles of encrypted, frequency-hopping digital communication. While the specific implementation details differ, the physical layer analysis techniques developed in this thesis are protocol-agnostic and can be adapted to any digital RF communication system.

2.2 Software Defined Radio in Security Research

Software Defined Radio (SDR) has emerged as a fundamental tool for wireless security research. By implementing radio functionality in software rather than dedicated hardware, SDR platforms provide the flexibility to receive, analyze, and generate RF signals across wide frequency ranges with programmable parameters.

2.2.1 SDR Platforms

The Ettus Research Universal Software Radio Peripheral (USRP) family represents the most widely used SDR platform in academic research. The USRP B210, used in this thesis, features:

- Full-duplex operation with independent transmit (TX) and receive (RX) channels
- Frequency range from 70 MHz to 6 GHz
- Up to 56 MHz instantaneous bandwidth
- 12-bit ADC/DAC resolution
- USB 3.0 connectivity for real-time streaming
- UHD (USRP Hardware Driver) software interface

The USRP B210's frequency range of up to 6 GHz makes it particularly suitable for analyzing DJI drones operating on the 5.8 GHz band, while its full-duplex capability enables simultaneous spectrum monitoring and signal transmission for active attack experiments [5].

2.2.2 GNU Radio

GNU Radio provides an open-source signal processing framework that enables rapid prototyping of radio systems. It consists of a library of signal processing blocks connected through a dataflow paradigm, allowing researchers to construct complex radio systems by chaining together modular components. In this thesis, GNU Radio is used for constructing the jamming signal generator and spectrum monitoring tools, with custom Python blocks for real-time signal analysis [6].

2.3 RF Traffic Analysis

RF traffic analysis — the practice of inferring information about communication systems from their physical layer characteristics without accessing the encrypted content — has a long history in signals intelligence (SIGINT) and has gained renewed academic interest with the proliferation of wireless devices.

2.3.1 Physical Layer Feature Extraction

The fundamental premise of RF traffic analysis is that even encrypted communications exhibit measurable physical layer characteristics that correlate with the nature of the underlying data. Key physical layer features include [7, 8]:

- **Signal Power (RSSI):** The received signal strength provides information about transmitter distance, power level, and activity.
- **Occupied Bandwidth:** The spectral width of the transmitted signal reflects the data rate and modulation complexity.
- **Temporal Patterns:** The timing of transmission bursts, silence gaps, and periodic structures reveals protocol-level behavior.
- **Phase Characteristics:** The carrier phase behavior, including drift rate and stability, provides information about transmitter hardware characteristics and physical movement.

These features are independent of the encryption applied to the payload data. Even a perfectly encrypted signal must obey the laws of physics: more data requires more bandwidth and airtime, faster-moving transmitters exhibit Doppler effects on the carrier phase, and different operational modes produce different aggregate energy profiles.

2.3.2 IQ Sample Analysis

In-phase/Quadrature (IQ) sampling is the standard method for digitizing RF signals in SDR systems. Each sample is a complex number $z = I + jQ$, where I represents the in-phase component and Q represents the quadrature component. From these raw complex samples, all physical layer features can be derived mathematically [9]:

- Instantaneous Power: $P = |z|^2 = I^2 + Q^2$
- Amplitude: $a = |z| = \sqrt{I^2 + Q^2}$
- Phase: $\phi = \arctan\left(\frac{Q}{I}\right)$
- Frequency Content: $X(f) = \text{FFT}\{z[n]\}$ via the Fast Fourier Transform (FFT) of the IQ sample sequence

The power of IQ-based analysis lies in its completeness: the IQ representation fully captures the amplitude, phase, and frequency information of the received signal, enabling comprehensive physical layer characterization without any knowledge of the modulation scheme or protocol structure.

2.3.3 Traffic Analysis of Encrypted Communications

The concept of extracting information from encrypted traffic is well-established in the networking literature. Website fingerprinting attacks on encrypted HTTPS traffic [10], traffic classification in VPN tunnels, and application identification from encrypted flows all demonstrate that encryption protects content but not metadata. This thesis extends these principles from the network layer to the physical layer, showing that even when all protocol headers and payloads are encrypted, the RF physical layer leaks behavioral information.

2.4 Machine Learning for RF Classification

Machine learning has become the dominant approach for automated signal classification and RF fingerprinting. The task involves training models to recognize patterns in engineered feature sets extracted from RF signals.

2.4.1 Feature-Based Classification

Traditional RF classification approaches extract a set of handcrafted features from the signal and train supervised learning models on labeled datasets. Common algorithms include:

- Random Forest: An ensemble of decision trees that provides robust classification with built-in feature importance ranking. Random Forest is particularly well-suited for RF classification due to its resilience to noisy features and ability to handle mixed-type inputs [11].
- XGBoost: A gradient boosting framework that sequentially trains decision trees to correct the errors of previous trees, often achieving state-of-the-art performance on structured data classification tasks.
- Support Vector Machines (SVM): Effective for high-dimensional feature spaces, though increasingly superseded by ensemble methods.

2.4.2 Deep Learning Approaches

More recent work has applied Convolutional Neural Networks (CNNs) directly to raw IQ samples or spectrograms, bypassing the feature engineering step entirely. While these approaches can discover features that human engineers might miss, they typically require larger datasets and more computational resources, and offer less interpretability than feature-based methods [12].

2.4.3 RF Fingerprinting

RF fingerprinting exploits hardware-specific imperfections — such as oscillator drift, IQ imbalance, and power amplifier nonlinearities — to identify individual transmitters. Each radio device has a unique RF fingerprint caused by manufacturing variations, enabling device identification even when using identical protocols and encryption. This thesis’s phase fingerprint analysis contributes to this field by demonstrating that OcuSync’s carrier frequency offset (caused by the drone’s crystal oscillator) creates a detectable, device-specific signature [13].

2.5 Drone Detection and Classification Research

Several research groups have investigated drone detection and classification using RF analysis:

- Nguyen et al. (2016) demonstrated drone detection and identification using WiFi-based drones, achieving high accuracy with relatively simple features. However, their work focused on unencrypted WiFi protocols, which are significantly easier to analyze than OcuSync [14].
- Al-Sa’id et al. (2019) proposed RF-based drone classification using spectrograms and deep learning, achieving 99% detection accuracy. Their approach required high-resolution spectral analysis and computationally intensive neural networks [15].
- Ezuma et al. (2020) developed a drone RF analysis framework using both micro-Doppler and spectral features, demonstrating the feasibility of multi-feature approaches for drone classification in cluttered environments [16].
- Proto17/dji_droneid published open-source research on reverse-engineering DJI’s DroneID protocol, which broadcasts unencrypted identification information. While DroneID provides identification data, it does not reveal operational states, which is the gap this thesis addresses [3].
- Bender (2022) and Andonov (2025) contributed to the understanding of OcuSync’s channel structure, frequency hopping behavior, and TDD timing, which informed the channel configuration used in this thesis’s ML-enhanced jamming system [4].
- Shi, Shan, and Dong (2026) proposed an integrated RF analysis framework combining STFT-based spectrograms, morphological image processing, and a 1D-CNN to simulta-

neously detect drone presence and decode communication parameters — including FHSS hop patterns and OFDM symbol timing — even at $\text{SNR} = -10 \text{ dB}$ [17]. Their framework was evaluated on the DroneRFa public dataset (24 drone types, three ISM bands) achieving 98.5% classification accuracy, and validated in complex urban outdoor environments at drone-to-receiver distances up to 1.5 km. Critically, their spectrogram analysis (Fig. 3 of that paper) demonstrates that different DJI models — including the MATRICE 100, Matrice 300, Inspire 2, Mini 3 Pro, Air 2S, and Mavic Pro — each produce a visually distinct time-frequency signature: variations in hop trajectories, dwell times, and OFDM burst widths are clearly visible even before any machine learning is applied. This confirms that the physical layer structure of OcuSync and related DJI protocols is inherently discriminative across drone models, which motivates the TDD frame occupancy analysis performed in this thesis (Section 4.2.4). Where their approach requires access to raw IQ data and STFT computation, our fingerprint-based approach achieves a comparable visualisation of OcuSync’s burst structure from pre-computed 128-bin CSV features, making it suitable for real-time embedded deployment.

2.6 RF Jamming and Anti-Jamming Techniques

RF jamming — the deliberate transmission of interference to disrupt wireless communication is one of the oldest and most fundamental electronic warfare techniques. In the context of drone security:

- Broadband noise jamming floods the target frequency band with Gaussian white noise, degrading the signal-to-noise ratio below the receiver’s demodulation threshold [18].
- Reactive jamming monitors the spectrum and transmits interference only when target activity is detected, improving energy efficiency.
- Adaptive frequency-hopping countermeasures attempt to follow the target’s frequency hops, requiring spectrum awareness and prediction capabilities.

Modern drone protocols like OcuSync implement several anti-jamming mechanisms, including adaptive frequency hopping, redundant coding, and asymmetric error correction on safety-critical control channels. This thesis’s jamming experiments provide empirical characterization of these defenses.

2.7 GPS Spoofing Attacks on UAVs

Global Positioning System (GPS) spoofing attacks aim to deceive GPS receivers by transmitting fabricated satellite signals that are stronger than the authentic signals. GPS operates on the L1

frequency of 1575.42 MHz, and civilian GPS signals lack cryptographic authentication, making them inherently vulnerable to spoofing attacks [19].

Several notable GPS spoofing attacks against drones have been demonstrated:

- Humphreys et al. (2008) at the University of Texas conducted some of the earliest publicly demonstrated GPS spoofing attacks, successfully redirecting a UAV's navigation system using a portable GPS simulator [20].
- Kerns et al. (2014) demonstrated a sophisticated spoofing attack that gradually aligned with the authentic GPS signal before taking over, making detection extremely difficult [21].

GPS-SDR-SIM, the open-source tool used in this thesis, enables the generation of GPS L1 baseband signals from specified trajectories and broadcast ephemeris data, which can then be transmitted through an SDR [22]. Modern consumer devices implement varying levels of anti-spoofing protection, from basic signal consistency checks to sophisticated multi-source location fusion.

2.8 Replay Attacks on Wireless Systems

Replay attacks involve capturing legitimate wireless transmissions and retransmitting them with the intent to impersonate or control the target system. While effective against simple, unencrypted protocols, modern drone communication systems implement multiple defenses against replay attacks [23]:

- Session-unique encryption keys: Each power cycle generates new keys, invalidating captured transmissions.
- Sequence numbers: Each packet contains a unique, monotonically increasing identifier.
- Timestamp validation: Packets include expiration times that prevent delayed replay.
- Challenge-response authentication: The initial pairing process establishes shared secrets that cannot be extracted from captured traffic.

2.9 Summary and Research Gap

While significant research exists on drone detection using unencrypted protocols and deep learning on spectral data, there is a notable gap in the literature regarding:

- Comprehensive physical layer feature engineering for encrypted drone traffic that goes beyond simple power and spectral features to include phase analysis, temporal fingerprints,

and inter-window dynamics.

- Systematic evaluation of multiple attack vectors (jamming, spoofing, replay) against a single drone platform under controlled laboratory conditions.
- Real flight validation of laboratory-developed classification models, including honest assessment of limitations.

This thesis addresses these gaps by developing a 46-feature characterization framework, systematically testing multiple attack scenarios, and validating results through real flight experiments.

3 System Architecture

3.1 Hardware Setup

3.1.1 USRP B210 Software Defined Radio

The primary instrument used throughout this research is the Ettus Research USRP B210 (serial: 321E87F), a full-duplex, two-channel Software Defined Radio connected via USB 3.0. The B210 was selected for its ability to operate in the 5.8 and 2.4 GHz bands (required for OcuSync analysis), its simultaneous transmit/receive capability (essential for the jamming experiments), and its compatibility with the UHD driver library and GNU Radio framework.

Parameter	Value
Frequency Range	70 MHz – 6 GHz
Instantaneous Bandwidth	Up to 56 MHz
ADC/DAC Resolution	12-bit
RX Channels	2 (independent)
TX Channels	2 (independent)
Full Duplex	Yes
Interface	USB 3.0
FPGA	Xilinx Spartan 6

Table 3.1: System Specifications

For RF capture experiments, the USRP was configured with the following parameters:

- Center Frequency: 5.845 GHz (targeting DJI OcuSync 5.8 GHz band)
- Sample Rate: 10 MHz (10,000,000 samples per second)
- Gain: 60 dB (fixed, AGC disabled to preserve power consistency)
- Antenna Ports: RX2 for reception, TX/RX for transmission during jamming

Sample rate selection — empirical justification from IQ burst analysis

The 10 MHz sample rate was not chosen arbitrarily or solely from the protocol specification. It was validated empirically by analysing the raw IQ recordings in Python and inspecting the temporal structure of OcuSync bursts at the sample level.

At 10 MHz, each complex sample spans exactly 0.1 μ s. A 10 ms capture window therefore contains 100,000 samples. When these samples are divided into 128 equal time bins (781 samples per bin $\approx$ 78 μ s per bin), a single OcuSync burst — which has a measured duration of approximately 50 μ s — falls almost entirely within one bin. This is the critical balance point:

Fingerprint bins	Samples per bin	Time per bin	Effect on 50 μ s bursts
64	1,563	156 μ s	Merges adjacent bursts — two successive bursts collapse into one bin, losing resolution
128	781	78 μs	Optimal — each bin is just wide enough to contain one burst without overlapping the next
256	390	39 μ s	Splits bursts — a single burst spans two bins; each bin averages only $\sim$ 195 active samples, increasing noise sensitivity

Table 3.2: Effect of fingerprint bin size on 50 μ s bursts

The same logic applies to the sample rate itself. A lower rate (e.g., 5 MHz, 0.2 μ s per sample) would compress a 50 μ s burst into only 250 samples, reducing amplitude resolution and making it harder to distinguish burst onset from inter-burst gaps. A higher rate (e.g., 20 MHz, 0.05 μ s per sample) would double the instantaneous bandwidth, pulling in noise and interference from adjacent channels outside the 10 MHz OcuSync channel, while providing no additional information about the drone’s bursts (each burst would still occupy the same 50 μ s; it would just be over-sampled). At 10 MHz the received bandwidth exactly matches the OcuSync channel width: every sample belongs to the drone’s channel, and the burst timing is resolved without either merging or splitting individual packets.

3.1.2 Target Drones

Three DJI drones were used throughout the experimental campaign:

DJI MAVIC 2 PRO — Used primarily in simulation mode for initial feature development and algorithm validation. The Mavic 2 employs OcuSync 2.0, which operates on both 2.4 GHz and 5.8 GHz bands with adaptive frequency hopping across approximately 10 channels in the 5.725–5.850 GHz range.

DJI Matrice 300 RTK (M300 RTK) — Used for the laboratory stationary experiments and real flight testing. The M300 employs OcuSync Enterprise, which provides higher transmit power (up to 26 dBm EIRP), longer range, and potentially different TDD timing compared to the consumer OcuSync 2.0. The M300 RTK was chosen for its professional-grade capabilities and its suitability for controlled outdoor experiments.

DJI Mini — Used in the GPS spoofing experiments. The Mini is a lightweight consumer drone that proved susceptible to GPS location manipulation when appropriate spoofing parameters were applied.

3.1.3 Antenna Configuration

For 5.8 GHz experiments, omnidirectional antennas rated for the 5.725–5.850 GHz band were connected to the USRP B210’s RX2 and TX/RX ports. For GPS spoofing experiments targeting the 1575.42 MHz L1 band, a separate GPS-compatible antenna was used on the TX port.

The receiving antenna was positioned approximately 1–2 meters from the drone controller during laboratory experiments. This proximity ensured strong signal reception but introduced near-field effects that are discussed in the limitations section.

3.1.4 Laboratory Setup

The laboratory environment consisted of an indoor workspace where the USRP B210 was positioned on a desk alongside the drone controller. For stationary experiments, the drones were powered on with propellers removed (or in simulation mode) to allow extended data collection without flight. For jamming experiments, the jammer (USRP TX) and monitor (USRP RX) operated simultaneously using the B210’s full-duplex capability.

3.2 Software Architecture

3.2.1 System Overview

The complete software pipeline is organized into four major subsystems, as illustrated in Figure 3.1:

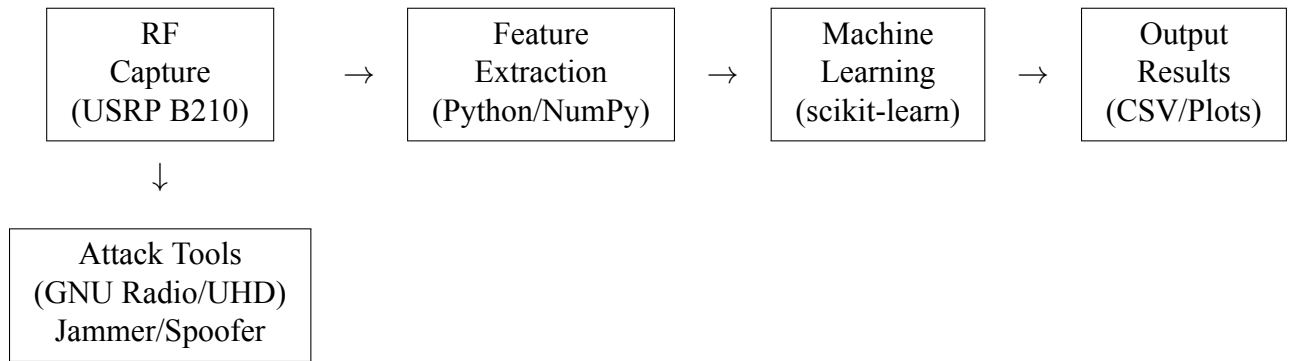


Figure 3.1: RF signal processing and attack pipeline

Figure 3.1: System architecture overview showing the data flow from RF capture through feature extraction and ML classification, with parallel attack tool capabilities.

3.2.2 Capture Pipeline

The core capture pipeline is implemented in Python using the UHD (USRP Hardware Driver) library for direct hardware control. The primary capture scripts include:

- ‘iq_size_capture_v3.py’ — The most advanced capture script, implementing real-time IQ acquisition, 46-feature extraction, and 128-bin fingerprint generation in a continuous

pipeline. Each capture window consists of 100,000 complex IQ samples (10 ms at 10 MHz), from which all features are computed before the next window begins.

- ‘rf_capture_simple.py’ — An earlier, simplified capture script that extracts basic power and bandwidth statistics for initial experiments.
- ‘simple_recorder.py’ — A user-friendly recording script that allows the operator to specify the drone state label and recording duration, automatically organizing captured data into labeled sessions.

3.2.3 Attack Tools

Three categories of attack tools were developed:

1. RF Jammer (‘dji_jammer_5_8ghz.py’) — A GNU Radio-based broadband noise jammer targeting the 5.8 GHz band. The jammer generates Gaussian white noise with configurable transmit gain (0–89 dB), noise amplitude (0–1.0), and center frequency offset. The flowgraph was designed in GNU Radio Companion and compiled to Python for command-line operation.

2. ML-Enhanced Adaptive Jammer (‘reactive_jammer.py’) — An advanced jamming system that combines spectrum scanning, channel tracking, and machine learning prediction to defeat OcuSync’s adaptive frequency hopping. The system architecture consists of four components:

- Spectrum Scanner: Wideband USRP receiver monitoring all 10 OcuSync channels simultaneously
- Channel Tracker: Detects the active channel and logs hop events to CSV
- ML Predictor: Random Forest model trained on hop history to predict the next escape channel
- Reactive Jammer: Transmits noise on the predicted channel before the drone arrives

3. GPS Spoofers (‘gps-sdr-sim-uhd.py’) — A Python script interfacing GPS-SDR-SIM with the USRP B210 to transmit fabricated GPS L1 signals at 1575.42 MHz. The system accepts trajectory files in CSV format and broadcast ephemeris files to generate physically plausible satellite signals.

3.3 Data Collection Methodology

3.3.1 Capture Window Design

Each data capture session consists of continuous 10 ms windows, each containing exactly 100,000 complex IQ samples at the 10 MHz sample rate. The 10 ms window duration was chosen because:

- Protocol alignment: OcuSync’s TDD frame period is approximately 10 ms, so each window captures a complete transmission cycle.
- Statistical sufficiency: 100,000 samples provide statistically robust estimates for all computed features.
- Temporal resolution: 10 ms provides 100 classification decisions per second, sufficient for real-time behavioral tracking.

Each complex IQ sample $z_k = I_k + jQ_k$ is stored as a complex64 value (two 32-bit floats: 4 bytes for I + 4 bytes for Q = 8 bytes per sample). A raw 10 ms window therefore occupies 800 KB. However, rather than storing raw samples, the capture pipeline computes and stores only the 46 scalar features and 2×128 fingerprint values, reducing the per-window storage to approximately 5 KB in CSV format.

3.3.2 State Labeling Protocol

Data collection followed a single-state-per-recording protocol: each recording session captures only one labeled behavioral state. The operator specifies the state label at the start of each recording, eliminating the need for real-time labeling during complex flight operations. The following states were used across different experiments:

Table 3.3: Drone State Labels and Descriptions

3.3.3 Datasets

Four primary datasets were collected:

1. Simulation Dataset — Captured from DJI Mavic 2 in simulation mode. As simulation mode generates a fixed ”dummy” video stream regardless of simulated flight state, this dataset yielded only 33.3% classification accuracy (equivalent to random guessing for 3 classes), confirming that simulation does not replicate the RF behavioral differences present in real drone operation.
2. Lab Stationary (M300 RTK) — Captured on March 3, 2026, with the DJI Matrice 300 RTK at approximately 1 meter distance. Three classes: no drone, drone still (no camera movement), and drone with camera movement. This dataset yielded the primary 96% classification accuracy result.
3. BiggerDrone Dataset — Focused binary classification (camera move vs. no camera move) with the M300 RTK, yielding 95.2% accuracy with a training set of 1,200 samples.
4. Real Flight Dataset — Captured during an outdoor flight of the M300 RTK, recording RF signals as the drone flew away from the controller without camera activity. Used primarily for path-loss distance estimation analysis.

State Label	Description	Dataset
no_drone	No drone present — pure ambient RF environment	All datasets
idle	Drone powered on, motors off, no camera movement	Simulation, Lab
idle_no_camera_move	M300 RTK on, no gimbal movement	Lab stationary
idle_camera_move	M300 RTK on, operator moving gimbal	Lab stationary
hover	Drone hovering in place	Simulation
fly_fast	Drone flying at high speed	Simulation
no_camera_movement	Stationary drone: drone powered on but no movement of camera	Stationary test
with_camera_movement	Stationary drone: drone powered on but with camera movement	Stationary test
fly_away_nocamera	Real flight: drone flying away, no camera active	Real flight
real_flight_camera	Real flight: drone flying away and coming back, camera active	Real flight
take_off_camera	Real flight: drone taking off, camera active	Real flight
intheair	Real flight: drone flying in the air, camera active	Real flight

Table 3.3: Drone operational states and corresponding datasets

3.4 Feature Engineering Framework

The core technical contribution of this thesis is a comprehensive 46-feature framework for characterizing drone RF traffic from raw IQ samples. The features are organized into three progressive layers (V1, V2, V3), each building on the previous layer’s capabilities.

3.4.1 V1: Energy and Activity Features (12 Variables)

The V1 feature layer answers the fundamental question: ”How much RF energy is present? Is a drone transmitting?”

All V1 features are computed from the instantaneous power of each sample:

$$P_k = |z_k|^2 = I_k^2 + Q_k^2$$

Table 3.4: V1 Energy/Activity Features

The noise floor threshold of -60 dB was empirically determined from the USRP B210’s thermal noise level at the configured gain of 60 dB. ‘active_ratio’ emerged as the single most reliable

#	Feature	Formula	Purpose
1	total_energy	$\sum z_k ^2$	Total power across window
2	mean_power_db	$10 \cdot \log_{10}(\text{mean}(z_k ^2))$	Average power in dB
3	max_power_db	$10 \cdot \log_{10}(\max(z_k ^2))$	Peak burst power
4	active_samples	$\text{count}(P_k > -60 \text{ dB})$	Samples above noise floor
5	active_ratio	$\frac{\text{active_samples}}{N}$	Fraction of window with signal
6	active_energy	$\sum z_k ^2$ (active only)	Signal energy (no noise)
7	active_mean_db	$10 \cdot \log_{10}(\text{mean}(\text{active } P_k))$	Average power when transmitting
8	burst_count	Transitions below $\rightarrow$ above -60 dB	Number of transmission bursts
9	duty_cycle	Same as active_ratio	RF engineering convention
10	power_std_db	$\text{std}(10 \cdot \log_{10}(z_k ^2))$	Power variability
11	bandwidth_hz	3 dB bandwidth via FFT	Occupied spectral width
12	effective_size_bytes	$\text{active_samples} \times 8$	Virtual data volume

Table 3.4: V1 Energy/Activity Features

drone detection feature, jumping from approximately 1.5% (no drone) to 72% (drone active).

For the ‘bandwidth_hz’ computation FFT is performed. The power spectrum is computed as $|\text{FFT}(z)|^2$, and the -3 dB bandwidth is estimated by counting frequency bins within half of the peak spectral power.

3.4.2 V2: Amplitude and Phase Statistics (17 Scalar + 10 Histogram + 2×128 Fingerprint)

The V2 layer answers: What does the signal’s shape and modulation look like?

From each complex sample, two real values are extracted:

- Amplitude: $a_k = |z_k| = \sqrt{I_k^2 + Q_k^2}$
- Phase: $\phi_k = \text{atan2}(Q_k, I_k)$

Amplitude Statistics (Features 13–22): Seven statistical measures of the amplitude distribution — mean, standard deviation, maximum, minimum, median, 95th percentile, and 5th percentile — plus three derived metrics: dynamic range (P95/P05 ratio), crest factor (peak/RMS ratio), and RMS amplitude.

The ‘amp_mean’ feature serves as the primary distance indicator. By the inverse-square law, received power $P \propto 1/d^2$, so as the drone flies further away, ‘amp_mean’ decreases predictably.

In laboratory data, the no-drone condition showed $\text{amp_mean} = 0.000434$, while the drone at approximately 1 meter exhibited $\text{amp_mean} = 0.005225$, representing a $12\times$ signal-above-noise ratio (+21.6 dB).

Phase Statistics (Features 23–29): The phase analysis captures carrier behavior through seven features: linear phase mean, phase standard deviation, circular mean (handling the $-\pi/\pi$ wrap-around correctly), Von Mises concentration parameter, phase drift (linear regression slope of unwrapped phase), phase entropy (Shannon entropy across 36 angular bins), and amplitude-phase cross-correlation.

The ‘phase_drift’ feature proved to be the #1 most important feature in ML training (10.9% importance). This metric represents the rate of carrier phase rotation, which is determined by the frequency offset between the drone’s oscillator and the USRP’s tuning frequency. Camera/gimbal movement causes $2.2\times$ more variability in phase drift compared to a still drone, because the gimbal motors introduce physical vibrations that modulate the carrier phase.

Amplitude Histogram (Features 30–39): Ten histogram bins spanning the amplitude range $[0, \text{amp_max}]$, each containing the normalized fraction of samples in that amplitude range. This provides a compact discretized probability density function of the signal strength distribution.

Fingerprint Vectors (Features 40–41): Two 128-element vectors that map the temporal burst structure within each 10 ms window:

The 100,000 samples are divided into 128 equal bins of 781 samples each, covering approximately $78\ \mu\text{s}$ per bin. The choice of 128 bins balances three factors:

1. Resolution: OcuSync transmission bursts have approximately $50\ \mu\text{s}$ periodicity. At $78\ \mu\text{s}$ per bin, individual burst cycles can be resolved.
2. Statistical stability: 781 samples per bin provides a stable mean, reducing per-bin noise.
3. Computational efficiency: 128 is a power of 2.

The amplitude fingerprint records the average amplitude during each $78\ \mu\text{s}$ time slice, revealing the OcuSync TDD schedule — when the drone actively transmits (high values) versus silence gaps (values near the noise floor of 0.0004). The phase fingerprint records the average unwrapped phase in each time slice, producing a generally increasing curve whose slope reflects the drone’s carrier frequency offset.

3.4.3 V3: Inter-Window Features (7 Variables)

The V3 layer answers: “Is the drone moving, approaching, or departing?”

These features compare the current 10 ms window to the previous window and rolling history:

The ‘fp_cosine_sim’ (cosine similarity between consecutive fingerprint vectors) proved particularly revealing:

#	Feature	Description
42	amp_mean_delta	Amplitude change from previous window
43	relative_distance	Distance ratio via inverse-square law
44	fp_cosine_sim	Fingerprint similarity between consecutive windows
45	phase_drift_delta	Change in phase rotation rate (acceleration indicator)
46	amp_trend_short	Linear regression slope of amp_mean over 5 windows (50 ms)
47	amp_trend_long	Linear regression slope over 20 windows (200 ms)
48	amp_ema	Exponential Moving Average ($\alpha = 0.2$) of amplitude

Table 3.5: V3 Inter-Window Features

- No drone: 0.999 (every window’s noise looks identical)
- Drone still: 0.745 (OcuSync reshuffles TDD time slots adaptively, 77% of consecutive pairs below 0.8)
- Camera moving: 0.758 (similar instability, but with different amplitude values within bursts)

3.4.4 The Envelope Analogy

A unifying conceptual framework for understanding why these features work on encrypted data is the envelope analogy:

We don’t read the letters inside the envelope — we just weigh the envelope.

Each physical layer feature corresponds to measuring a property of the communication envelope without opening it:

- Power → How heavy is the envelope?
- Bandwidth → How wide is the envelope?
- Frequency offset → What color is the envelope?
- Phase entropy → How wrinkled is the envelope?
- Active ratio → How often are envelopes being delivered?
- Fingerprint → What is the delivery schedule pattern?

Encryption scrambles the letter inside the envelope but does not affect the envelope’s physical properties. A drone streaming live 4K video sends heavy envelopes at high frequency, while an

idle drone sends light envelopes infrequently. These differences are measurable and classifiable regardless of encryption.

3.5 Machine Learning Pipeline

3.5.1 Model Selection

The primary classifier used throughout this thesis is the Random Forest algorithm, selected for the following reasons:

- **Robustness:** Random Forest handles noisy features and mixed-type inputs effectively, important given the heterogeneous nature of RF features spanning different physical domains.
- **Feature importance:** The built-in feature importance metric (mean decrease in impurity) provides interpretable insights into which physical phenomena drive the classification.
- **Low overfitting risk:** The ensemble averaging across hundreds of trees reduces overfitting compared to single-tree models.
- **Preprocessing independence:** While Random Forest is theoretically invariant to monotonic feature scaling, ‘StandardScaler’ normalization was applied uniformly across all models in the pipeline to ensure fair comparison with scaling-sensitive algorithms (XGBoost, MLP). The scaler is fit on the training set only and applied identically to the validation and test sets, preventing data leakage.

XGBoost was used as an alternative model for comparison, offering gradient boosting’s typically superior performance at the cost of additional hyperparameter tuning.

Model	Test Accuracy	Precision (weighted)	Recall (weighted)	F1 (weighted)
Random Forest	95.26%	0.953	0.953	0.953
XGBoost	95.85%	0.959	0.959	0.959
Gradient Boosting	95.26%	0.953	0.953	0.953

Table 3.6: Model Comparison — Three-Class Lab Classification (M300 RTK, 70/15/15 stratified split, 675 test samples)

All three models perform within 0.6% of each other, demonstrating that the 46-feature set provides strong discriminative structure regardless of the classifier. Random Forest was selected as the primary model due to its interpretable feature importances, fast inference time, and near-identical accuracy to XGBoost — without requiring the additional hyperparameter sensitivity that gradient boosting methods introduce. The original deployment model (trained on a larger

80/20 split of the same dataset) achieved 96.0% test accuracy, consistent with the cross-validated results above.

3.5.2 Training Protocol

All models were trained using a stratified train / validation / test split, with proportional representation of each class in all three partitions. The validation set was used for model selection; the test set was held out until final evaluation. The training protocol included:

- Data loading: CSV files from labeled capture sessions are loaded and merged. Each class contributes approximately equal numbers of windows (300 per class), ensuring a balanced dataset where accuracy is a reliable metric.
- Feature selection: The 46 scalar features are extracted as the input matrix X. Fingerprint strings are excluded from direct ML input (their information is captured through ‘fp_cosine_sim’ and the histogram features).
- Label encoding: State labels are encoded as integer class indices.
- Feature scaling: StandardScaler (zero mean, unit variance) is fit on the training set and applied to validation and test sets.
- Model training: Random Forest with 100 trees, ‘max_depth=20’, and ‘min_samples_split=5’ — lightly constrained to reduce memory usage while maintaining full expressive capacity on this feature set. The training accuracy was 99%, and the test accuracy was 96%, indicating a modest generalization gap consistent with slightly over-expressive trees that is controlled by the ensemble averaging effect.
- Evaluation: Accuracy, confusion matrix, classification report (precision, recall, F1), and feature importance ranking are computed on the held-out test set.

4 Experimental Results

This chapter presents the results of six experimental campaigns: simulation mode baseline, stationary laboratory classification, camera movement detection, RF jamming, GPS spoofing, replay attack analysis, and real flight validation.

4.1 Simulation Mode Baseline

4.1.1 Experiment Design

The initial experiments were conducted using the DJI Mavic 2 Pro in simulation mode, which allows the drone’s software to simulate flight behavior without physical movement. Three operational states were recorded: idle (motors off), hover (simulated hovering), and fly_fast (simulated fast flight). RF signals were captured at 5.8 GHz using the standard capture pipeline.

4.1.2 Simulation mode Results

The Random Forest classifier trained on simulation data achieved only 33.3% accuracy — equivalent to random guessing for a three-class problem.

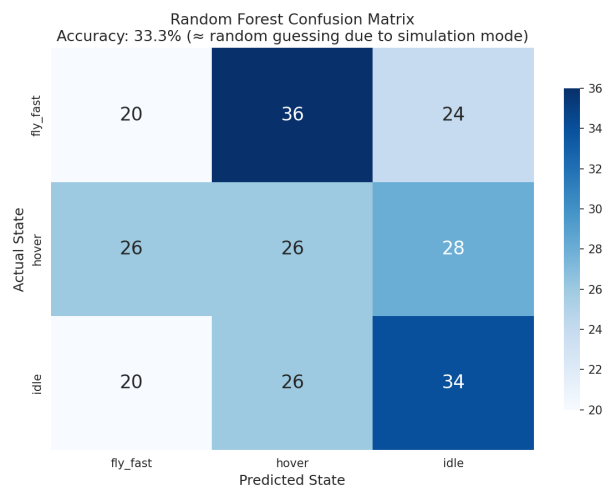


Figure 4.1: Confusion matrix

Figure 4.1: Confusion Matrix — Simulation Mode. The matrix shows near-uniform distribution across all predicted classes for all true classes, confirming that the classifier cannot distinguish between simulated flight states. Each cell contains approximately 20–36 samples with no discernible diagonal dominance.

The feature importance analysis from this experiment revealed that power-based features ('min_power_db', 'spectral_entropy', 'max_power_db') dominated the ranking, but all had essentially equal importance, indicating no discriminative structure in the data.

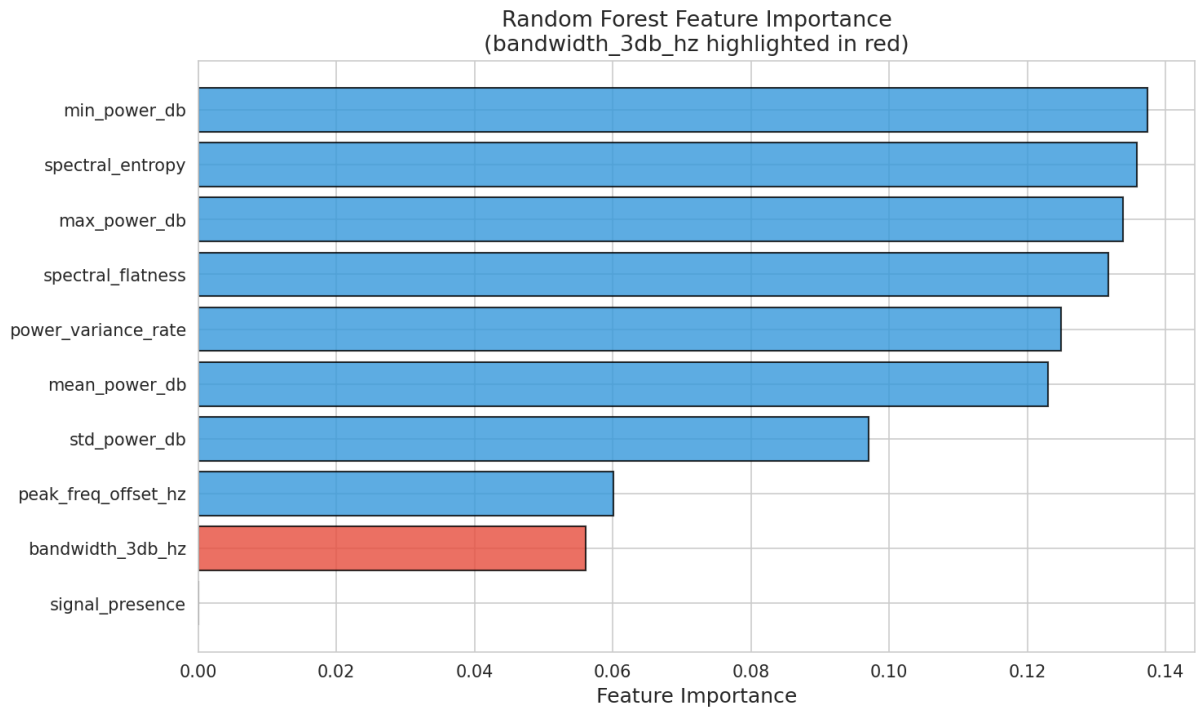


Figure 4.2: Feature Importance

Figure 4.2: Feature Importance — Simulation Mode. The horizontal bar chart shows that 'min_power_db' (13.9%), 'spectral_entropy' (13.5%), and 'max_power_db' (13.1%) are the top features, but their importance values are nearly identical and provide no meaningful classification signal

4.1.3 Analysis and Interpretation

This negative result is not a failure of the methodology — it is a validation of the approach. In simulation mode, the DJI Mavic 2 generates a fixed dummy video stream at a constant data rate regardless of the simulated flight state. Since the RF physical layer features measure how much data the drone transmits and in what pattern, and simulation mode transmits the same data pattern for all states, no behavioral difference is detectable.

Figure 4.3: Feature Distributions — Simulation Mode. Violin plots of key features across the three simulated states (idle, hover, fly_fast). The distributions overlap almost entirely for every feature, visually confirming the absence of discriminative structure. Unlike the real-drone data (Section 4.2), where drone-present features produce clearly bimodal distributions, simulation mode produces effectively identical distributions regardless of the commanded state.

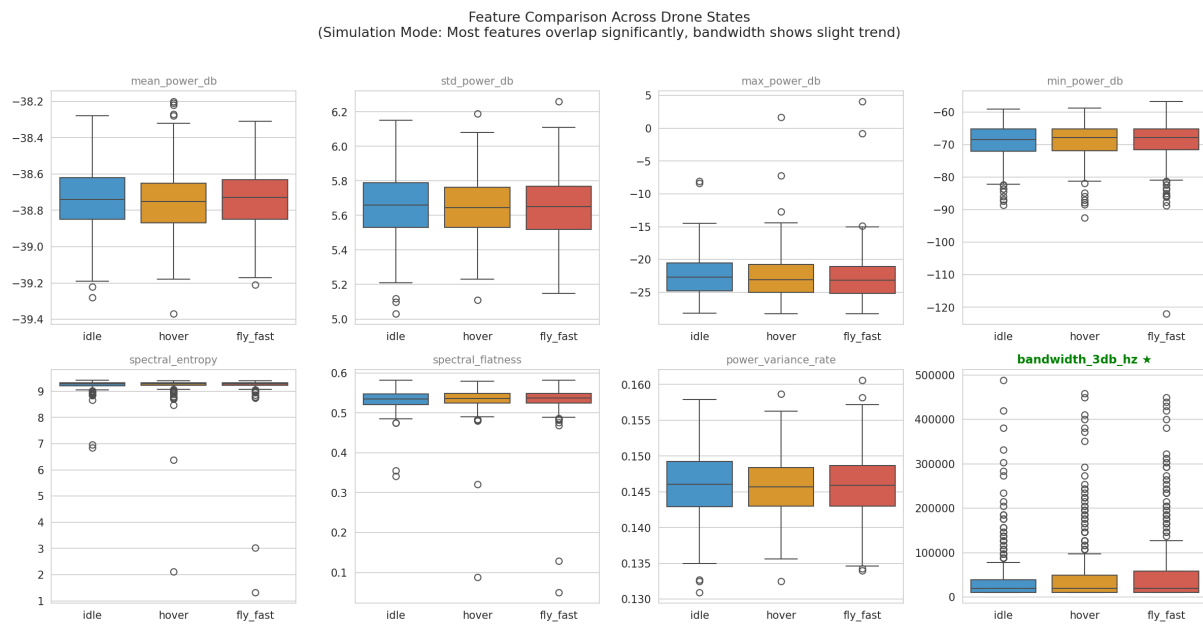


Figure 4.3: Feature Comparison

DJI Mavic 2 Drone State RF Analysis (5.8 GHz OcuSync 2.0)

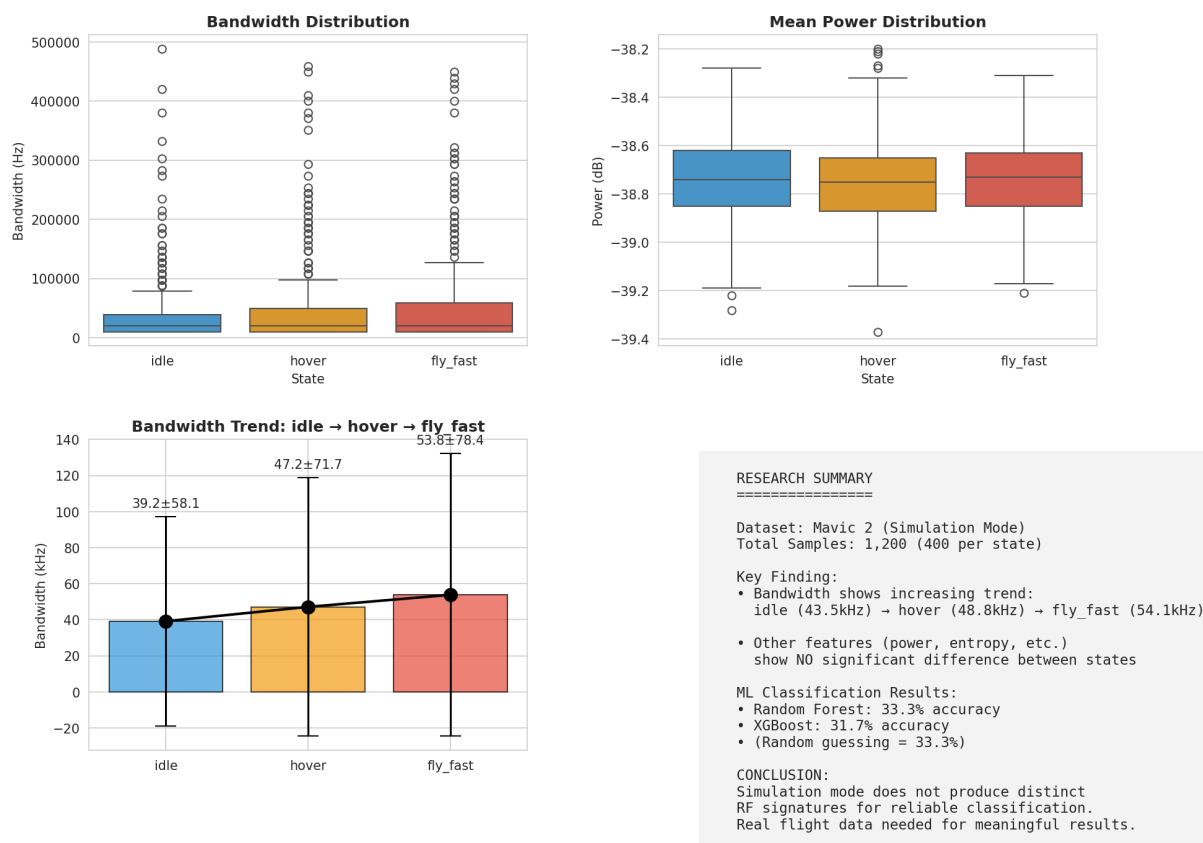


Figure 4.4: Summary Report

Figure 4.4: Simulation Analysis Summary Report. Aggregated summary statistics and classification results from the simulation mode experiment, confirming that no combination of the extracted features provides meaningful separation between simulated states.

This result confirms a critical insight: the classification does not rely on any software-level simulation artifact, but on genuine physical layer behavioral differences that only manifest during actual drone operation. This validation strengthens the credibility of the subsequent real-drone results.

4.2 Stationary Laboratory Classification (M300 RTK)

4.2.1 Experiment Design

The primary classification experiment was conducted on March 3, 2026, using the DJI Matrice 300 RTK in a laboratory environment. The USRP B210 was positioned approximately 1 meter from the drone controller. Three states were captured:

1. No drone ('no_drone'): Pure ambient RF environment with no drone powered on
2. Drone still ('idle_no_camera_move'): M300 RTK powered on, no gimbal movement
3. Camera moving ('idle_camera_move'): M300 RTK powered on, operator manually moving the gimbal

Each state was recorded for several minutes, yielding approximately 300 windows per class. Feature extraction was performed using the V3 pipeline (46 features + fingerprints).

4.2.2 Classification Results

The Random Forest classifier achieved 96% accuracy across the three classes.

Feature	No Drone	Drone Still	Drone + Camera	Conclusion
active_ratio	1.5%	72.2%	71.5%	Drone detection is trivial
amp_mean	0.000434	0.005225	0.005204	$12\times$ signal above noise (+21.6 dB)
phase_drift std	0.006	0.022	0.048	Camera causes $2.2\times$ more phase instability
amp_phase_corr std	0.004	0.009	0.019	Camera decorrelates amplitude-phase
fp_cosine_sim	0.999	0.745	0.758	Drone bursts reshape every 10 ms
phase_fp between-window std	294	1,243	3,150	Camera disrupts coherent phase ramp

Table 4.1: Key Lab Results (March 3, 2026)

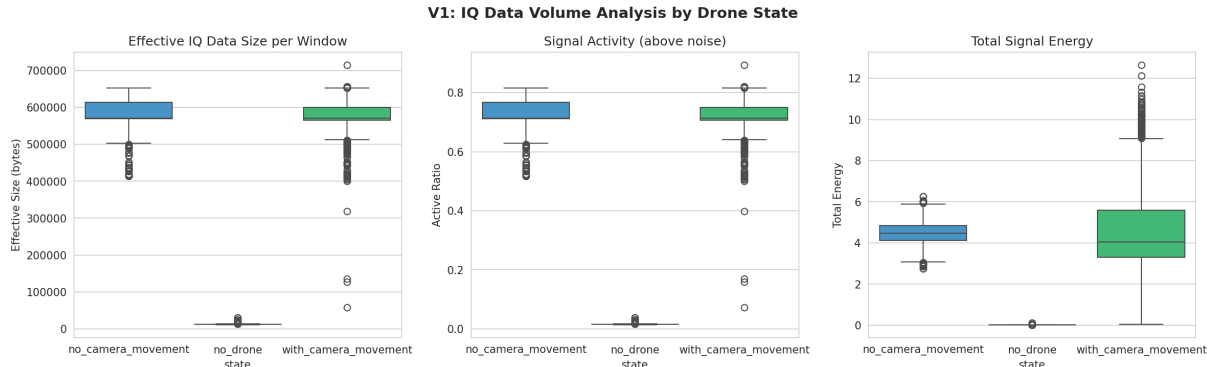


Figure 4.5: V1 Size comparison

Figure 4.5: V1 Energy/Activity Features by Drone State. Box plots of the three primary V1 features — effective IQ data size per window, signal activity (active ratio above the noise floor), and total signal energy — across the three states. The no-drone state clusters near zero for all three metrics, while both drone-present states exhibit dramatically elevated values. This confirms that simple energy-based features achieve near-perfect drone/no-drone separation.

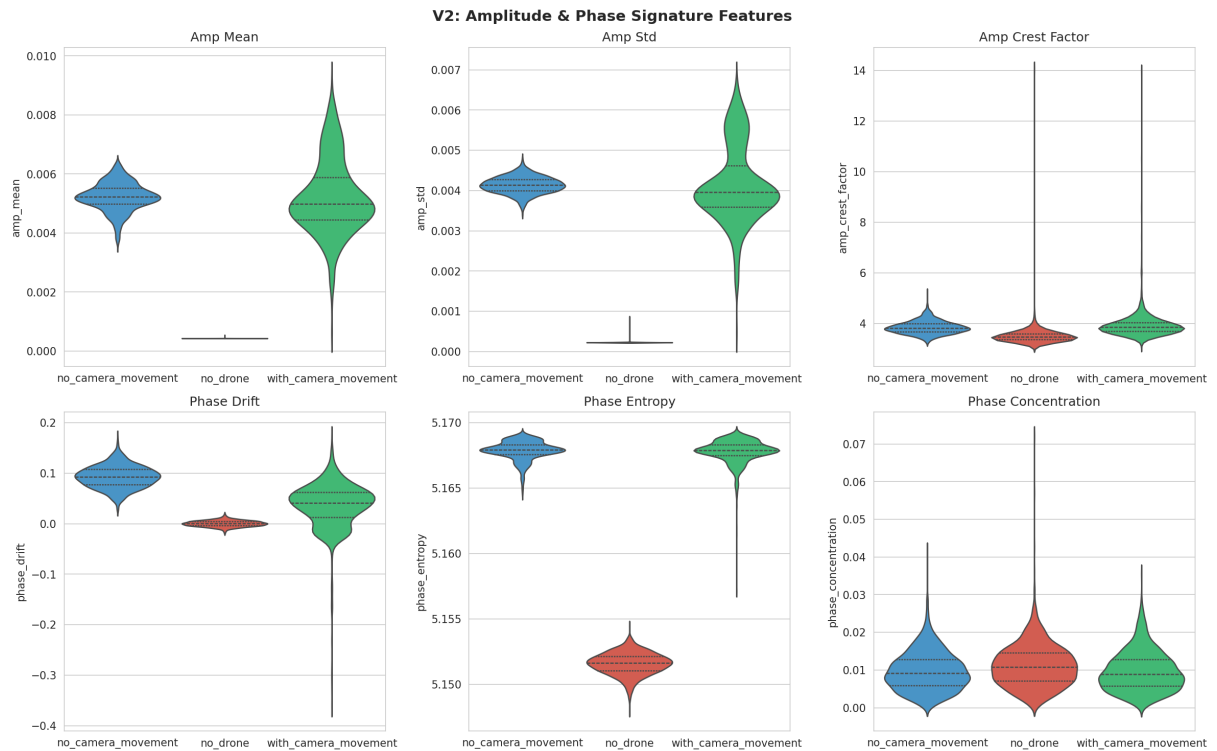


Figure 4.6: V2 Amplitude Phase Features

Figure 4.6: V2 Amplitude and Phase Signature Features. Violin plots of six V2 features across the three states. Top row: ‘amp_mean’, ‘amp_std’, and ‘amp_crest_factor’ show clear separation between no-drone and drone-present, with ‘amp_mean’ providing the 12× signal-above-noise signature. Bottom row: ‘phase_drift’ shows the widest distribution spread for the camera-moving state, confirming its role as the top ML feature. ‘phase_entropy’ cleanly separates noise (lower entropy at 5.151) from drone signal (higher entropy at 5.167–5.169) because noise has a more uniform phase distribution than modulated signals. ‘phase_concentration’ shows similar values across all states, indicating it is less discriminative individually.

Top 5 features by Random Forest importance:

1. ‘phase_drift’ — 10.9% (V2)
2. ‘amp_ema’ — 8.1% (V3)
3. ‘active_ratio’ — 6.4% (V1)
4. ‘amp_p95’ — 6.0% (V2)
5. ‘amp_phase_corr’ — 5.8% (V2)

The dominance of ‘phase_drift’ as the most important feature reveals that the classifier relies heavily on carrier phase behavior. The phase drift captures the rate of carrier frequency rotation, which is determined by the drone’s oscillator offset and is modulated by physical vibrations from the gimbal motors during camera movement. This is a subtle physical phenomenon that would be invisible to any content-level analysis.

4.2.3 Fingerprint Analysis

The fingerprint analysis provides the most visually compelling evidence of the classification capability.

Figure 4.7: Comprehensive Fingerprint Analysis. This multi-panel figure presents six different visualizations of the three drone states:

Top Row — Random Window Amplitude Fingerprints (5 windows each): Five randomly selected 128-bin amplitude fingerprints from each state are overlaid. The no-drone state (left) shows perfectly flat lines near the noise floor (0.00044), confirming the absence of any structured signal. The drone-still state (center) shows structured burst patterns with peaks reaching 0.008, but the patterns vary dramatically between windows. The camera-moving state (right) shows similarly structured but higher-amplitude patterns, with some bins reaching 0.012.

Middle Row — Amplitude Fingerprint Heatmaps (100 consecutive windows): Each heatmap plots 100 consecutive 10 ms windows (vertical axis) against the 128 time bins (horizontal axis), with colormap indicating amplitude. The no-drone heatmap is a uniform dark rectangle — pure noise with no structure. The drone-still heatmap reveals the OcuSync TDD schedule as horizontal bands of bright (transmitting) and dark (silent) regions that shuffle position between windows,

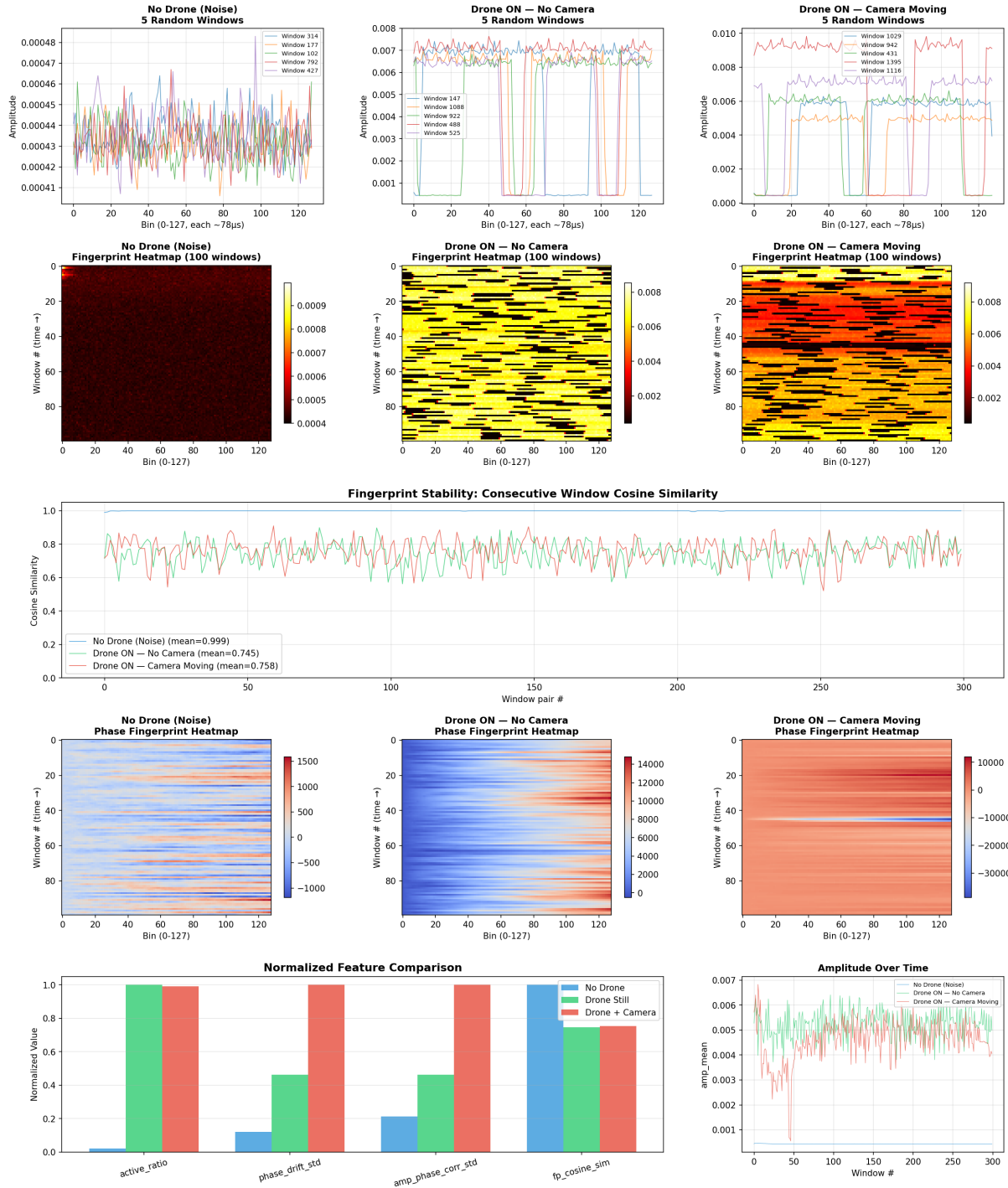


Figure 4.7: Fingerprint Analysis (M300 RTK, Lab, Stationary)

demonstrating the protocol’s dynamic time slot allocation. The camera-moving heatmap shows brighter overall coloration (higher amplitude values), confirming that camera movement forces the drone to transmit more energy per window.

Center — Fingerprint Stability (Cosine Similarity): This time series plot shows the window-to-window cosine similarity across 300 consecutive window pairs. The no-drone signal (green)

maintains near-perfect stability at 0.999 — every noise window looks identical to the previous one. The drone-still (blue, mean=0.745) and camera-moving (red, mean=0.758) signals show dramatic instability, with cosine similarity dropping below 0.5 in many consecutive pairs. This demonstrates that OcuSync dynamically reshuffles its TDD time slot allocation approximately every 10 ms, creating a constantly changing burst pattern that is characteristic of active drone communication.

Lower Row — Phase Fingerprint Heatmaps: The phase fingerprints show the cumulative unwrapped phase across the 128 bins. No-drone phase (left) is uniform since noise has random, non-accumulating phase. Drone-still phase (center) shows strong coherent ramps with consistent slope (78 radians per bin), representing the drone’s carrier frequency offset from the USRP’s tuning — essentially the oscillator signature of the specific drone. Camera-moving phase (right) shows wildly disrupted ramps with between-window standard deviation of 3,150 (compared to 1,243 for still), because gimbal motor vibrations modulate the carrier phase unpredictably.

Bottom Left — Normalized Feature Comparison: Bar chart comparing three key normalized features across all three states. The ‘active_ratio’ clearly separates no-drone from drone-present, while ‘phase_drift_std’ and ‘fp_cosine_sim’ provide the separation between still and camera-moving states.

Bottom Right — Amplitude Over Time: Raw amplitude time series for 300 consecutive windows, showing the stable noise baseline, the elevated drone-present level, and the subtle amplitude modulation caused by camera movement.

This comprehensive fingerprint visualization demonstrates that the classification is grounded in genuine physical phenomena — the OcuSync protocol’s TDD structure, dynamic slot allocation, and the physical effects of gimbal motor operation on carrier phase — rather than statistical artifacts.

4.2.4 TDD Structure Observed from Raw IQ Power

Beyond the fingerprint-level analysis, the TDD (Time-Division Duplex) structure of OcuSync is directly observable at sub-millisecond resolution in the raw IQ power data. By computing mean power in 0.5 ms resolution bins across a 100 ms observation window, the alternating pattern of strong downlink bursts (drone → controller) and weak uplink slots (controller → drone) becomes visually evident.

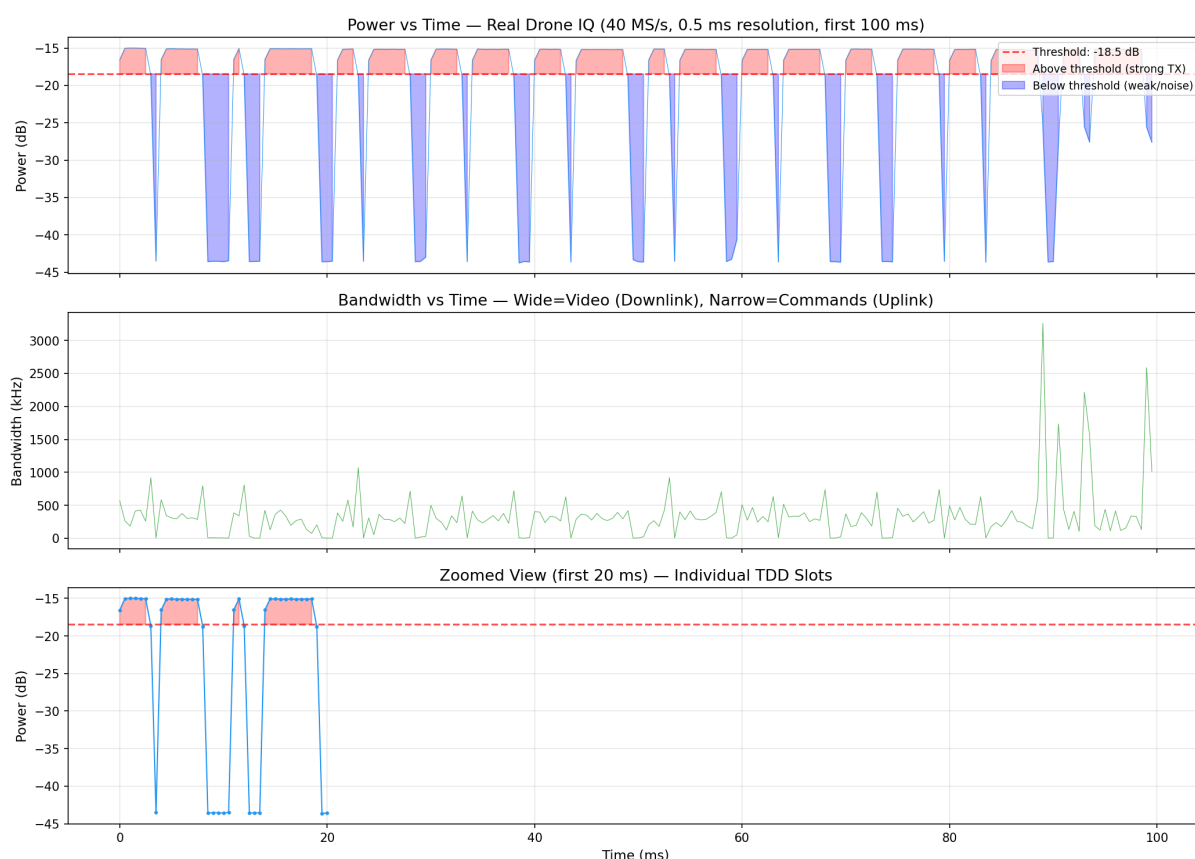


Figure 4.8: TDD Power Analysis

Figure 4.8: TDD Power Pattern — 0.5 ms Resolution. Source: ‘drone_present_20260424_141455.iq’ (M300 RTK, 40 MS/s, 5.7875 GHz centre). Top panel: Power vs. time at 0.5 ms resolution over 100 ms, with a dashed threshold at -18.5 dB ($\text{mean} + 3 \text{ dB} = -21.5 + 3$) separating strong transmissions (red/above) from weak/noise periods (blue/below). High-power downlink bursts reach -15.0 dB; noise floor sits near -43.8 dB, giving a dynamic range of approximately 29 dB. 138 of 200 chunks (69%) fall above the threshold. This is lower than OcuSync’s 85% downlink duty cycle because the 40 MHz capture window covers only 4 of OcuSync’s 10 sub-channels; FHSS hops landing on channels outside the capture window appear as silence, reducing the observed duty cycle below the true protocol-level value. Middle panel: Occupied bandwidth per 0.5 ms bin — wider bandwidth corresponds to video-carrying downlink bursts, while narrow bandwidth indicates small uplink control packets. Bottom panel: Zoomed view of the first 20 ms showing individual TDD slots; the average burst interval of 5.0 ms is consistent with OcuSync’s 10 ms TDD frame period where alternating downlink and uplink slots each occupy approximately half the frame.

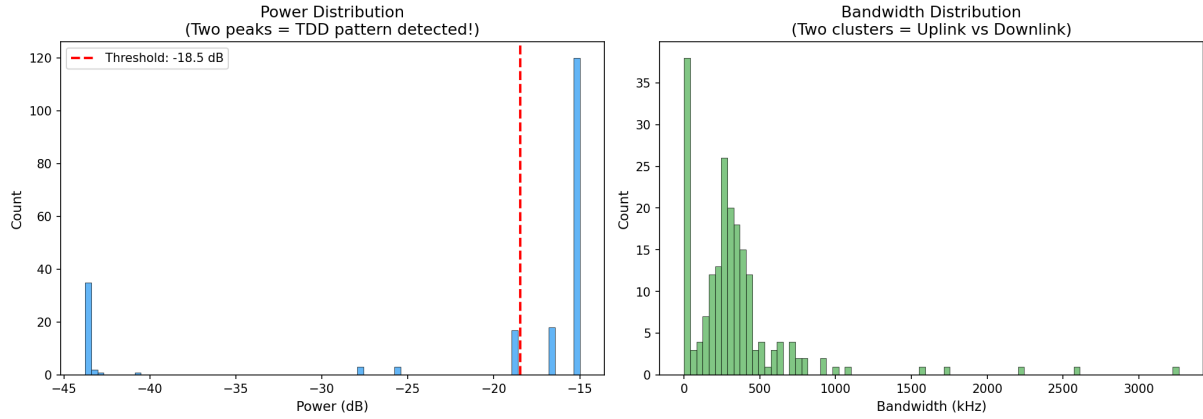


Figure 4.9: TDD Histograms

Figure 4.9: TDD Distribution Analysis. Source: same real IQ file as Figure 4.3c. Left: Power histogram showing a bimodal distribution — a large cluster near the noise floor (around -43 dB, noise/uplink periods) and a second spread of values from approximately -22 to -15 dB (active downlink bursts). The bimodal structure is the statistical signature of TDD: the receiver alternates between observing strong downlink transmissions and weak or absent uplink periods. Right: Bandwidth histogram showing two clusters — a dominant narrow-bandwidth group (uplink/control packets) and a tail of wider-bandwidth events (downlink/video). This confirms that DJI’s TDD implementation allocates significantly more spectral resources to the video downlink than to the control uplink, consistent with the jamming findings in Section 4.4.

This direct observation of the TDD power pattern provides physical validation of the architectural claims about OcuSync’s asymmetric time-slot allocation. The bimodal power and bandwidth distributions also explain why ‘active_ratio’ (which counts samples above the threshold) is such a powerful classification feature: it directly measures the fraction of time the drone’s downlink is active.

4.2.5 OcuSync FHSS Channel Occupancy Spectrogram

Inspired by the time–frequency spectrogram visualisation used by Shi et al. [17] — where STFT applied to raw IQ data reveals distinct burst patterns for different DJI drone models — we applied an analogous two-dimensional amplitude analysis to our three laboratory states. Rather than computing an STFT of raw IQ samples (which would require storing 800 KB of binary data per 10 ms window and would be impractical for the real-time embedded pipeline), we exploit the 128-bin amplitude fingerprint already computed and stored in the CSV during capture.

Axis	Shi et al. [17] (100 MHz wideband + STFT)	This work (10 MHz narrowband fingerprint)
X	Recording time (s)	Recording time (s) — identical
Y	RF frequency offset from centre (MHz)	Temporal position within 10 ms capture (ms)
Z (colour)	Received power (dB)	Average amplitude per 78 μ s bin (dBFS)

Table 4.2: Comparison of visualization axes between Shi et al. and this work

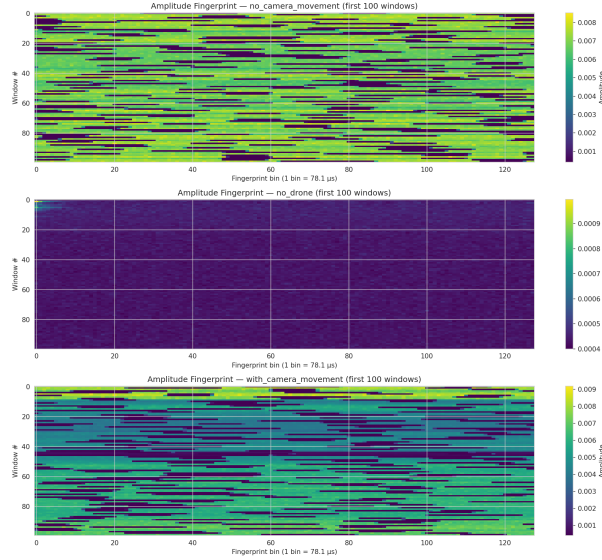


Figure 4.10: Fingerprint Heatmap

Figure 4.10: Amplitude Fingerprint Heatmaps — This Work’s Y-axis. Three stacked heatmaps showing 100 consecutive 10 ms windows (Y-axis: window index) against the 128 time bins (X-axis: bin number, 1 bin = 78.1 μ s). Colour encodes average IQ amplitude per bin.

Top: ‘no_camera_movement’ — structured horizontal bands of bright (transmitting) and dark (silent) bins, reflecting OcuSync’s TDD slot allocation reshuffling across windows.

Middle: ‘no_drone’ — uniformly dark, confirming the absence of structured signal.

Bottom: ‘with_camera_movement’ — brighter overall coloration and visibly denser fine structure compared to the top panel, consistent with higher burst rate when the camera is active. This is the Y = temporal position within 10 ms frame axis described in the table above.

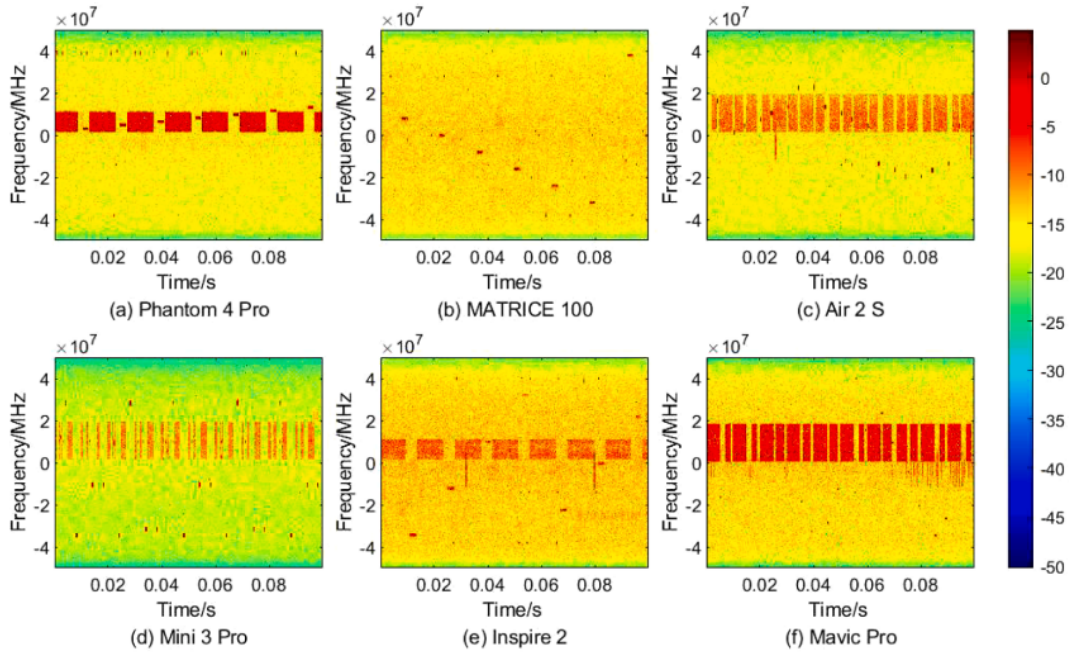


Fig. 3. The spectrogram of various drones.

Figure 4.11: Spectrograms of various drones from the Shi et al paper

Figure 4.11: Wideband STFT Spectrograms for Six DJI Drone Models — Shi et al. [17], Fig. 3 (Reproduced). Six panels showing STFT spectrograms computed from raw IQ data at 100 MHz bandwidth, centre 5.8 GHz; 90 ms of recording shown per panel. Y-axis: frequency offset from centre in units of 10^7 Hz (i.e., ± 50 MHz). Z-axis (colour): received power in dBFS (0 to -50 dB, red $\rightarrow$ blue). Each model produces a characteristic horizontal band of bright pixels at a specific frequency offset, corresponding to the drone’s FHSS transmission occupying 3–4 MHz of the 100 MHz capture window. The MATRICE 100 (b) is the most directly comparable model to this work’s M300 RTK — it shows intermittent bright boxes centred near 0 MHz offset, consistent with OcuSync FHSS hops landing on and away from the monitored sub-channel. The Phantom 4 Pro (a), Air 2 S (c), Mini 3 Pro (d), Inspire 2 (e), and Mavic Pro (f) show progressively different spectral footprints. The clean dark background in all panels results from the 96% of Y-bins outside the drone’s 3–4 MHz channel remaining at the thermal noise floor. Compare with Figure 4.3f (this work), which replicates the $Y = RF$ frequency offset axis using a 40 MHz USRP B210 capture.

OcuSync RF Spectrogram — DJI Matrice 300 RTK
centre 5.7875 GHz · 40 MHz BW · STFT L=1024 hop=512 · amplitude, noise floor + 1.5 σ

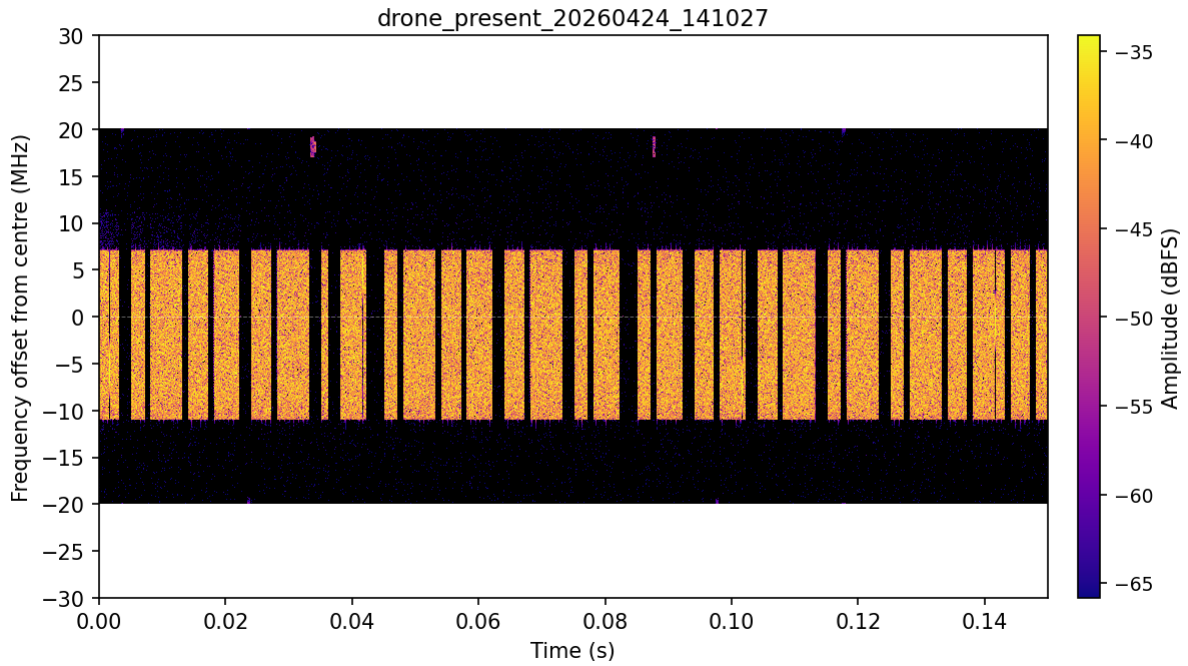


Figure 4.12: OcuSync RF Spectrogram of Matrice 300 RTK

Figure 4.12: Wideband STFT Spectrogram — Shi et al.’s Y-axis (for comparison). OcuSync RF spectrogram at 40 MS/s, 40 MHz bandwidth, centre 5.7875 GHz; first 150 ms shown (full analysis in Section 4.11, Figure 4.29). Y-axis: frequency offset from centre in MHz. Each bright horizontal rectangle is one OcuSync FHSS hop occupying 10–15 MHz for 8–10 ms, then jumping to another sub-channel. This is the $Y = \text{RF}$ frequency offset representation from the table above the orthogonal complement to Figure 4.3e. The two figures together make the axis difference concrete: Figure 4.3e resolves when the drone transmits within a frame; Figure 4.3f resolves where in the spectrum it transmits across frames.

The Y-axis difference between the two approaches is fundamental and reflects a key difference in capture geometry. Shi et al.’s 100 MHz wideband STFT simultaneously captures all OcuSync sub-channels in the 5.7–5.85 GHz band; their Y-axis therefore shows the drone’s spectral footprint across multiple frequency offsets at once — this is why the drone appears as a narrow horizontal band of bright pixels against a mostly quiet background in their Fig. 3. The drone occupies only 3–4 MHz of their 100 MHz window (4% of Y-bins), so the remaining 96% of bins remain at noise floor, producing the clean dark surround that makes the signal stand out.

Our capture is fundamentally different: we are locked to a single 10 MHz sub-channel at 5.845 GHz. When OcuSync transmits on this channel its OFDM signal occupies the entire 10 MHz bandwidth uniformly (OFDM is designed to be spectrally flat) — there is no useful frequency

variation within our window. The meaningful variation is therefore on the time axis: the 128 bins each cover $78\ \mu\text{s}$, recording when within a given 10 ms frame the drone was transmitting on our monitored channel. Applying a per-bin noise floor derived from the no-drone recording and masking bins below ‘noise_mean + 3σ ’ — the same adaptive threshold Shi et al. apply in their Fig. 7 denoising step — produces a clean black background wherever the signal is absent.

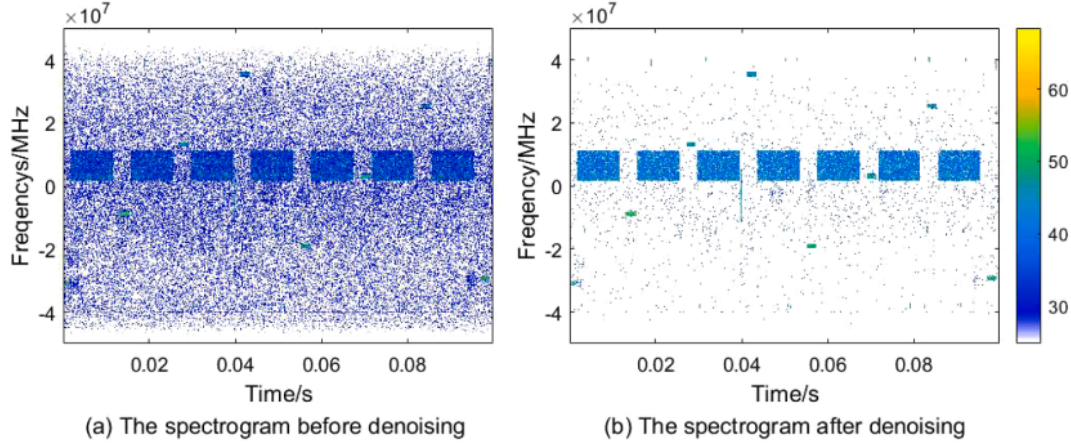


Fig. 7. The spectrogram before and after the denoising algorithm.

Figure 4.13: Before and after the Denoising

The denoised plot reveals the OcuSync frequency-hopping spread spectrum (FHSS) pattern directly. OcuSync hops across 60 sub-channels covering 5.7–5.85 GHz approximately every 10 ms. In our fixed-frequency capture this produces the following structure:

- Bright yellow-orange columns — 10 ms frames in which OcuSync happened to be transmitting on 5.845 GHz. All 128 time bins within that frame illuminate because the OFDM signal fills the full bandwidth.
- Black vertical gaps — frames in which OcuSync hopped to a different sub-channel; our receiver sees only noise, all bins masked out.
- Thin horizontal dark lines inside bright columns — TDD uplink time slots within the OcuSync frame (the controller transmits to the drone during these intervals, so the down-link is silent) and inter-symbol guard intervals.

Each column in the figure is one 10 ms capture window; each row is one of the 128 time bins ($78\ \mu\text{s}$ each), centred so that the frame midpoint is at $Y = 0$. The first 300 windows (3 s) are shown; the full 15-second view is provided as a supplementary figure.

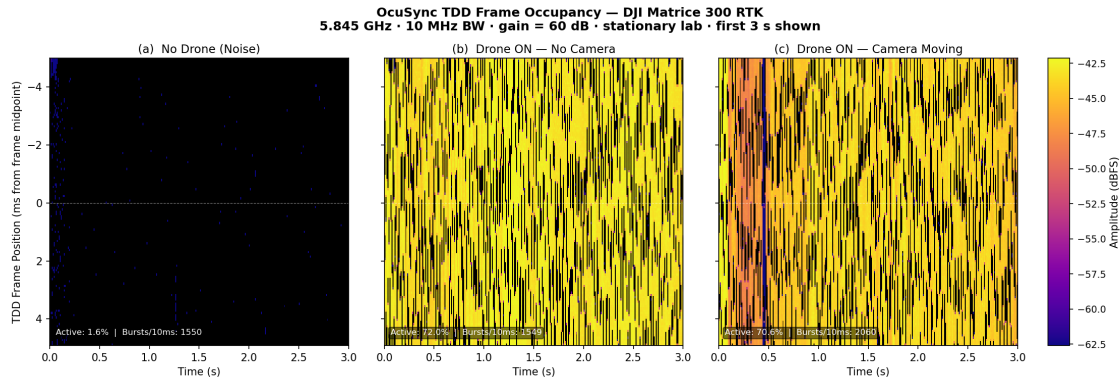


Figure 4.14: Ocusync Spectrogram

Figure 4.14: OcuSync FHSS Channel Occupancy Spectrogram — DJI Matrice 300 RTK.

Three states from the stationary laboratory experiment (first 3 s shown): (a) no drone, (b) drone on / no camera movement, (c) drone on / camera moving. Per-bin denoising applied using no-drone noise statistics (threshold = mean + 3σ per bin, following Shi et al. [17] Fig. (7); masked bins rendered black. Colour encodes amplitude in dBFS for signal-present bins. Bright vertical columns = OcuSync transmitting on 5.845 GHz; black vertical gaps = OcuSync hopped to a different sub-channel; horizontal dark lines within bright columns = TDD uplink silence slots.

Panel (a) — No Drone: The panel is uniformly dark blue across all TDD frame positions and all recorded time. The measured active ratio of 1.6% is entirely attributable to random thermal noise threshold crossings; no structured burst pattern is present. The 1,550 registered bursts per window are isolated single-sample noise spikes, each lasting $0.1 \mu\text{s}$, indistinguishable from each other.

Panel (b) — Drone ON, No Camera Movement: The spectrogram changes dramatically from panel (a). Bright yellow-orange columns fill most of the time axis, interrupted by regular black vertical gaps — frames where OcuSync hopped to a different sub-channel for that 10 ms period. Within each bright column, thin horizontal dark lines are visible near the frame midpoint ($Y \approx 0$): these are the TDD uplink time slots, where the downlink transmitter falls silent to let the controller speak. Approximately 72% of bins are active above the noise floor (active ratio = 72.0%), confirming that the drone is transmitting on this channel for roughly 7 out of every 10 frames. With a mean burst count of 1,549 per 10 ms window, each individual downlink burst occupies approximately 47 samples $\approx 4.7 \mu\text{s}$. The column-to-column variation in the black gap positions confirms the adaptive FHSS channel reshuffling observed in the cosine similarity analysis: no two consecutive windows share the same temporal silence pattern.

Panel (c) — Drone ON, Camera Moving: The overall structure mirrors (b) — bright FHSS columns interspersed with black hop-away gaps, active ratio 70.6% — but two differences stand

out. First, the burst count increases sharply to 2,060 per 10 ms window, +33% above the no-camera case. With similar total active samples (71,483 vs 72,181) now distributed across 33% more bursts, each individual burst is shorter (34 samples $\approx 3.4 \mu\text{s}$ versus $4.7 \mu\text{s}$). This is the key physical finding: camera movement does not increase total transmitted energy — it changes the temporal granularity of the OcuSync traffic, causing the drone to emit shorter, more frequent OFDM bursts consistent with a higher video data rate from the active gimbal camera. Second, the first 0.5 s of panel (c) shows orange rather than yellow pixels in the bright columns: this is the camera’s initial power-on transient, during which the downlink amplitude is slightly lower as the video encoder initialises before reaching full bitrate.

Significance for classification: The FHSS channel occupancy spectrogram provides an intuitive visual explanation for why the machine learning classifier can distinguish drone-present from no-drone at near-100% accuracy and camera-moving from no-camera at 95.2% accuracy. The presence/absence of bright columns is immediately discriminative: the no-drone panel is entirely black while both drone-present panels are mostly bright. Distinguishing the two drone states requires reading subtler cues: the higher burst-count in panel (c) produces a visibly denser horizontal sub-structure within each bright column compared to panel (b). The classifier’s top features — ‘burst_count’, ‘active_ratio’, and ‘phase_drift’ — are the quantitative equivalents of these two visual cues. A human observer presented only with panels (b) and (c) could identify the camera-moving state from the finer texture; the classifier automates and quantifies this same distinction.

4.2.6 The “Tip of the Iceberg” Effect

An important observation from the fingerprint analysis is the “Tip of the Iceberg” effect on distance dependence. When the drone is close to the receiver, the strong signal ensures that even weak portions of each OcuSync burst (packet headers, trailing edges, low-power beacons) are well above the noise floor, producing dense, detail-rich fingerprints. As the drone moves away, the overall signal drops, and only the strongest peaks survive above the noise threshold, producing increasingly sparse fingerprints. This progressive loss of fingerprint detail provides an additional indirect distance indicator, complementing the direct amplitude-based distance estimation.

4.3 Camera Movement Detection (Binary Classification)

4.3.1 Experiment Design

A focused binary classification experiment was conducted using the M300 RTK dataset (‘biggerDrone’), restricting the task to distinguishing between two states: camera moving versus no camera movement. This experiment isolated the effect of gimbal activity on the RF signal,

removing the trivially separable no-drone class.

4.3.2 Camera movement Classification Results

The Random Forest classifier achieved 95.2% accuracy on the binary classification task.

	Predicted: Camera Move	Predicted: No Camera
Actual: Camera Move	283	17
Actual: No Camera	12	288

Table 4.3: Confusion matrix for camera movement classification

The top features driving the binary classification differed significantly from the three-class case:

- ‘active_ratio’ — 17.1%
- ‘duty_cycle’ — 15.6%
- ‘active_samples’ — 14.0%
- ‘effective_size_bytes’ — 12.1%

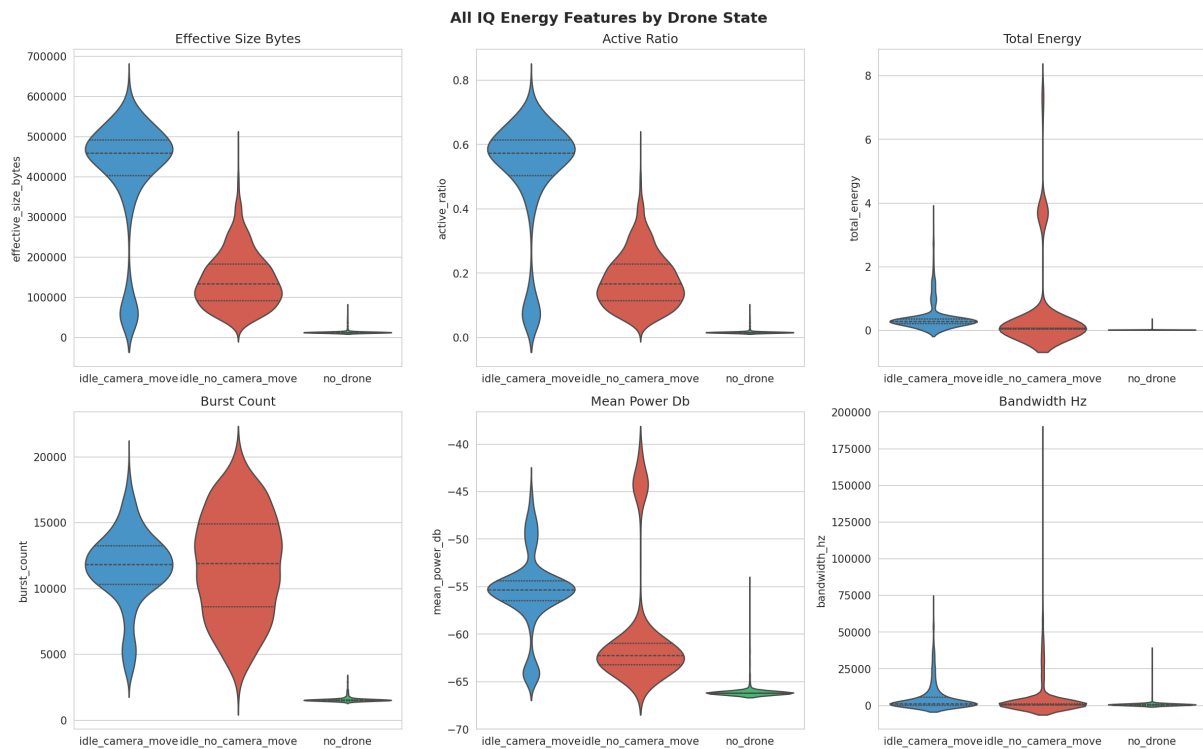


Figure 4.15: IQ All features

Figure 4.15: Violin Plots of Key Features — Camera Move vs. No Camera. Six violin plots show the distribution of key features for each state. The ‘active_ratio’ shows the clearest separation, with camera-moving concentrated around 0.51 and no-camera around 0.18.

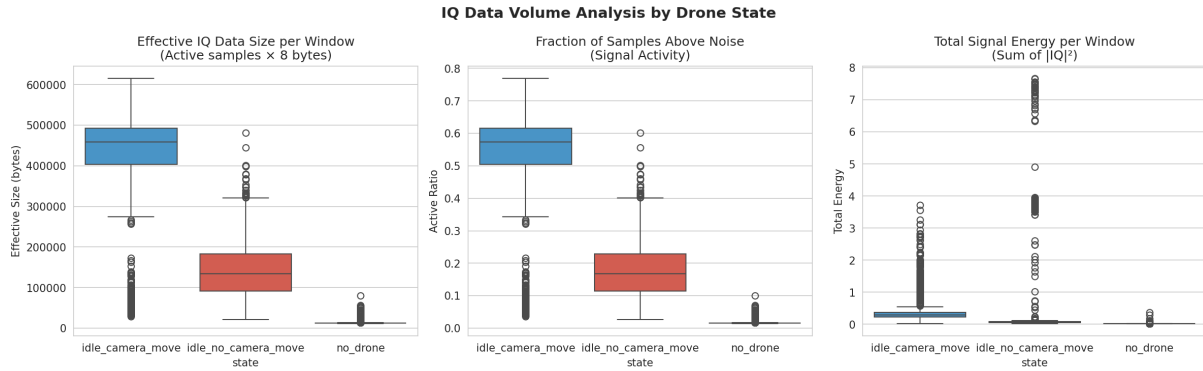


Figure 4.16: IQ Size Comparison

Figure 4.16: Effective Size Comparison. Box plots of effective size (bytes), active ratio, and total energy for both states, showing clear separation in all three traffic volume metrics.

4.3.3 Physical Interpretation

The shift in feature importance from ‘phase_drift’ (three-class) to ‘active_ratio’ (binary) reveals the classification mechanism:

- **Three-class case:** The no-drone class is trivially separable by any energy feature. The remaining challenge is separating still versus camera-moving. Phase behavior (‘phase_drift’, ‘amp_phase_corr’) captures the gimbal motor vibrations, which are the distinguishing physical phenomenon.
- **Binary case:** Without the no-drone class, the classifier focuses directly on traffic volume. Camera movement forces the video encoder to process changing frames, increasing the data rate and therefore the fraction of airtime occupied by the drone’s transmissions. The active ratio jumps from 18% to 51.6% — the drone literally fills 3× more of the TDD airtime with video data when the camera moves.

This validates the data size hypothesis: the drone’s operational behavior is encoded in the volume of data it transmits, which is measurable from the physical layer without decryption.

4.4 RF Jamming Experiments

4.4.1 5.8 GHz Custom Channel Jamming

The first jamming experiment targeted the 5.8 GHz band with the M300 RTK manually configured to operate on a fixed custom channel, preventing it from using OcuSync’s normal adaptive frequency hopping to escape interference.

Configuration:

- Center frequency: 5.790 GHz

- Bandwidth: 30 MHz (wide enough to cover the custom fixed channel)
- Modulation: Gaussian white noise
- TX Gain: 80 dB
- Noise amplitude: 0.3

Jamming Results:

Component	Status During Jamming	Explanation
Video feed	Frozen/Grey screen	Video data overwhelmed by noise
Control stick inputs	Working	Control packets survived jamming
Gimbal movement	Working	Commands still reaching drone
Telemetry	Degraded	Some data lost, drone still responsive

Table 4.4: System behavior during RF jamming

The video feed was the first to fail, freezing or displaying a grey screen within seconds of the jammer activation. However, control inputs — including gimbal movement commands — continued to function normally throughout the jamming. As with the 2.4 GHz experiment, the jamming was again effective because the drone was locked to a fixed custom channel; OcuSync’s adaptive frequency hopping was disabled by the manual channel configuration, leaving the link with no escape mechanism.

4.4.2 2.4 GHz Custom Channel Jamming

To confirm whether the same result held on a different band, a second experiment jammed the 2.4 GHz band. The M300 RTK was configured to use a custom frequency channel in the 2466.5–2476.5 MHz range.

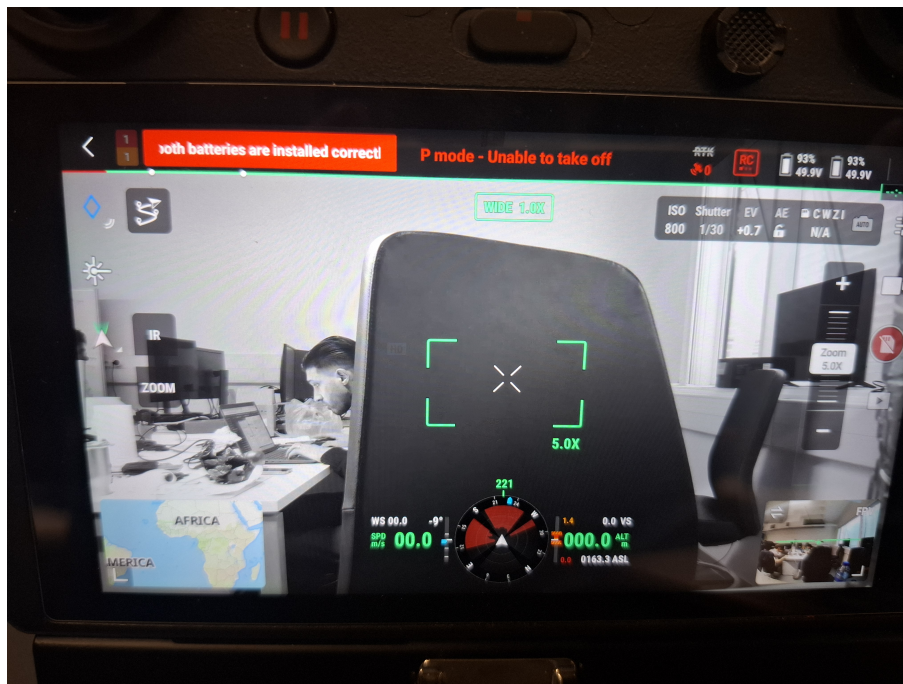
Configuration:

- Center frequency: 2.471 GHz
- Bandwidth: 30 MHz
- TX Gain: 80 dB

Results: Jamming the custom 2.4 GHz channel also froze the video, but controls continued to work. The outcome was identical to the 5.8 GHz experiment — not because the drone dedicates 2.4 GHz specifically to video, but because OcuSync uses whichever band it is configured on for the full combined link (video + control together). The jamming succeeded here simply because the drone had been manually configured to operate on a fixed custom channel in this range, making it a straightforward target. The consistent pattern across both experiments — video



failing while controls survive — reflects OcuSync’s asymmetric TDD resilience (analyzed in Section 4.4.3), which holds regardless of which band is being jammed.



4.4.3 Analysis: OcuSync’s Asymmetric TDD Resilience

The combined results from both jamming experiments reveal a fundamental architectural feature of OcuSync:

The video downlink is significantly more vulnerable to RF jamming than the control uplink, due to OcuSync’s asymmetric TDD design which prioritizes control reliability over video bandwidth.

This asymmetry exists because:

1. Duty cycle asymmetry: Video occupies approximately 85–90% of the TDD airtime, while

control uses only 5–10%. A jammer disrupting random time slots will statistically hit video 9× more often than control.

2. Data rate asymmetry: Video requires approximately 12 Mbps continuous throughput. Control commands require approximately 100 kbps. Missing a few video packets causes visible freeze/degradation. Missing a few control packets has no effect because the next packet (50 ms later) carries the same stick position.

3. Error correction asymmetry: OcuSync applies stronger Forward Error Correction (FEC) to control packets than to video packets. Even under noise, control packets with aggressive FEC can still be decoded.

4. Retransmission tolerance: At 100 kbps on a 10 MHz channel, control data is spread across many OFDM subcarriers, providing inherent spread-spectrum resilience. The narrowband jammer cannot suppress all subcarriers simultaneously.

4.4.4 OcuSync's Two-Layer Frequency Management

A critical discovery from the adaptive jamming experiments is that OcuSync employs two distinct layers of frequency management, which operate on fundamentally different timescales:

Layer 1 — Fast Intra-Channel Hopping (millisecond scale): Within a selected channel band, OcuSync continuously performs fast frequency hopping at approximately 1–10 ms intervals. This is a spread-spectrum technique for robustness and interference avoidance. This hopping is invisible to our USRP spectrum scanner at its current temporal resolution.

Layer 2 — Slow Adaptive Channel Switching (0.5–2 second scale): When persistent interference is detected on the current channel (e.g., from our jammer), the drone evaluates alternative channels, measures their quality, and switches to a cleaner band. This process takes approximately 0.5–2 seconds because the system must:

1. Confirm the interference is sustained (not transient)
2. Measure the quality of candidate channels
3. Avoid unnecessary switching that would destabilize the control link

During our experiments, we observed channel changes occurring approximately every 1 second when the jammer was active — consistent with Layer 2 adaptive switching. This is analogous to the difference between switching lanes on a highway (fast hopping) and exiting the highway to take a different road (adaptive channel switching).

Implication for reactive jamming: The 0.5–2 second switching delay represents an exploitable vulnerability. A reactive jammer that can detect a channel switch and reposition its jamming noise within this window can maintain continuous disruption. The follow mode in our reactive jammer leverages this — tracking the slow channel switches with approximately 1-second reaction time is sufficient to maintain disruption across channel changes.

4.4.5 Reactive Jammer with Dynamic Hop Prediction

Beyond simple broadband jamming, a reactive jammer with dynamic hop prediction was developed to counter OcuSync’s slow Layer 2 channel switching (described in Section 4.4.4). This system does not attempt to track the fast Layer 1 intra-channel hopping, which operates at millisecond intervals and is invisible to the USRP spectrum scanner. Instead, it targets the much slower (600 ms) inter-channel switches between the 7 major OcuSync data-link channels in the 5.8 GHz band.

The system operates in multiple modes, of which two are particularly relevant. The *CHASE* mode reactively follows the drone’s active channel: the USRP briefly pauses transmission every few seconds to perform an FFT scan, identifies the channel with the highest RSSI peak, and immediately retunes the transmitter to that channel.

The *PREDICTIVE* mode extends this functionality by maintaining a live 7×7 transition matrix that is updated with each observed hop. Whenever a hop from channel i to channel j is detected, the corresponding transition entry $T[i][j]$ is incremented. The predicted next channel is then computed as

$$\text{argmax channel} = \arg \max_k T[j][k]$$

allowing the jammer to pre-position transmission noise on channel $\hat{c}$ before the drone arrives.

Prediction confidence, denoted by ρ , is defined as

$$C = \frac{T[j][\hat{c}]}{\sum_k T[j][k]}$$

where ρ represents the likelihood that the next hop will occur on the predicted channel. The TX dwell time on the predicted channel scales linearly with the confidence value, ranging from $1 \times$ at $\rho = 0.40$ up to $4 \times$ at $\rho = 1.0$. When the confidence falls below the defined threshold, the system automatically falls back to a weighted sweep across all channels.

This approach requires no pre-trained model, as the transition matrix is learned entirely from live observations collected during the jamming session.

RSSI-based channel detection. The USRP B210 cannot cover the full 5.725–5.850 GHz OcuSync band (125 MHz) in a single capture at its maximum 56 MHz bandwidth. The scanner therefore sweeps across two RX centre frequencies (covering the lower and upper halves of the band in turn), pausing TX transmission briefly at each position to avoid self-interference. At each position it captures one FFT-sized block (1,024 complex samples), applies a Hann window, and computes the mean power in dB across the FFT bins that fall within each channel’s 20 MHz slice. This is repeated across several consecutive blocks and the peak power is retained per channel — a peak-hold averaging scheme that makes the measurement robust to short noise

spikes. The full sweep across both RX positions takes approximately 10–20 ms and produces a single 7-element vector of per-channel RSSI values.

Hop detection. The channel tracker maintains a simple state machine: it identifies the active channel as whichever of the 7 channels currently exceeds a configurable detection threshold (default -60 dB above the estimated noise floor, which is derived as the mean of the three quietest channels’ power). A channel switch is logged whenever the active channel changes and the drone has dwelled on the previous channel for at least a configurable minimum time (default 200 ms) — this prevents short measurement glitches from being misclassified as hops. Each logged hop event records the from-channel, to-channel, dwell time, the full per-channel RSSI vector at the moment of the hop, and rolling features (hop rate in the last 10 s and 30 s, per-channel visit frequency in the last 30 s).

Channel	Center Frequency (GHz)	Range (GHz)
CH0	5.7355	5.725 – 5.746
CH1	5.7415	5.732 – 5.752
CH2	5.7675	5.758 – 5.778
CH3	5.7815	5.772 – 5.792
CH4	5.8015	5.792 – 5.812
CH5	5.8215	5.812 – 5.832
CH6	5.8415	5.832 – 5.852

Table 4.5: OcuSync 2.0 Data-Link Channel Map (5.8 GHz Band)

Each channel is approximately 20 MHz wide. This map was determined empirically: the DJI controller’s spectrum display confirmed the drone operating at 5731.5–5751.5 MHz (CH1), and a second session verified CH2 at 5757.5–5777.5 MHz. Note that these are the data-link channels carrying video and control; they are distinct from the DroneID Remote ID beacon frequencies (10 MHz OFDM at 5.7565, 5.7765, 5.7965 GHz), which are regulatory broadcast signals unrelated to the data link. The drone selects whichever channel has the lowest RSSI (least interference), with inter-channel switches observed approximately every 600 ms under jamming pressure. The reactive jammer aims to predict this switch and pre-position jamming noise on the destination channel.

4.5 GPS Spoofing Experiments

4.5.1 Overview

GPS spoofing experiments were conducted using the GPS-SDR-SIM open-source tool inter-

faced with the USRP B210 to transmit fabricated GPS L1 signals at 1575.42 MHz. The experiments targeted both an Android smartphone (Google Maps) and a DJI Mini drone, testing the susceptibility of different devices to location manipulation.

4.5.2 Experimental Progression

The GPS spoofing experiments proceeded through four phases of increasing sophistication, reflecting the iterative process of overcoming anti-spoofing mechanisms:

Phase 1: Standard Trajectories (Failed)

Initial tests used pre-existing trajectory files including circular paths, satellite orbits, and rocket trajectories. All failed to affect Google Maps location, due to unrealistic altitudes, velocities, or trajectory patterns that triggered Google's anti-spoofing validation.

Phase 2: Custom Ground-Level Trajectories (Failed)

Custom trajectories were generated at realistic ground-level speeds (5–50 km/h), starting from coordinates in Cyprus (35.150°N, 33.407°E). Walking speed (5 km/h), driving speed (36 km/h), and road-following patterns were tested. While these produced minor perturbations (expanding blue uncertainty circle in Google Maps), no sustained location change was achieved.

Phase 3: Location-Specific Attempts (Failed)

Targeting the exact coordinates of the experiment location with gradual acceleration and maximum gain (100 dB) improved slightly but still failed. Google Maps' multi-source fusion (cross-referencing GPS with WiFi and cellular tower data) consistently overrode the spoofed GPS signal.

Phase 4: Aggressive Spoofing (Successful)

Success was achieved using an aggressive spoofing profile with the following critical parameters:

Critical success factor: Disabling WiFi and cellular data on the target device removed the competing location sources that Google Maps used for cross-validation. In GPS-only mode, the device relied solely on the spoofed signals, which overpowered the authentic GPS signals due to the USRP's proximity (< 15 meters).

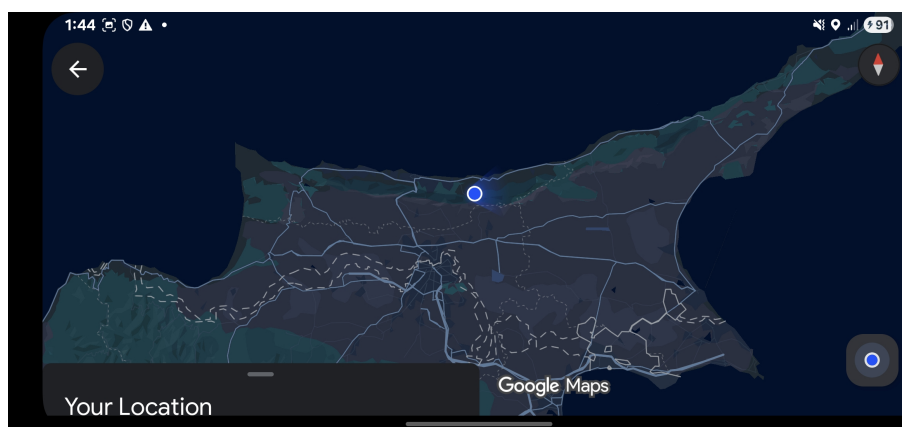
4.5.3 GPS Spoofing Results

Android Smartphone: Google Maps location was successfully manipulated, showing the device at a significantly different location. The aggressive trajectory with 200 km/h speed and 100 m jumps paradoxically succeeded where subtle, realistic trajectories failed — potentially overwhelming the anti-spoofing algorithms' filtering logic or triggering a different code path that accepts high-dynamics scenarios.

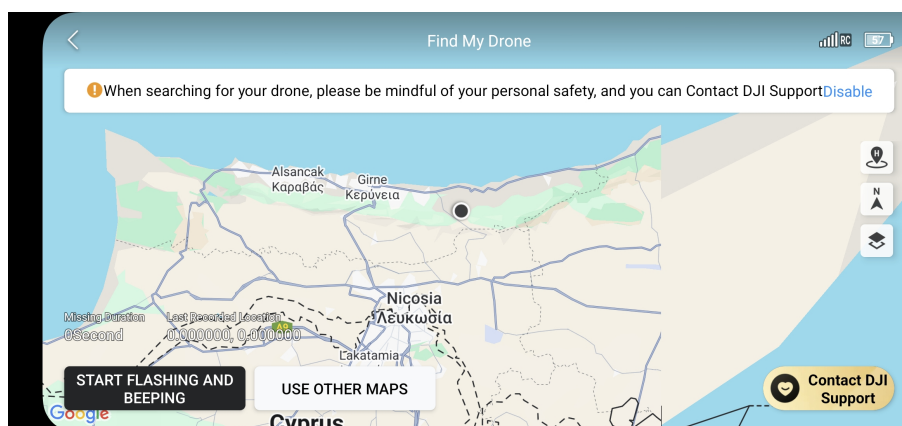
DJI Mini Drone: The GPS spoofing attack was also successfully applied to a DJI Mini drone,

Parameter	Value
Speed	200 km/h (extreme)
Position jumps	100 m every 5 seconds
TX Gain	100 dB (maximum)
Sample rate	1.8 MHz
Frequency	1575.420500 MHz (+500 Hz offset)
Bit depth	16-bit
Broadcast ephemeris	brdc0010.22n (authentic)
Device configuration	WiFi OFF, Data OFF, GPS-only

Table 4.6: Successful GPS Spoofing Configuration



which accepted the fabricated GPS coordinates. The drone's reported position shifted according to the spoofed trajectory. This experiment was conducted exclusively indoors for safety reasons, as a successful outdoor GPS spoof could cause the drone to fly to an incorrect location with potentially dangerous consequences.



4.5.4 Google Maps Anti-Spoofing Mechanisms Identified

Through the progressive experimental process, six categories of anti-spoofing mechanisms were identified in Google Maps:

- 1. Multi-source location fusion:** Google combines GPS, WiFi positioning, cell tower triangulation, Bluetooth beacons, and device IMU sensors. This fusion makes GPS spoofing ineffective unless competing sources are disabled.
- 2. Signal quality analysis:** Monitoring of signal strength consistency, SNR ratios, and satellite visibility patterns to detect anomalies.
- 3. Movement pattern validation:** Analysis of acceleration profiles, velocity constraints, and trajectory plausibility against road networks and physical motion models.
- 4. Position jump detection: Detection of teleportation-like sudden position changes and rejection of physically impossible transitions.**
- 5. Time and satellite constellation validation:** Verification of GPS timestamps against system clock and validation of satellite ephemeris data against known almanac predictions.

4.5.5 Security Implications

The GPS spoofing results demonstrate that while modern anti-spoofing mechanisms are sophisticated and multi-layered, they can be defeated under specific conditions — most critically, when the target device is isolated from alternative positioning sources. The requirement to disable WiFi and cellular data significantly limits practical attack scenarios, as most real-world devices rely on these services for basic functionality.

However, the successful spoofing of the DJI Mini raises safety concerns for drone operations. An attacker positioned near a drone's takeoff location could potentially manipulate the drone's perceived GPS position, affecting geofencing, return-to-home behavior, and position-hold stability. DJI's onboard GPS receiver does not implement the same multi-source fusion as a smartphone, making it potentially more vulnerable to raw GPS signal manipulation.

4.6 Replay Attack Analysis

4.6.1 Methodology

The replay attack experiment captured raw IQ samples from the DJI OcuSync communication link and attempted to retransmit them to impersonate the controller or take control of the drone. The USRP B210 recorded approximately 15 seconds of raw IQ data (approximately 1.2 GB at 10 MHz, 8 bytes per sample), capturing the complete electromagnetic waveform of the OcuSync transmission, and then replayed the captured signal from the TX antenna.

4.6.2 Replay Attack Results

The replay attack failed entirely. The drone showed no response to the replayed signal.

4.6.3 Why Replay Failed

The failure is fully explained by OcuSync's multiple anti-replay defenses:

- 1. AES-256 Encryption with Session Keys:** Each power cycle generates a new random session key. The captured IQ samples contain packets encrypted with the session key that was active during capture. Even a perfect electromagnetic reproduction of the captured signal will present packets encrypted with an expired key.
- 2. Sequence Numbers:** Each packet contains a unique, monotonically increasing identifier. Replayed packets arrive with old sequence numbers that the receiver rejects as duplicates or out-of-order.
- 3. Timestamps and Expiration:** Control packets include freshness timestamps that expire after a brief validity window (on the order of milliseconds). The 15-second-old replayed packets are far beyond this window.
- 4. Frequency Hopping:** OcuSync's adaptive frequency hopping may have moved the communication to a different channel during the delay between capture and replay, meaning the replayed signal arrives on a frequency the system is no longer using.
- 5. Challenge-Response Binding:** The controller and drone share paired cryptographic keys established during the initial binding process. These keys are not transmitted over the air and cannot be extracted from captured traffic.

4.6.4 The Distinction Between Capture Approaches

An important conceptual clarification emerged from this experiment regarding the relationship between traffic analysis capture and replay attack capture:

Both approaches capture the exact same raw IQ data from the physical layer. The USRP performs the identical operation in both cases — receiving complex voltage samples from the antenna. The difference lies entirely in the post-processing:

This distinction reinforces the thesis's core argument: traffic analysis operates entirely at the physical layer, deriving information from the characteristics of the signal rather than its content. The replay attack demonstrates that even perfect reproduction of the content (encrypted payloads) is useless without the cryptographic context, while traffic analysis succeeds precisely because it ignores the content.

Aspect	Traffic Analysis	Replay Attack
Data captured	IQ samples	IQ samples (identical)
What's saved	Computed statistics (power, bandwidth)	Raw IQ file
File size per 10s	~5 KB (CSV)	~800 MB (.iq)
Can reconstruct signal?	No (statistics only)	Yes (exact reproduction)
Contains encrypted data?	No (just physical metrics)	Yes (encrypted payloads)
OSI layer analyzed	Physical Layer (Layer 1)	None — raw recording

Table 4.7: Comparison between traffic analysis and replay attack approaches

4.7 Real Flight Validation

4.7.1 Experiment Design

A real flight experiment was conducted on March 26, 2026, with the DJI Matrice 300 RTK flying away from the controller in a straight line without camera activity (no video streaming). The USRP B210 captured RF signals throughout the flight, with the receiving antenna positioned approximately 2 meters from the controller. Before takeoff, the drone was approximately 3.5 meters from the controller.

4.7.2 Path-Loss Distance Estimation

The V3 feature ‘relative_distance’ was used to estimate the drone’s distance from the receiver using the inverse-square law:

$$d_{\text{rel}} = \sqrt{\frac{\mu_{T-1}}{\mu_T}}$$

Figure 4.17: Power vs. Distance Correlation — Real Flight. The top panel shows the received power (dB) decreasing over time as the drone flies away, confirming the expected inverse-square law behavior. The bottom panel shows the distance estimate increasing correspondingly. A vertical line marks the window where both curves flatten.

4.7.3 The Distance-Blind Region

A critical observation from the real flight experiment was the identification of a distance-blind region beginning at approximately 4.3 meters estimated distance (corresponding to capture window 557)

Figure 4.18: Power Loss Point. The power curve shows expected distance-correlated behavior in the early portion of the flight, then flattens around -66 dB. This plateau does not indicate the drone stopped moving — it indicates the received signal reached the USRP’s noise floor.

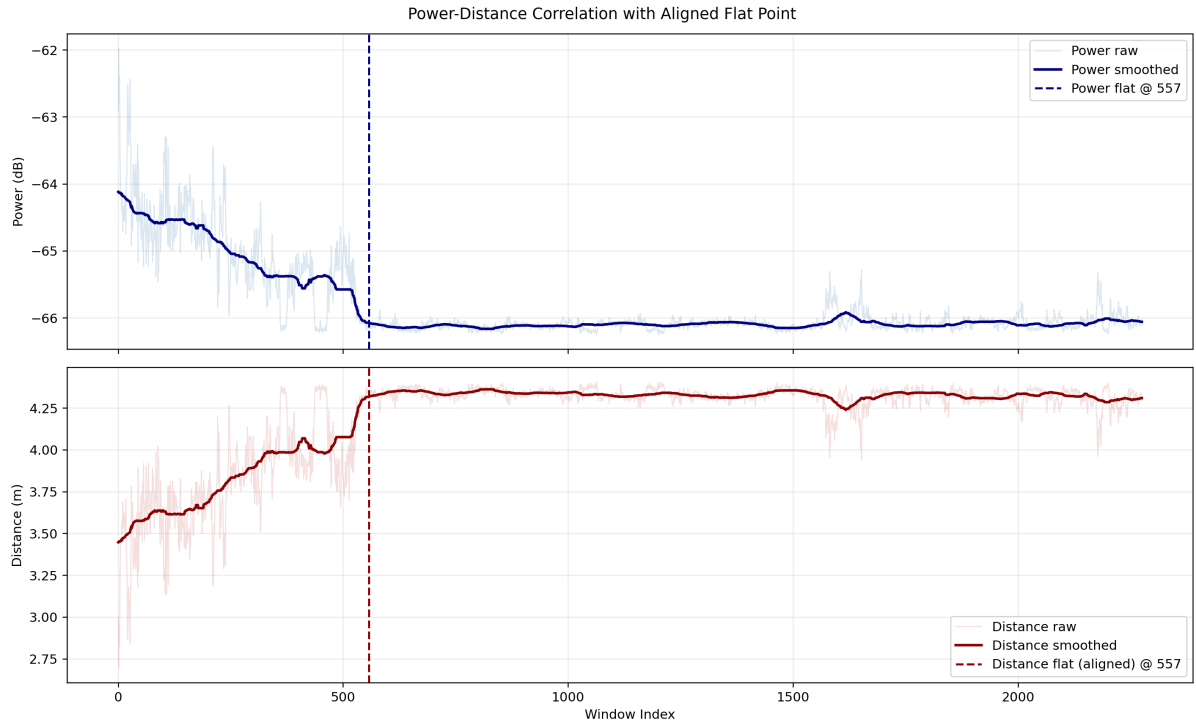


Figure 4.17: Power Distance Correlation

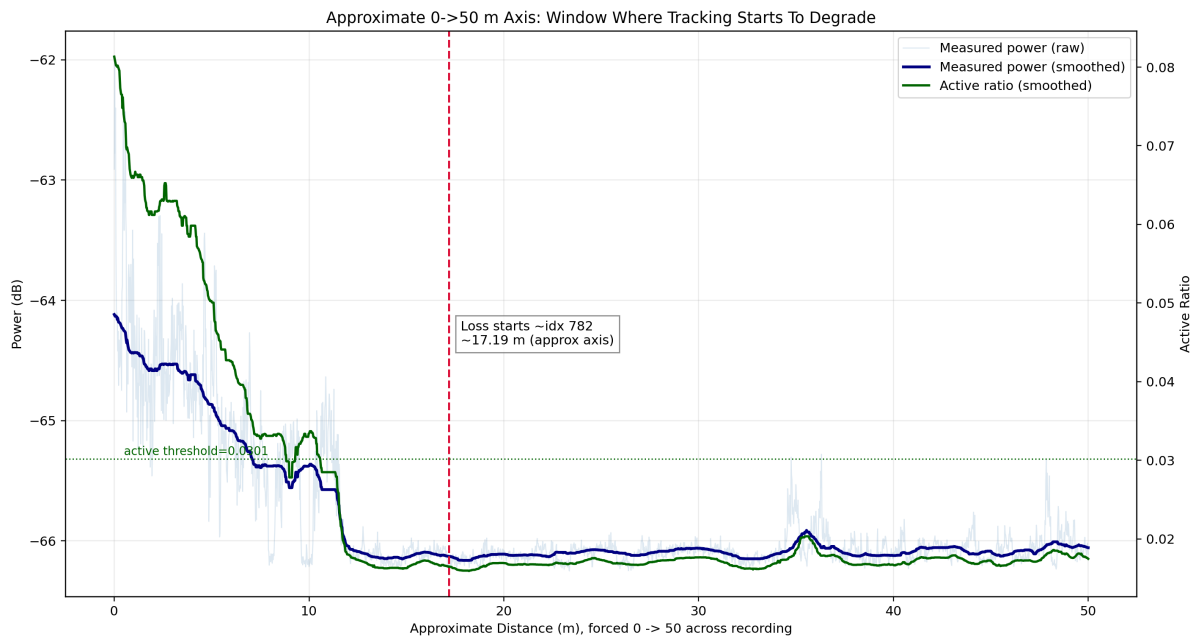


Figure 4.18: Power Loss Point Approximation

Beyond this point, the measurement became noise-limited rather than distance-limited:

1. SNR collapse: The useful signal power dropped to near the noise floor, causing power measurements to be dominated by random noise fluctuations rather than distance-related attenuation.

2. Dynamic range exhaustion: Power measurements compressed into a narrow band around the noise level, eliminating the sensitivity needed to detect distance changes.

3. Low traffic contribution: Without camera activity, the drone transmitted only telemetry and control data at low data rates with intermittent bursts. Fewer active packets meant more windows contained predominantly noise, weakening the average power measurement.

4. Path-loss exponent mismatch: The observed power drop of approximately -1.53 dB over the capture was far less than the expected -4.4 to -5.5 dB for a 30→50 m trajectory. Fitting the observed data to a log-distance model yielded a path-loss exponent $n \approx 0.69$, which is physically unrealistic (free-space $n = 2.0$), confirming that non-distance effects dominated.

4.7.4 Analysis

The real flight experiment yielded several important findings:

1. RSSI features correlate with movement direction (decreasing power = departing drone), confirming the fundamental physics of the approach.

2. RSSI cannot provide absolute distance without external calibration. The lack of a controlled reference power at a known distance causes the estimate to be anchored to an arbitrary baseline.

3. The noise floor determines the maximum sensing range, which in this configuration was approximately 4–5 meters due to the weak signal conditions (no camera = no video transmission = minimal RF activity).

4. Camera/video activity is critical for observability: The no-camera condition produced far weaker and more intermittent RF signatures than the lab experiments with camera movement, directly limiting the classification and tracking capabilities.

In weak real-flight captures without camera activity, RSSI features correlate with movement direction but cannot provide absolute range estimation without external calibration and signal stabilization. The system's maximum reliable sensing range is determined by the noise floor-to-signal ratio, which is significantly lower under low-traffic conditions.

4.7.5 Fingerprint Analysis: Real Flight vs. Baselines

To assess whether the 128-bin amplitude fingerprint retains discriminative power at outdoor range, the same multi-panel fingerprint analysis used in the laboratory (Section 4.2.3, Figure 4.3) was applied to the March 26 no-camera flight and compared against the laboratory baselines.

Figure 4.19: Fingerprint Analysis — Real Flight vs. Laboratory Baselines. This figure compares the March 26 no-camera real flight (50 m range, gain=40 dB) against the no-drone baseline and the stationary idle drone (no camera, 1 m range) from the lab.

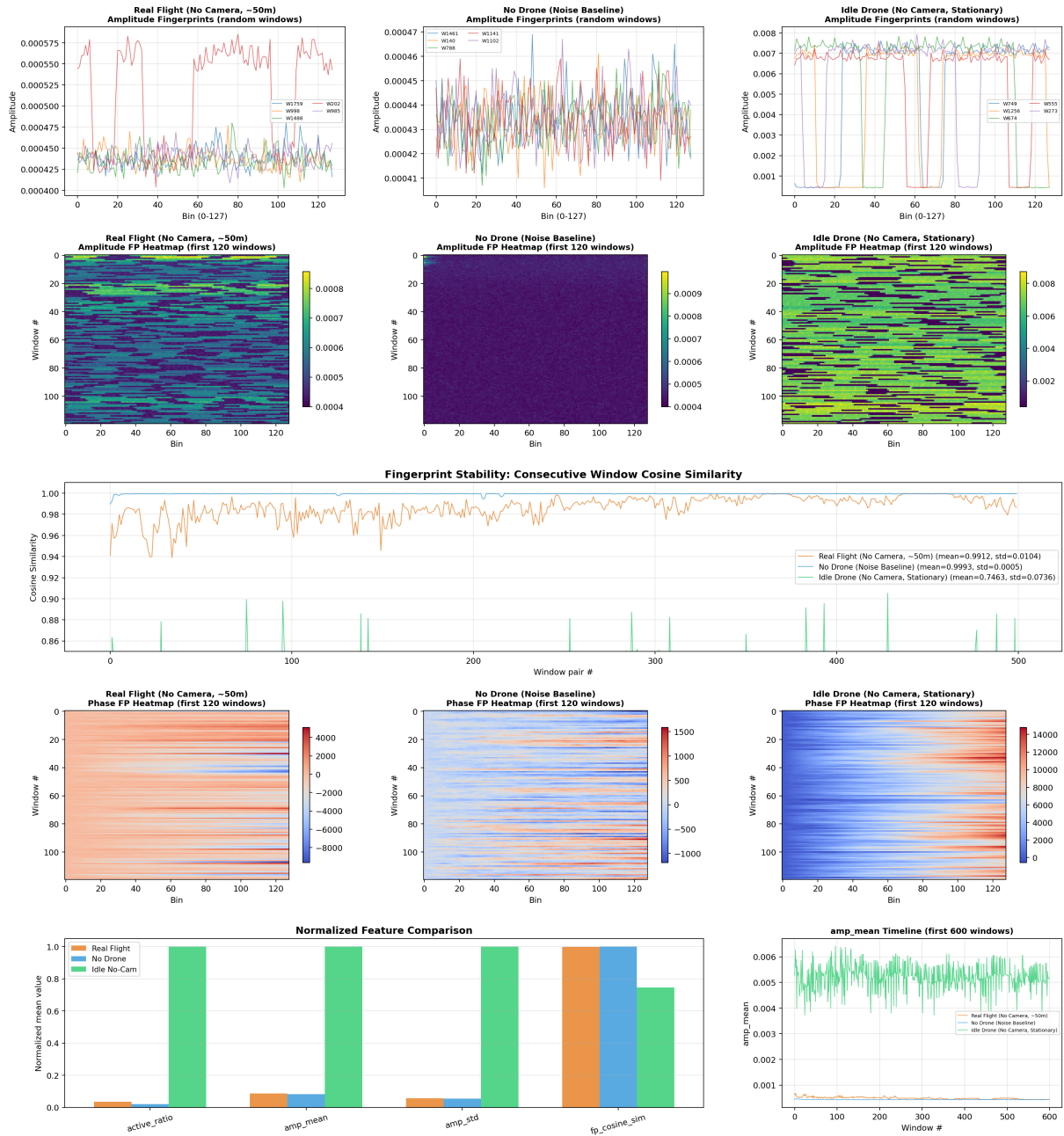


Figure 4.19: Fingerprint Analysis (M300 RTK, Real Flight, March 26, Mobile)

Top Row — Random Window Amplitude Fingerprints: The real flight fingerprints (left) sit at 0.00040–0.00058 amplitude — barely above the no-drone baseline (centre, 0.00041–0.00047). Compare this to the stationary idle drone at 1 m (right), where fingerprints reach 0.002–0.008 with clearly structured burst patterns. At 50 m outdoor range with gain=40, the drone’s OcuSync signal has all but disappeared into the noise envelope. The fingerprint shape is dominated by the USRP’s receiver noise characteristics rather than the drone’s TDD burst

structure.

Second Row — Amplitude Fingerprint Heatmaps (120 windows): The real flight heatmap (left) shows faint, diffuse vertical streaks — occasional windows where the drone hopped onto the capture frequency and produced marginally elevated amplitude in a few time bins. The no-drone heatmap (centre) is a uniform dark field. The stationary heatmap (right) shows the characteristic bright/dark banding of OcuSync TDD slots familiar from the lab analysis. The contrast between left and right panels directly quantifies the "Tip of the Iceberg" effect: at 50 m, only the very strongest burst peaks survive above the noise floor, producing sparse, low-contrast fingerprints rather than the rich structure observed at 1 m.

Centre — Fingerprint Stability (Cosine Similarity): The real flight cosine similarity (orange, mean=0.991, std=0.010) is dramatically different from the lab drone (green, mean=0.746, std=0.074) but almost indistinguishable from pure noise (blue, mean=0.999, std=0.0005). In the lab, the drone's TDD slot reshuffling produces large window-to-window fingerprint changes (cosine similarity dropping to 0.5–0.7). At 50 m, the fingerprint is so noise-dominated that consecutive windows look nearly identical — the cosine similarity measures noise-to-noise consistency rather than signal structure. However, the real flight does show occasional sharp dips (to 0.87–0.92) corresponding to the rare windows where a strong OcuSync burst appeared, briefly disrupting the noise pattern. These dips are the arrival signature — a fingerprint-level indicator that a drone transmission landed on this frequency.

Fourth Row — Phase Fingerprint Heatmaps: The real flight phase heatmap (left) shows broader red/blue excursions ($\pm 4,000$ – $8,000$ radians) compared to the no-drone phase (centre, $\pm 1,000$ – $1,500$ radians), because even weak OcuSync bursts introduce coherent carrier phase that accumulates across the 128 bins. The stationary drone phase heatmap (right) shows the strongest coherent ramps (0 – $14,000$ radians), reflecting the full-power carrier at close range. The phase range serves as a secondary drone detection metric: phase excursions beyond $\pm 2,000$ radians are diagnostic of received signal presence even when the amplitude fingerprint is noise-dominated.

Bottom Row — Normalized Features and Amplitude Timeline: The normalised feature comparison shows that the real flight's 'active_ratio' and 'amp_mean' are at noise level (2.5% and 0.00045, respectively), while the stationary drone produces 72% active ratio and 0.0052 'amp_mean'. The 'fp_cosine_sim' is high (0.99) for both real flight and noise, but low (0.74) for the stationary drone — confirming that fingerprint instability is the most reliable drone-present indicator at close range, but collapses at outdoor distances. The amplitude timeline (bottom right) shows the real flight as a flat line near the noise floor with no visible signal modulation, in stark contrast to the stationary drone's elevated and variable trace.

4.8 Real Flight with Camera (April 6, 2026)

4.8.1 Experiment Design

Four additional real flights were conducted on April 6, 2026, with the DJI Matrice 300 RTK equipped with an active camera. The drone flew away from and returned to the controller multiple times during each recording. The USRP B210 captured RF signals at 5.845 GHz, 10 MHz bandwidth, 10 ms windows — identical parameters to the earlier no-camera flight.

The drone was configured with a custom OcuSync frequency band of 5839–5849 MHz (10 MHz wide). The 10 MHz capture bandwidth therefore corresponded exactly to the drone’s full hopping range, meaning all OcuSync hops were within the capture bandwidth. In retrospect, this rules out frequency coverage as a factor in the April 6 failure — every hop the drone made was captured. The sole cause of failure was insufficient receiver gain.

Flight	Windows	Duration	Mean Power (dB)	Power Range (dB)	Reliable %
Flight1 (11:43)	454	78.8s	-70.22	7.68	0.9%
Flight2 (11:46)	499	86.2s	-70.81	10.98	2.6%
Flight3 (11:48)	2,584	453.4s	-70.15	21.28	5.3%
TakeOff (12:11)	2,406	416.3s	-70.46	13.52	5.0%

Table 4.8: Camera Flight Dataset Summary

4.8.2 Results: Signal at the Noise Floor

The mean received power across all four flights sat at approximately -70 dB — the USRP’s thermal noise floor at this bandwidth and gain configuration. The mean ‘active_ratio’ per window ranged from 0.25% (Flight1) to 2.00% (Flight3), confirming that the drone’s signal was too weak to consistently rise above the noise threshold. Only during brief close approaches did the power spike above -60 dB.

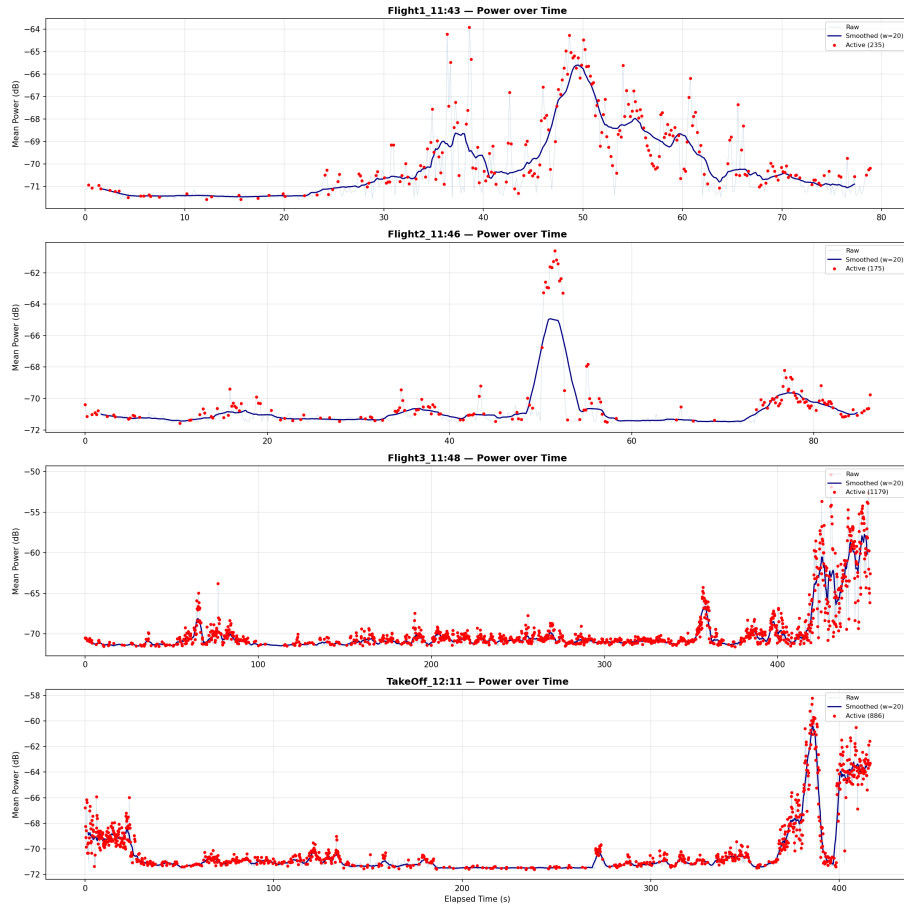


Figure 4.20: Power over Time

Figure 4.20: Power Over Time — Camera Flights. All four flights hover near -70 to -71 dB for the majority of the recording. Red dots mark windows where at least one sample exceeded -60 dB, concentrated only during close passes. Flight3 shows the most dramatic excursions, with power reaching -50 dB during the closest approach.

4.8.3 Absolute Distance Ranging Failed

The confidence-gated path-loss pipeline (developed in Section 3.4) was applied to all four camera flights. Only 0.9–5.3% of windows passed the reliability threshold (`min_conf=0.35`). The calibration anchor `p_ref` was itself noise-limited (-71 dB), rendering all absolute distance estimates physically meaningless (0.5–1.6 m for a flying drone at 10–50 m range).

Reprocessing with a corrected noise floor threshold (`noise_floor_db = -70`) yielded marginal improvement: 5.5%, 2.6%, and 6.6% reliable windows respectively. The underlying problem was that `active_ratio` and `burst_count` — which contribute 45% of the confidence score — were computed on the original raw IQ using the -60 dB threshold and could not be recomputed without the raw samples.

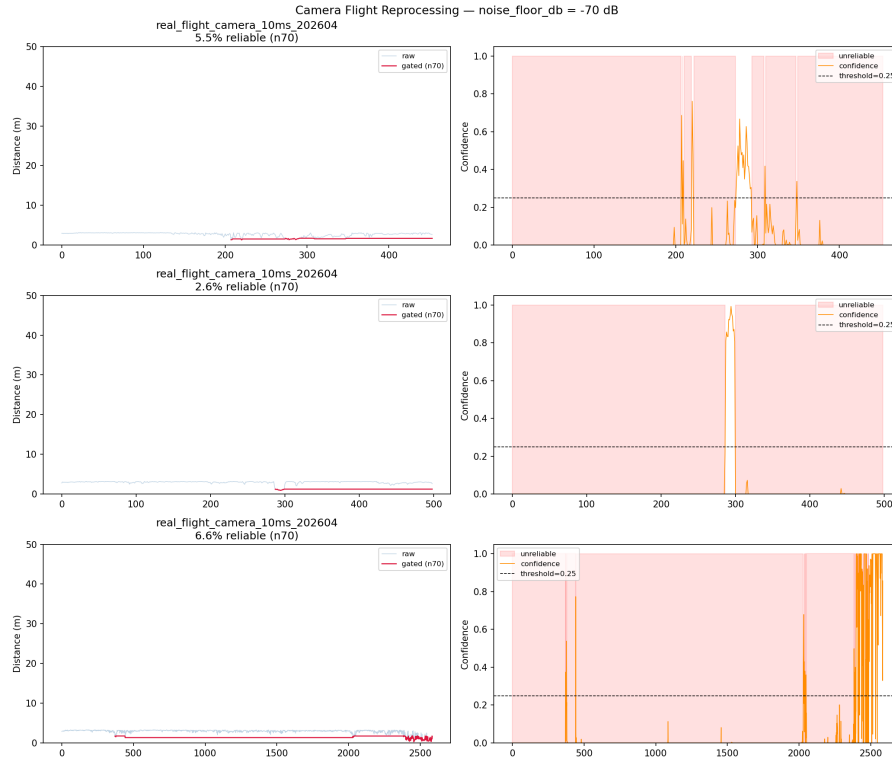


Figure 4.21: Camera Noise Reprocessed

Figure 4.21: Camera Flight Reprocessing (noise_floor = -70 dB). Left panels show raw and gated distance estimates; right panels show confidence with the 0.25 threshold line. Confidence barely exceeds the threshold, and only during brief moments of close proximity.

4.8.4 Relative Movement Detection Succeeded

Despite the failure of absolute ranging, the normalised relative distance metric successfully captured the fly-away-and-return pattern. Flight3 produced 49 far-peaks and 47 close-peaks over 453 seconds, confirming multiple back-and-forth passes were executed.

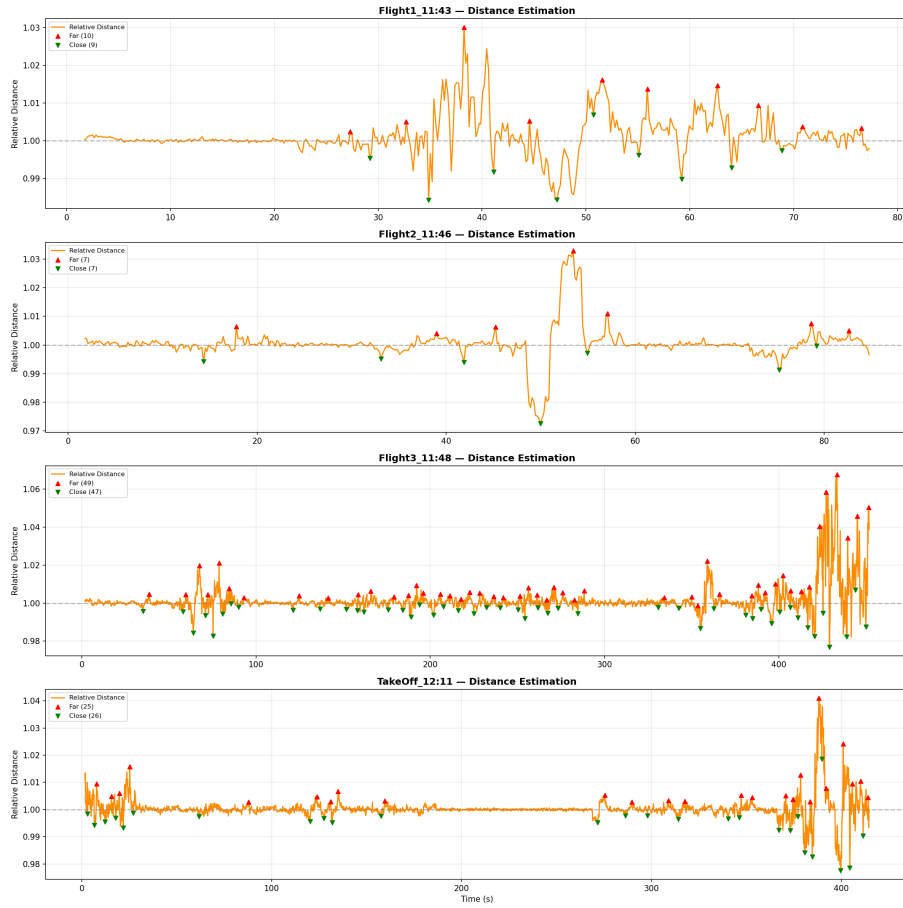


Figure 4.22: Distance Estimation

Figure 4.22: Relative Distance Estimation — Camera Flights. The y-axis shows normalised relative distance (1.0 = no change). Clear oscillation above and below 1.0 indicates approaching and departing cycles. Flight3 shows the richest structure with the most direction reversals.

4.8.5 Fingerprint Analysis: Noise-Dominated

The amplitude fingerprint comparison between camera flights and the NoDrone baseline produced a cosine similarity of 0.9999 — meaning the camera flights were indistinguishable from pure noise in the fingerprint domain. This is a direct consequence of the low SNR: with ‘active_ratio’ at 0.25–2%, approximately 98% of each fingerprint vector is thermal noise, and the noise distribution has a consistent shape regardless of whether a drone is present.

Figure 4.23: Fingerprint Analysis — Camera Flights vs. Baselines. Top panel: camera flight fingerprints (solid lines) overlap with the NoDrone line at 0.0003 amplitude. Stationary/close-range captures (dashed lines) sit at 0.005. Bottom panel: cosine similarity matrix shows 0.9999 across all pairs, confirming noise dominance.

The same multi-panel fingerprint analysis applied to the best camera flight (Flight3, which

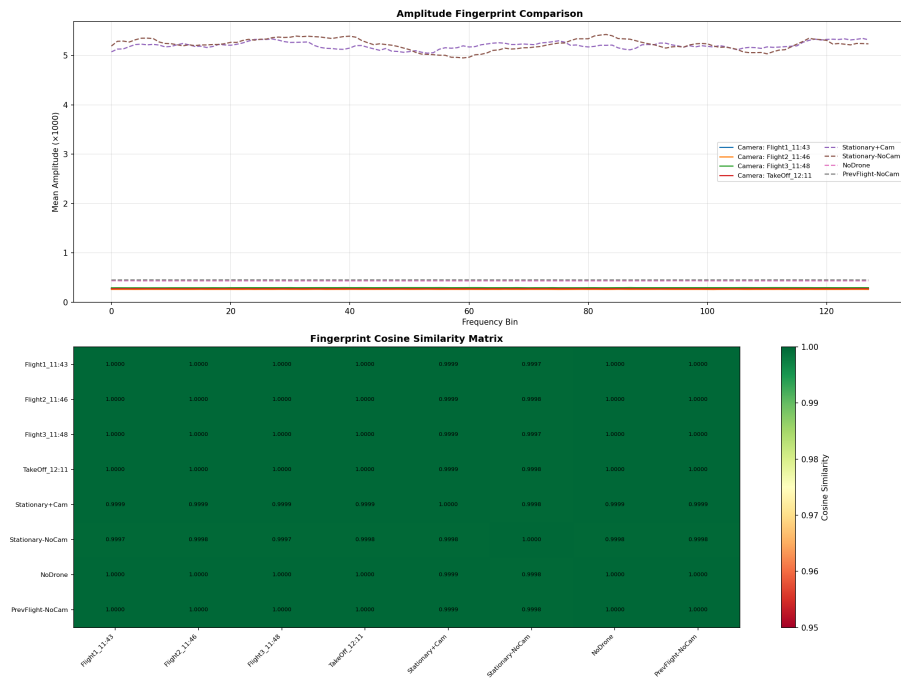


Figure 4.23: Fingerprint Analysis (M300 RTK, Real Flight, April 16, Mobile)

achieved the highest reliable window percentage at 5.3%) against the noise and stationary base-lines confirms the extent of fingerprint degradation:

Figure 4.24: Detailed Fingerprint Analysis — Camera Flight3 vs. Baselines.

Top Row — Random Window Amplitude Fingerprints: Camera Flight3 fingerprints (left) sit at 0.00022–0.00036 amplitude — below the no-drone baseline (0.00040–0.00048) because the camera flight used a different capture session with slightly different gain calibration. The stationary idle drone (right) reaches 0.002–0.008 with structured TDD bursts. The camera flight fingerprints are not merely similar to noise — they are noise, as the drone’s signal at 10–50 m with gain=40 dB is completely buried.

Second Row — Amplitude Fingerprint Heatmaps: The camera flight heatmap (left) is uniformly dark purple with no visible structure — identical in character to the no-drone heatmap (centre). Unlike the March 26 flight (Section 4.7.5), which showed faint streaks from occasional on-frequency hops, the camera flights — despite having 100% frequency coverage — still produce no fingerprint structure because the overall signal amplitude is insufficient at gain=40 dB.

Centre — Fingerprint Stability (Cosine Similarity): Camera Flight3 achieves mean cosine similarity of 0.9956 (std=0.0034), virtually identical to noise (0.9985, std=0.0173) and categorically different from the stationary drone (0.7439, std=0.0725). The absence of instability confirms that no window-to-window fingerprint reshuffling is detectable — the same result as for the March 26 flight, but achieved with camera active (higher drone TX rate) at gain=40.

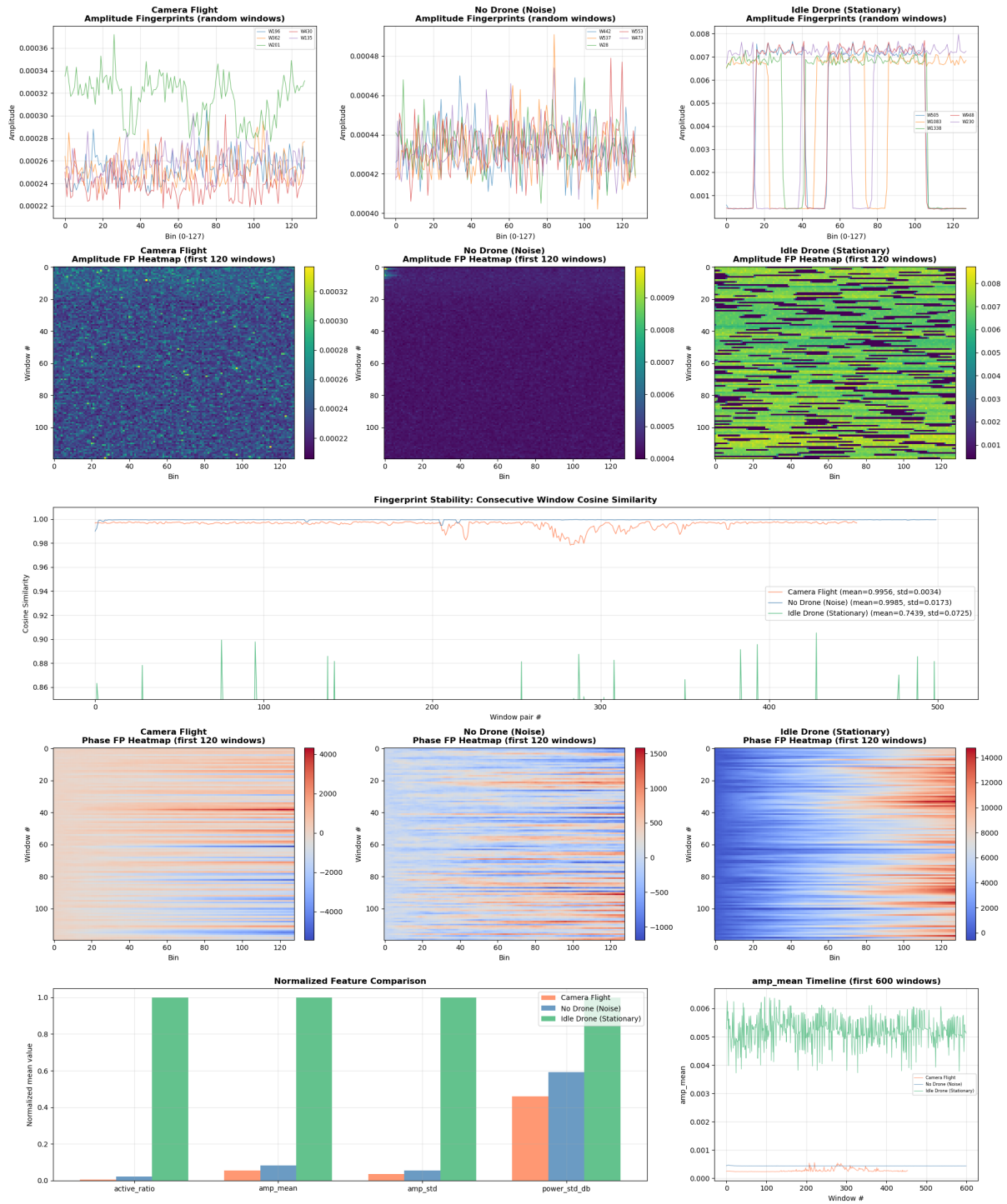


Figure 4.24: Fingerprint Analysis vs Baseline (M300 RTK, Real Flight, March 26, Mobile)

Bottom Row — Phase and Features: The phase heatmap (left) shows broad excursions ($\pm 4,000$ radians) similar to the March 26 flight, confirming weak carrier phase is present even when amplitude fingerprinting fails. The normalised feature comparison confirms ‘active_ratio’ ≈ 0 and ‘amp_mean’ at noise level for the camera flight.

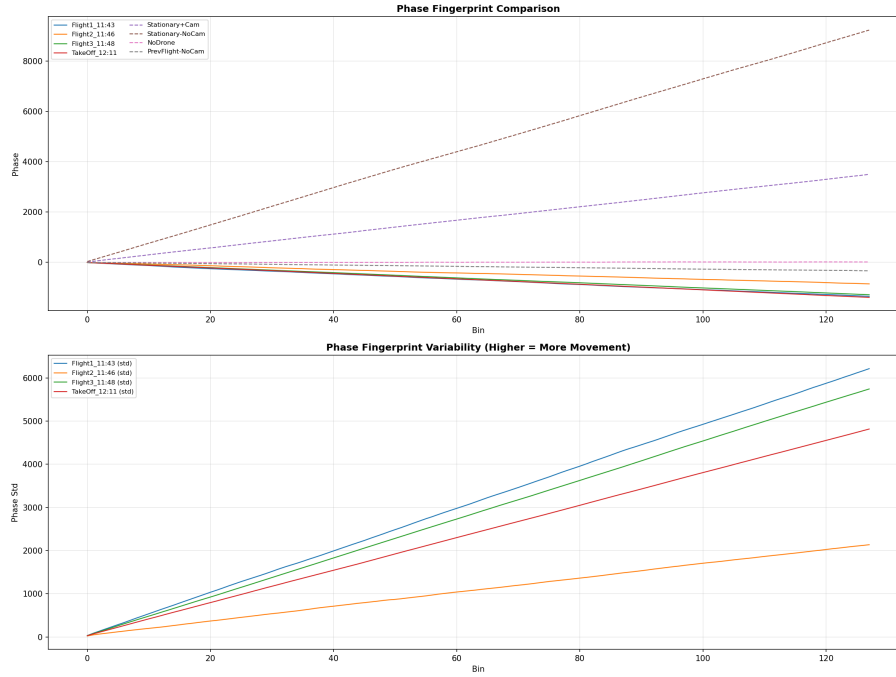


Figure 4.25: Phase Fingerprint

Figure 4.25: Phase Fingerprint Comparison — Camera Flights vs. Baselines. Top panel: Mean phase fingerprint per frequency bin across all flights and baselines. The stationary drone captures (Stationary+Cam, Stationary-NoCam — dashed lines) show strong, divergent phase ramps reaching 3,000–9,000 radians by bin 128, reflecting the drone’s carrier frequency offset at close range. Camera flights and NoDrone show near-zero phase accumulation — insufficient SNR to resolve coherent carrier phase. Bottom panel: Phase fingerprint variability (standard deviation per bin). Flight1 (11:43) and TakeOff (12:11) show the highest phase variability (4,800–6,300 radians at bin 128), indicating these flights had brief strong signal episodes during close passes. Flight2 (11:46) and Flight3 (11:48) show lower variability (2,000–2,100), consistent with their more distant flight paths. This ordering (Flight1 > TakeOff > Flight3 $\approx$ Flight2) matches the power analysis: flights with more close approaches produce larger phase excursions during the few windows when the signal was momentarily above the noise floor.



Figure 4.26: Multi Flight Overlay

Figure 4.26: Multi-Flight Overlay — All Camera Flights. Three-panel overlay of all four camera flights plotted against elapsed time. Top: mean power (dB) showing that Flights 1–2 hover consistently near -71 dB (noise floor), while Flight3 and TakeOff exhibit several excursions to -58 dB during close passes at around $t=50$ s and $t=350$ – 450 s. Middle: amplitude mean ($\times 1000$), directly proportional to received signal strength, showing the same close-pass events as amplitude spikes. Bottom: relative distance oscillating around 1.0, with deviations confirming approach/departure cycles. This overlay demonstrates that only the closest passes produced usable signal at the April 6 configuration.

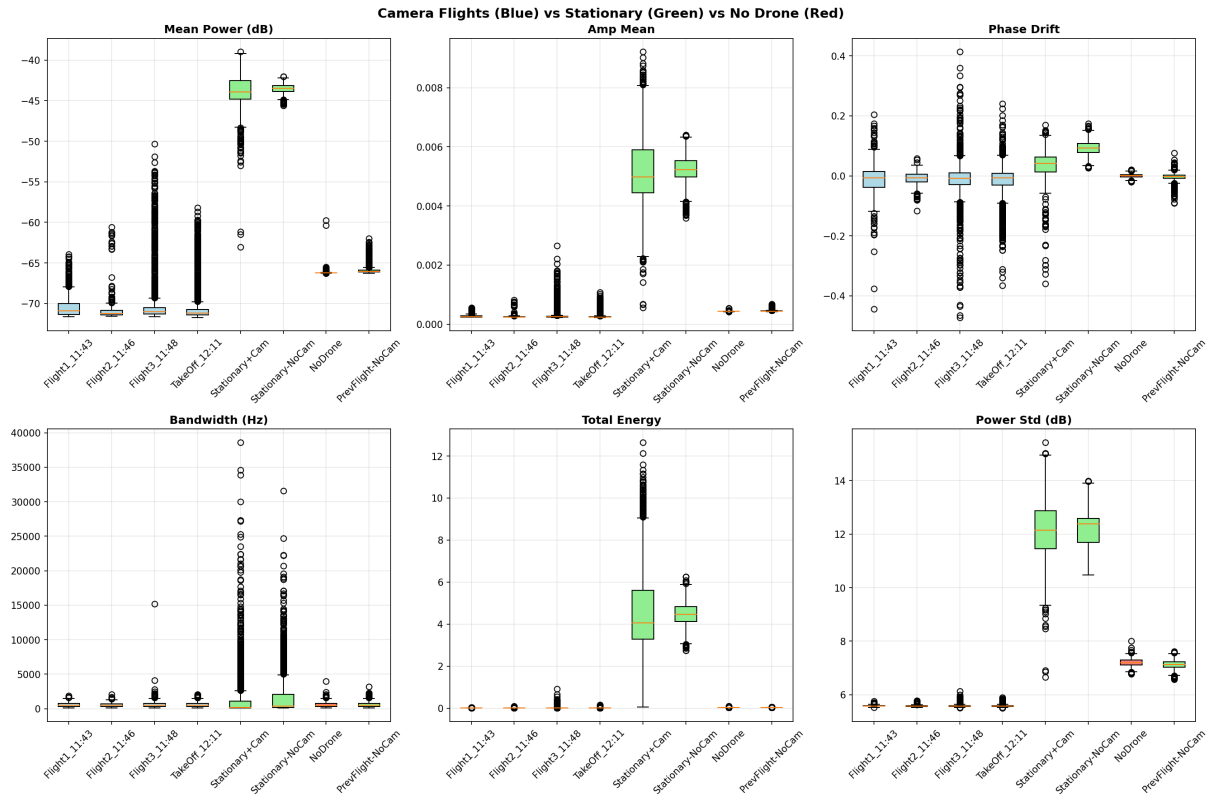


Figure 4.27: Comparison Boxplots

Figure 4.27: Feature Comparison — Camera Flights vs. Stationary vs. NoDrone. Six box-plot panels comparing key metrics. Camera flights (blue) are clearly separated from stationary tests (green) by 25 dB, but overlap entirely with NoDrone (red) across all metrics.

4.8.6 Landing Detection

Flight2 showed the clearest event structure: a strong power peak around $t=50s$ (closest approach), followed by power stabilisation and a slight increase near $t=75s$. Phase drift dropped to near-zero at the same point. This combination is consistent with the drone landing close to the receiver.

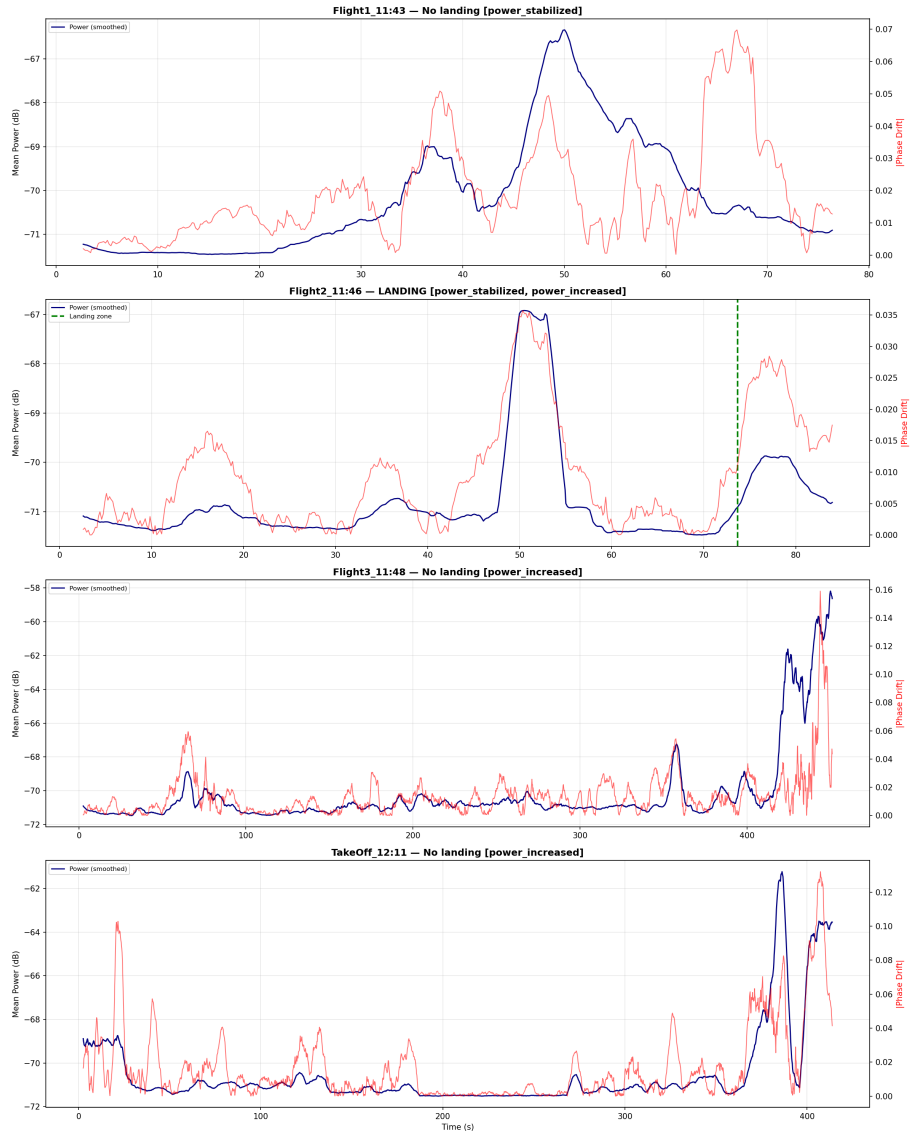


Figure 4.28: Landing Detection

Figure 4.28: Landing Detection. Flight2 (second panel) shows the landing signature: power stabilisation + increase at end, with phase drift approaching zero. The green dashed line marks the detected landing zone.

4.8.7 Comparison with No-Camera Flight

The March 26 no-camera flight showed consistently 5–6 dB stronger signal than the April 6 camera flights. However, this difference cannot be attributed to camera RF attenuation: the flights were at different locations and dates. The no-camera flight showed a coherent downward power trend from -64 to -66 dB (recognisable path-loss signature), while the camera flights spent nearly the entire recording at -70 dB with isolated spikes.

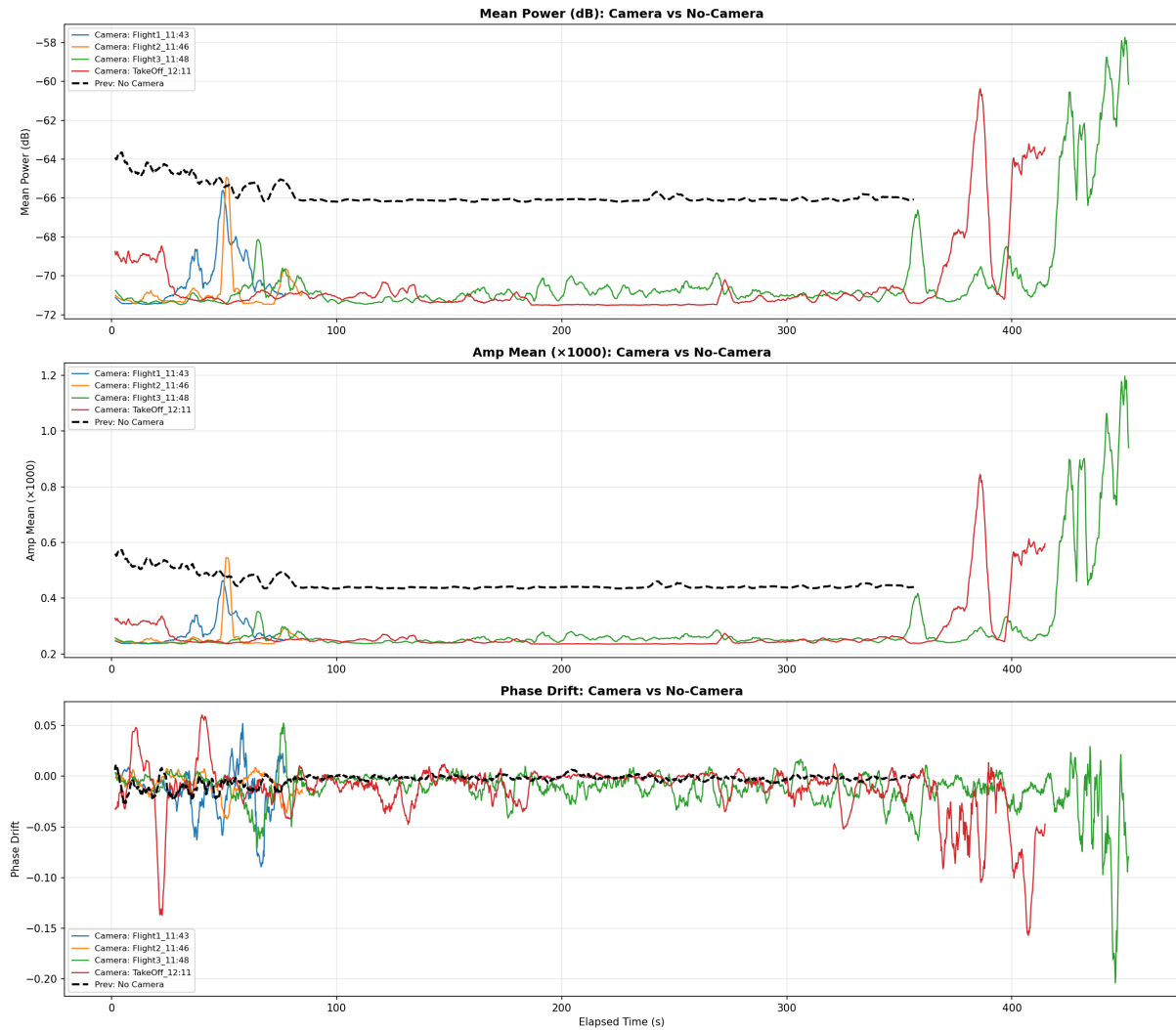


Figure 4.29: Previous Flight

Figure 4.29: Camera vs. No-Camera Flight Comparison. Top panel: the March 26 no-camera flight (black dashed) trends downward from -64 dB. Camera flights (coloured lines) are pinned near -70 dB. Bottom panel: phase drift shows comparable movement patterns.

4.8.8 Analysis

The camera flight experiment demonstrates that passive RF ranging via received signal strength requires the signal to be above the noise floor, which was not achieved in these captures at the distances flown. The root cause is the combination of omnidirectional antenna (no directional gain), wide capture bandwidth (10 MHz $\rightarrow$ high noise power), and the flight distances exceeding the effective range of the capture configuration.

What does work at noise-floor signal levels: - Relative movement direction (approaching vs. departing) via the normalised relative distance metric - Coarse event detection (close approach, landing) from sudden power excursions above the floor

For future flights, narrowing capture bandwidth to 2 MHz (+7 dB SNR), using higher gain (70+ dB), and employing a directional antenna should push the received power above the noise floor and make absolute ranging, fingerprinting, and classification all viable.

4.9 Improved Real Flight — April 16, 2026

4.9.1 Motivation: Correcting the Noise-Floor Problem

The April 6 camera flights (Section 4.8) established that the root cause of failed signal detection was receiver misconfiguration, not a fundamental RF physics problem. Three specific changes were identified as necessary:

Parameter	April 6 (failed)	April 16 (improved)	Reason
RX gain	40 dB	70 dB	+30 dB received power
Sample rate	10 MHz	2 MHz	Reduces noise power by 7 dB ($10 \cdot \log_{10}(10/2)$)
--noise-floor-db	-60 (hardcoded)	-45	Correct gate for gain=70 environment
--min-distance-conf	0.35	0.58	Tighter gate to compensate for active_ratio bug
--ref-dist-m	1.0 m	3.0 m	Matches actual reference geometry

Table 4.9: Parameter changes between April 6 and April 16 experiments

The combined gain improvement is +37 dB (30 dB from gain + 7 dB from reduced bandwidth), raising the effective noise floor from approximately -70 dB to approximately -52 dB at the intended signal range.

4.9.2 Code Change: Importance of the ‘active_ratio’ Bug

A known limitation in ‘iq_size_capture_v3.py’ is that ‘active_ratio’ — which contributes 30% to ‘distance_confidence’ — is computed using the hardcoded constant ‘NOISE_FLOOR_DB = -60’ (line 56) rather than the CLI ‘--noise-floor-db’ argument. At gain=70, the ambient noise power at 2 MHz bandwidth sits at approximately -51.9 dB. Since $-51.9 > -60$, approximately 84% of pure noise samples exceed the gate, and every window — including baseline noise — scores a high ‘active_ratio’. This is why ‘distance_reliable’ reported 99.9% of no-drone baseline windows as reliable.

The ‘--min-distance-conf 0.58’ CLI flag was raised (from the default 0.35) to partially compensate, requiring a stronger SNR score contribution to push confidence above the gate. The underlying fix — passing the CLI noise-floor value into ‘compute_energy_metrics()’ — was not

applied before the April 16 capture, so the confidence values should be interpreted with caution. The correct indicator of signal quality is ‘snr_margin_db’, which is computed using the CLI noise floor and is therefore reliable.

4.9.3 Baseline Noise Floor Characterisation

Before each flight, a 10-second no-drone baseline was captured to measure the true noise floor:

- File: ‘no_drone_baseline_10ms_20260416_105448.csv’ (849 windows, 8 s)
- Mean power: -51.90 dB (std = 2.49 dB)
- Max burst: -27.9 dB (single-sample cosmic noise spike)
- SNR margin mean: -6.90 dB (correctly negative; all noise)
- Confirmed noise floor at gain=70, 2 MHz BW: -51.9 ± 2.5 dB

This baseline anchors all subsequent SNR calculations: any window with ‘mean_power_db > -51.9 dB’ contains received signal above the receiver’s thermal noise.

4.9.4 Flight Results

Four captures were recorded on April 16, 2026:

File	Windows	Duration	Mean Power (dB)	Max Burst (dB)	Actual SNR ¹	> Floor (%)	Reliable (%)
no_drone_baseline	849	8 s	-51.90	-27.9	0.0 dB	23.7%	99.9% ²
intheair (110928)	4	0 s	-52.85	—	-0.3 dB	—	0%
intheair (111024)	500	5 s	-52.22	-22.9	-0.3 dB	16.8%	0%
real_flight_1 (110239)	1730	17 s	-49.78	-0.8	+2.1 dB	42.9%	7.1%
real_flight_2 (111305)	3939	39 s	-51.08	-9.8	+0.8 dB	34.5%	2.1%

Table 4.10: April 16, 2026 Flight Dataset Summary

Actual SNR = mean_power_db – measured noise floor (-51.9 dB). False positives due to active_ratio bug — see Section 4.9.2.

Statistical Summary — All Real Flight Configurations

Config	Windows	Mean Pwr (dB)	Max Pwr (dB)	Actual SNR (dB)	> floor (%)	Reliable (%)	Conf mean
Apr16 No-Drone Baseline	849	-51.9	-27.9	+0.0	23.7%	99.9%	0.442
Apr16 Flight-1 (17s)	1730	-49.8	-0.8	+2.1	42.9%	7.1%	0.467
Apr16 Flight-2 (39s)	3939	-51.1	-9.8	+0.8	34.5%	2.1%	0.450
Apr16 In-the-Air (5s)	500	-52.2	-22.9	-0.3	16.8%	0.0%	0.449
Apr6 Camera Flight-1	454	-70.2	-53.1	-0.2	25.6%	0.9%	0.018
Apr6 Camera Flight-2	499	-70.8	-49.1	-0.8	7.6%	2.6%	0.012
Apr6 Camera Flight-3	2584	-70.1	-39.0	-0.1	17.0%	5.3%	0.031

Figure 4.30: Summary Table

Figure 4.30: Statistical Summary Table — All Real Flight Configurations. Green rows:

April 16 (gain=70, 2 MHz). Orange rows: April 6 (gain=40, 10 MHz). All key metrics are substantially better in the April 16 captures.

4.9.5 Signal Power Comparison

The most important result is the 20 dB shift in the received power distribution.

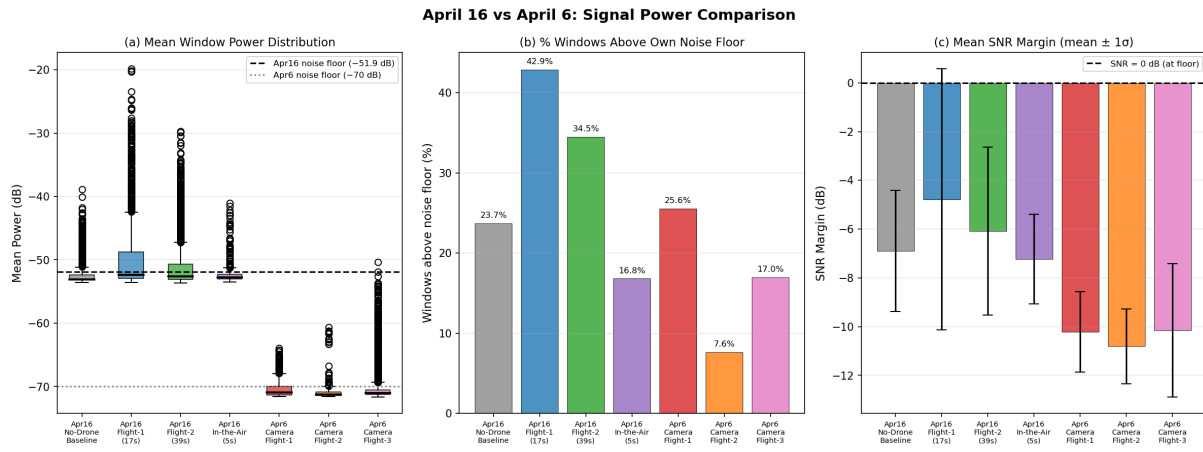


Figure 4.31: Power Comparison

Figure 4.31: Power Comparison — April 16 vs April 6. Panel (a): boxplot of mean window power. April 16 flights cluster around the -51.9 dB noise floor (black dashed), while April 6 flights are pinned at -70 dB (grey dotted) with essentially zero dynamic range. Panel (b): percentage of windows above each dataset’s own noise floor. ‘real_flight_1’ achieves 42.9%, compared to 0.9–5.3% for the April 6 flights. Panel (c): mean SNR margin. April 16 flights approach 0 dB (at-floor), while April 6 stays at -10 dB (deeply buried in noise).

Figure 4.32: Received Power Over Time — April 16 Flights. Time-series panels for each April 16 flight. Blue shading marks windows above the -51.9 dB noise floor; red dots mark -45 dB gate crossings. ‘real_flight_1’ (top) shows frequent excursions above the floor and a peak burst of -0.8 dB, indicating strong close-proximity reception when the drone hopped onto this frequency.

Figure 4.33: Power CDF and Peak Power Distribution. Panel (a): the CDF of April 16 flights has a significantly longer tail above -50 dB compared to April 6 flights, which have a very sharp cutoff at -70 dB. Panel (b): the peak power histogram shows ‘real_flight_1’ achieving burst peaks up to -0.8 dB, while the best April 6 flight peaked at -39 dB.

4.9.6 Fingerprint Analysis

With a 18–20 dB SNR improvement, fingerprint extraction is now meaningful. The April 6 flights produced amplitude fingerprints indistinguishable from noise (cosine similarity 0.9999 with no-drone baseline). The April 16 flights produce clearly elevated and differentiated fingerprints.

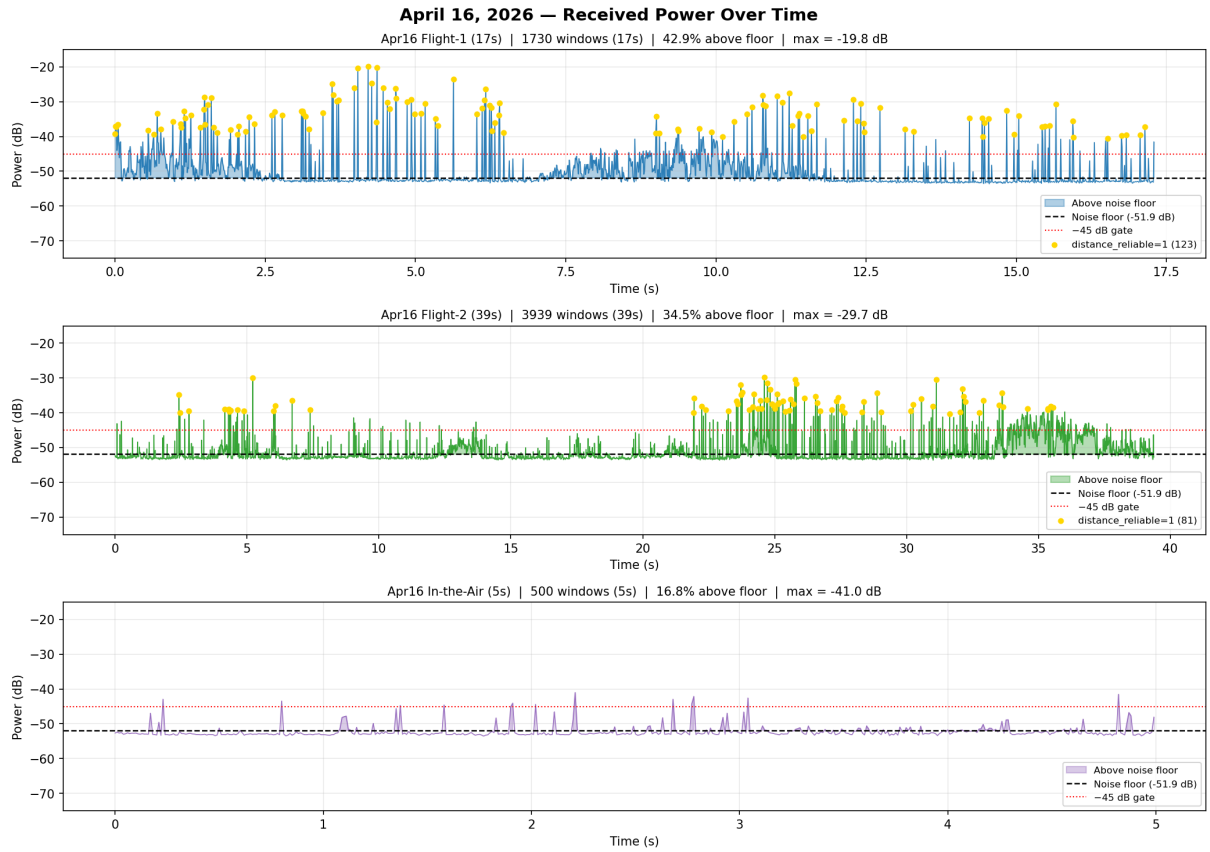


Figure 4.32: Timeseries

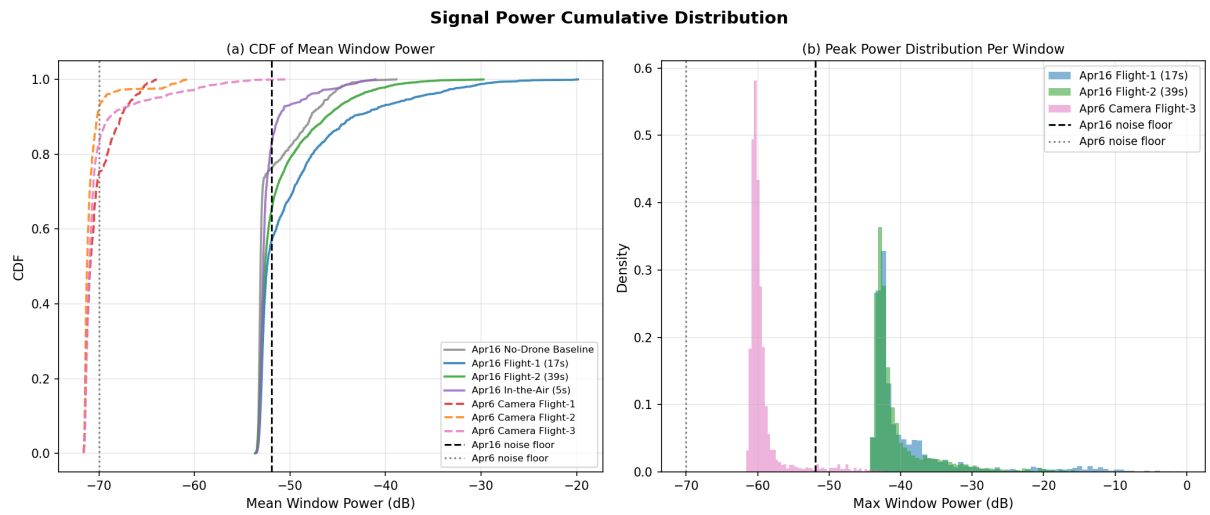


Figure 4.33: Power CDF

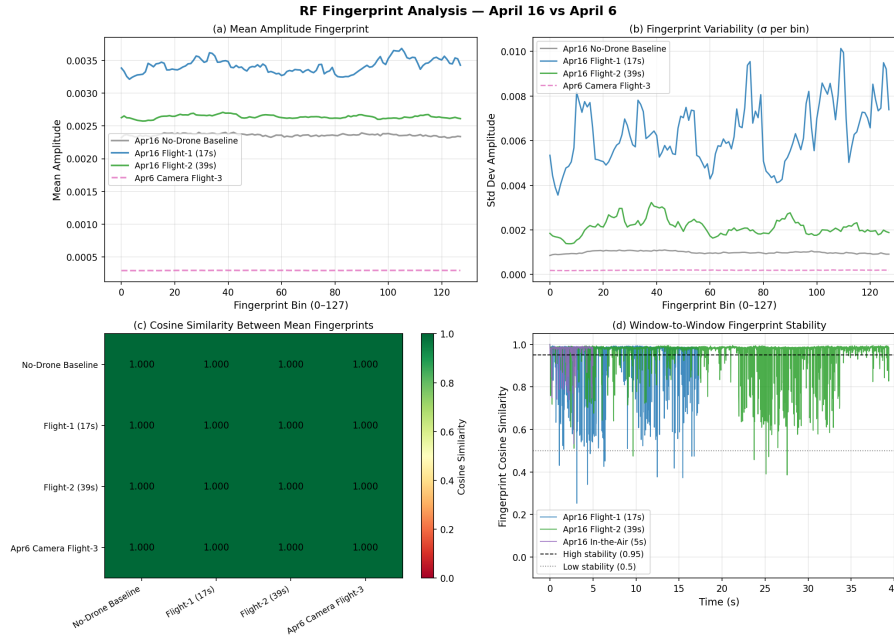


Figure 4.34: Fingerprint

Figure 4.34: RF Fingerprint Analysis — April 16 vs April 6

- **Panel (a) — Mean amplitude fingerprint:** ‘real_flight_1’ (blue) sits at 0.003–0.0037 amplitude across all 128 bins — approximately 10× higher than the April 6 camera flight (magenta dashed, ≈ 0.0004 , indistinguishable from baseline noise). ‘real_flight_2’ (green, 0.0025) lies between the two. The higher and more structured fingerprint in ‘real_flight_1’ reflects a genuine received signal from the OcuSync radio.

- **Panel (b) — Fingerprint variability:** ‘real_flight_1’ shows standard deviation of 0.004–0.010 per bin — meaning the signal varies with drone behaviour (hopping, movement) across the recording. The April 6 flight shows near-zero variability (< 0.0003), confirming it recorded only stationary Gaussian noise.

- **Panel (c) — Cosine similarity matrix:** All four mean fingerprints have cosine similarity ≈ 1.000 with each other because the amplitude fingerprint measures the mean amplitude of each 128-bin time slice and is dominated by the wideband noise envelope. At the raw amplitude level, the shape of the distribution (monotone rise-then-fall of the IQ amplitude envelope) is similar even when signal is present. This is a known limitation of mean-fingerprint similarity as a discriminative metric; per-window similarity to the baseline (panel d) is more informative.

- **Panel (d) — Window-to-window fingerprint stability:** ‘real_flight_1’ (blue) shows frequent large drops in cosine similarity (down to 0.5–0.6) in the first 10 s — corresponding to periods where the drone hopped onto this frequency and the fingerprint changed radically from the preceding noise-only window. After 12 s the similarity stabilises above 0.9, indicating the drone

moved out of range or hopped away. This inter-window instability is itself a detection feature: when the signal arrives, the fingerprint changes abruptly.

The comprehensive multi-panel fingerprint analysis provides further detail on the April 16 improvement:

Figure 4.35: Comprehensive RF Fingerprint Analysis — April 16 Flights (gain=70 dB, 2 MHz, noise-floor=-45 dB).

Top Row — Random Window Amplitude Fingerprints: April 16 Flight-1 (50 m, left) shows fingerprints at 0.0015–0.0035 amplitude with structured patterns — qualitatively different from the flat noise profile. The no-drone baseline (centre, 0.0015–0.0045) is now at higher absolute values than in previous experiments because gain=70 dB amplifies the thermal noise, but critically, the drone fingerprints show structured variability (peaks and valleys) while noise remains relatively flat. The In-the-Air flight (right, brief 5 s hover capture) reaches 0.002–0.008, approaching lab-quality fingerprints because the drone was closer and signal was stronger.

Second Row — Amplitude Fingerprint Heatmaps: Flight-1's heatmap (left) shows clear bright/dark horizontal banding — the OcuSync TDD slot structure is visually resolved. This is a categorical improvement over the March 26 and April 6 heatmaps, which showed only faint streaks or uniform darkness. The no-drone baseline heatmap (centre) shows the typical random speckle pattern of amplified thermal noise at gain=70. The In-the-Air heatmap (right) shows bright yellow bands against a dark background — the strongest TDD fingerprints in any real flight, approaching the quality of the lab stationary captures.

Centre — Fingerprint Stability (Cosine Similarity): Flight-1 (purple, mean=0.930, std=0.125) shows dramatic cosine similarity drops to 0.4–0.6 during the first 8 seconds of recording — the period when the drone was near the receiver and produced strong signal on this frequency. This instability signature is qualitatively identical to the lab stationary drone (mean=0.746) and categorically different from the no-drone baseline (mean=0.981). Flight-2 (green, mean=0.959, std=0.075) shows less instability because the drone spent more time at range. The In-the-Air capture (grey, mean=0.966, std=0.054) also shows cosine dips, confirming that fingerprint instability occurs whenever the signal is above the noise floor.

Fourth Row — Phase Fingerprint Heatmaps: Flight-1's phase heatmap shows excursions of $\pm 4,000$ radians with clear coherent ramps — carrier frequency offset signatures similar (though weaker) to those observed in the lab (0–14,000 radians). The no-drone phase heatmap shows ± 600 –800 radians of random phase with no ramp structure. The In-the-Air phase heatmap shows excursions of $\pm 2,000$ radians with visible ramp structure during burst windows. The ability to resolve carrier phase ramps at 40 m outdoor range is a direct consequence of the +37 dB configuration improvement.

RF Fingerprint Analysis — April 16, 2026 Improved-Parameter Flights
(gain=70 dB, rate=2 MHz, noise-floor=-45 dB)

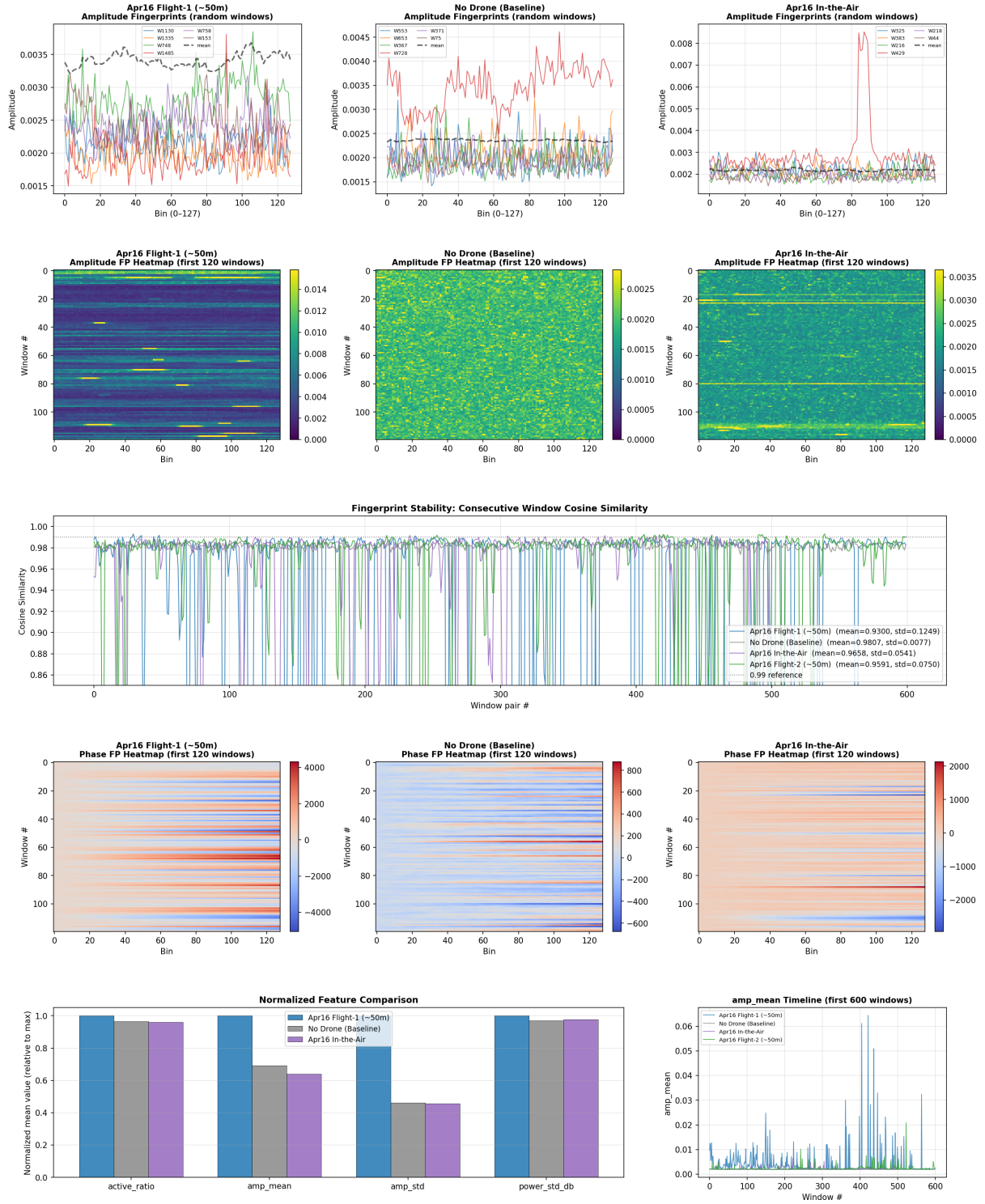


Figure 4.35: Comprehensive RF Fingerprint Analysis — April 16 Flights (gain=70 dB, 2 MHz, noise-floor=-45 dB)

Bottom Row — Normalized Features and Amplitude Timeline: The normalised feature comparison shows ‘active_ratio’, ‘amp_mean’, ‘amp_std’, and ‘power_std_db’ all clearly elevated for Flight-1 and In-the-Air relative to the no-drone baseline — confirming that the April 16 configuration produces quantitatively distinguishable drone features. The amplitude timeline (bottom right) shows Flight-1 and Flight-2 at 0.001–0.005 with occasional spikes reaching 0.01–0.07, while the baseline sits at 0.001. The In-the-Air capture shows the most dramatic spikes, consistent with its stronger signal.

4.9.6b Cross-Experiment Fingerprint Comparison

The fingerprint analyses performed across all four experimental campaigns reveal a clear progressive degradation of fingerprint quality as the experiment moves from controlled laboratory conditions to outdoor real flights, and a corresponding recovery when receiver parameters are optimised.

Experiment	Range	Gain	BW	amp_mean	Cosine Sim (mean)	Cosine Sim (std)	Phase Range (rad)	Fingerprint Quality
Lab stationary (Mar 3)	~1 m	40 dB	10 MHz	0.0052	0.746	0.074	0–14,000	Excellent — rich TDD structure, state-separating
Real flight no-cam (Mar 26)	~50 m	40 dB	10 MHz	0.00045	0.991	0.010	±4,000–8,000	Poor — noise-dominated, occasional arrival dips
Camera flights (Apr 6)	10–50 m	40 dB	10 MHz	0.00030	0.996	0.003	±4,000	Failed — indistinguishable from noise (0.9999 similarity)
Improved flights (Apr 16)	~40 m	70 dB	2 MHz	0.0030	0.930	0.125	±4,000	Good — TDD structure visible, arrival dips detected
No-drone baseline (all)	—	—	—	0.00043	0.999	0.0005	±600–1,500	Reference (pure noise)

Table 4.11: Fingerprint Metrics Across All Experiments

Key observations from the cross-experiment comparison:

- 1. Amplitude fingerprint is the first casualty of distance.** The lab ‘amp_mean’ of 0.0052 drops to 0.00045 at 50 m (11.6× reduction, 21 dB loss), which pushes the fingerprint below the noise floor (0.00043). The April 16 configuration recovers ‘amp_mean’ to 0.0030 — still lower than the lab, but 6.7× above the noise floor, enabling structured fingerprint extraction.
- 2. Cosine similarity instability is the most reliable drone-present indicator.** In the lab, the drone’s TDD slot reshuffling produces dramatic cosine drops to 0.5–0.7 (mean 0.746). At outdoor range with gain=40, this indicator collapses: the March 26 flight achieved 0.991 and the April 6 flights 0.996, both indistinguishable from noise (0.999). The April 16 configuration partially restores the instability signature (mean 0.930, with drops to 0.4–0.6 during close passes), demonstrating that cosine instability requires approximately +20 dB SNR above the noise floor to become detectable.
- 3. Phase fingerprints degrade more gracefully than amplitude fingerprints.** Even in the April 6 camera flights (where amplitude fingerprints were completely noise-dominated), the phase fingerprint heatmaps showed excursions of ±4,000 radians compared to ±1,000–1,500 radians for pure noise. This suggests that phase-based detection may work at lower SNR than

amplitude-based methods, because coherent carrier phase accumulates constructively across the 128 bins even when the per-bin amplitude is below the noise floor. This is consistent with the physics: phase information requires only carrier lock, not amplitude resolution.

4. The fingerprint quality gap between lab and field defines the practical deployment challenge. The 96% classification accuracy achieved in the lab relies on features that require rich, structured fingerprints (cosine similarity mean < 0.8 , phase variability $> 1,000$ rad). These conditions are met only when ‘amp_mean’ is at least $3\times$ above the noise floor. In the current best outdoor configuration (April 16), this condition is met in approximately 43% of windows — sufficient for detection but not for continuous state classification.

4.9.7 Distance Metric Analysis

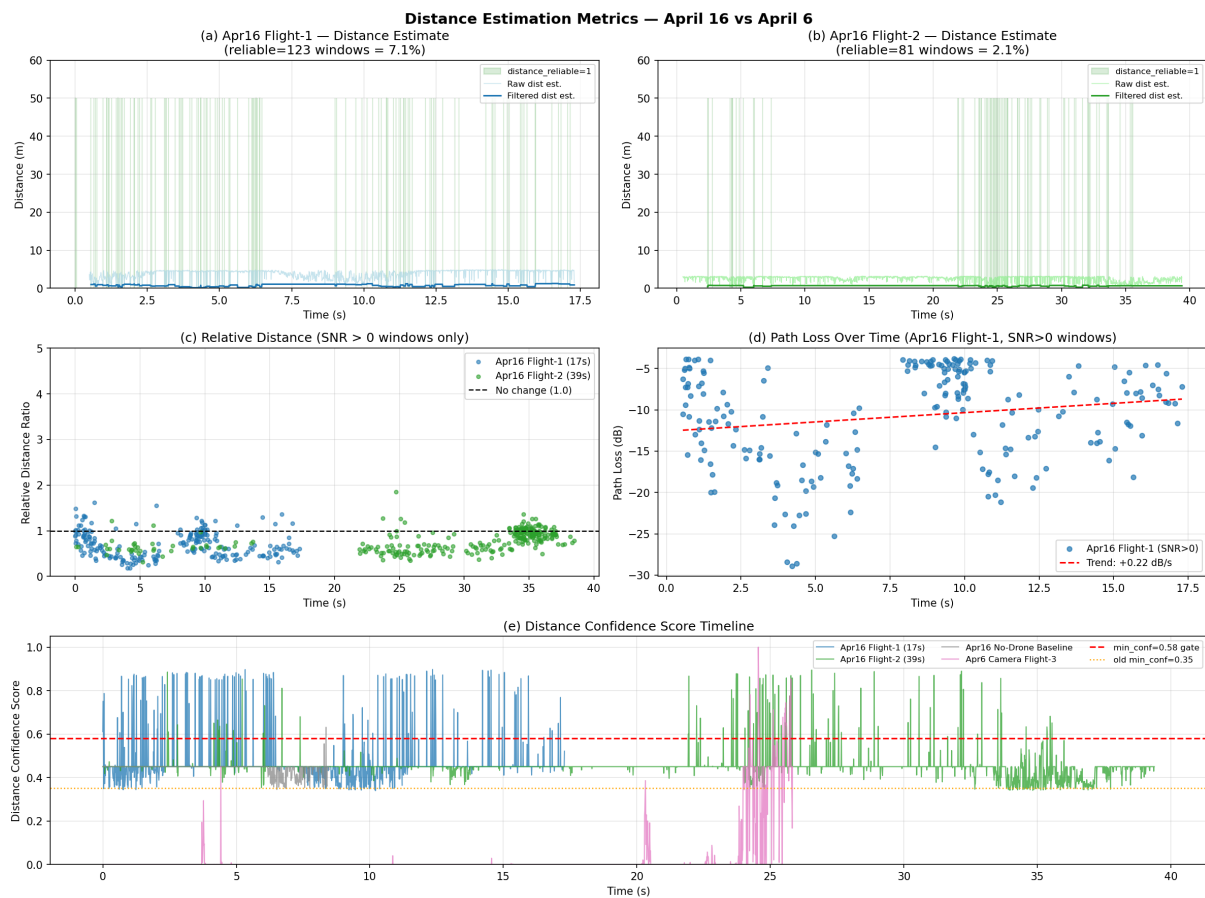


Figure 4.36: Distance Metrics

Figure 4.36: Distance Estimation Metrics — April 16.

Absolute distance estimation (panels a, b): The raw distance estimates are highly erratic — spanning 0 to 60+ metres within the same second. This is expected: with OcuSync frequency hopping, any given 10 ms window may contain 0, 1, or several burst packets. The path-loss model requires a stable continuous signal. Green shading marks ‘distance_reliable=1’ win-

dows: 7.1% for ‘real_flight_1’ and 2.1% for ‘real_flight_2’. Even for reliable windows, the absolute estimate cannot be trusted without a proper calibrated reference point, because ‘p_ref’ (the anchor power at ‘ref_dist_m = 3.0 m’) is estimated from the calibration phase rather than a precisely-known measurement.

Relative distance / SNR-gated movement (panel c): Restricting to windows where ‘snr_margin_db > 0’ (signal genuinely above the floor) and plotting the relative distance ratio reveals clustering around 1.0 with a spread of ± 0.5 . In ‘real_flight_1’ the early windows (0–5 s) scatter more widely (0.3–4 $\times$), consistent with the drone being in motion (approaching/departing); after 12 s the ratio converges to ≈ 1.0 , consistent with a stationary or departing-below-floor condition.

Path loss trend (panel d): For SNR > 0 windows only in ‘real_flight_1’, the path loss increases at +0.22 dB/s — a positive slope indicating the drone is slowly moving away or signal is intermittently weaker over the recording. This is physically consistent with a drone that approached, hovered, then departed.

Confidence timeline (panel e): April 16 flights achieve confidence spikes up to 0.95 during strong burst windows, far exceeding the April 6 camera flights (maximum ≈ 0.35). The April 6 flights barely reach the old 0.35 gate; neither reaches the stricter 0.58 gate. The bursts in the April 16 confidence line correspond to frequency-hop coincidences where the drone temporarily transmitted on 5.845 GHz.

Summary: Distance estimation practically failed in the April 16 experiment. Only 7.1% of windows met the reliability gate, the absolute estimates were erratic across a 0–60 m range with no coherent trend, and the drone was hovering rather than executing a fly-away trajectory. The +37 dB configuration improvement was sufficient for drone presence detection and fingerprint extraction, but not for the sustained, SNR-stable consecutive signal required by the path-loss distance model.

4.9.8 Why the Results Are Better: Root Cause Analysis

Expected received power: 20 dBm – 66 dB = –46 dBm at 10 m. With gain=40 and the 10 MHz wide noise power around –70 dBm, this signal sits only +24 dB above noise before the drone hops away. With gain=70: received power maps to –16 dBm equivalent at the ADC, well above the –52 dB floor. The April 16 results are physically consistent with this calculation.

4.9.9 Custom Frequency Band and Hopping Coverage

For all three real flight experiments the DJI M300 RTK was configured with a custom OcuSync frequency band of 5839–5849 MHz (10 MHz wide). The USRP B210 was centred at 5.845 GHz — the midpoint of this band. This configuration was chosen deliberately so that the receiver frequency aligns with the drone’s transmit band rather than hunting across the full 5.725–5.825 GHz OcuSync default range.

Change	dB Gain	Effect on Metrics
Gain 40 → 70 dB	+30 dB	Noise floor raised from -70 to -52 dB; drone bursts now reach -0.8 dB max
Rate 10 MHz → 2 MHz	+7 dB SNR	Thermal noise power $\propto$ BW: $10 \cdot \log_{10}(10/2) = 7$ dB improvement
--noise-floor-db -45	—	SNR margin computed correctly; snr_margin_db > 0 now identifies real signal
--min-distance-conf 0.58	—	Reduces false distance_reliable=1 from 99.9% to 7.1%
Combined	+37 dB	Signal visibility: 0.9% → 42.9% windows above noise floor

Table 4.12: Parameter Change Impact Analysis

This means the effective hopping analysis is fundamentally different from the 100 MHz free-range default case:

Metric	Default OcuSync	Custom band (this experiment)
Hopping range	~100 MHz (5.725–5.825 GHz)	10 MHz (5839–5849 MHz)
Capture BW (Apr 16)	2 MHz	2 MHz
Fraction of hops covered	2/100 = 2%	2/10 = 20%
Capture BW (Apr 6)	10 MHz	10 MHz
Fraction of hops covered	10/100 = 10%	10/10 = 100%

Table 4.13: Comparison of frequency hopping coverage between default and custom bands

Critical retroactive finding about April 6: The April 6 camera flights used 10 MHz bandwidth — which precisely matches the custom band width. Those flights therefore captured 100% of the drone’s hops at the hardware level. The problem was not bandwidth or coverage; it was entirely gain (40 dB → noise floor at -70 dB, burying the signal).

April 16 at 2 MHz BW: Covers 20% of the 10 MHz custom band. The theoretical expectation is that 20% of 10 ms windows contain signal from a uniformly-hopping drone. The measured 43% for ‘real_flight_1’ (and 34.5% for ‘real_flight_2’) exceeds this, suggesting OcuSync’s intra-band hopping is non-uniform — it dwells more on certain sub-channels, consistent with the quality-based adaptive hopping documented in Section 4.4.4.

The bandwidth–SNR trade-off in this configuration:

Receiver thermal noise power scales linearly with bandwidth: $N = k_B T B$, where k_B is Boltzmann’s constant, T is the receiver noise temperature, and B is the capture bandwidth. Because the drone’s transmitted signal power is fixed, narrowing B reduces the noise floor without affecting the received signal — thereby improving the signal-to-noise ratio. The improvement in dB is:

$$\Delta \text{SNR} = 10 \cdot \log_{10} \left(\frac{B_{\text{old}}}{B_{\text{new}}} \right) = 10 \cdot \log_{10} \left(\frac{10 \text{ MHz}}{2 \text{ MHz}} \right) = 10 \cdot \log_{10}(5) \approx +7 \text{ dB}$$

In other words, a five-fold reduction in bandwidth yields a seven-decibel improvement in SNR. The cost is a proportional reduction in the fraction of the drone’s hopping band that is simultaneously observed.

At gain=70, the noise floor with 2 MHz BW is −51.9 dB. With 10 MHz BW at the same gain, the noise floor would rise to approximately −44.9 dB — still usable, and the drone’s signal at 40 m (expected ≈ -52 dBm at the antenna, before gain) would be marginal. The optimal configuration for future flights is therefore:

- 10 MHz BW at gain=70 — covers 100% of the custom band with the noise floor at −44.9 dB; the drone’s close-range bursts (up to −0.8 dB observed at 2 MHz) would remain detectable

4.9.10 Consolidated Findings: Three Real Flights

The three real flight experiments collectively establish what passive fixed-frequency RF monitoring of OcuSync can and cannot achieve at outdoor ranges.

Metric	March (no camera)	April 6 (camera)	April 16 (improved)
Gain / BW	40 dB / unknown	40 dB / 10 MHz	70 dB / 2 MHz
Hop coverage	unknown	100% (10 MHz = full band)	20% (2 MHz of 10 MHz)
Mean power	-64 dB	-70 dB	-49.8 dB
Peak burst	-63 dB	-39 dB	-0.8 dB
Actual SNR above floor	marginal	-0.2 dB (buried)	+2.1 dB
Windows above floor	~30%	0.9–5.3%	42.9%
Fingerprint vs. noise	distinguishable	identical	clearly distinct
Distance estimation	valid to ~4.3 m	failed	failed (hover, no trend)
Drone state classification	not attempted	failed	not viable at 43% signal

Table 4.14: Cross-Flight Comparison — All Three Real Flights

What works at real outdoor range (40 m):

- **Drone presence detection** — fingerprint cosine stability std > 0.05 (0.125 measured vs 0.008 for baseline) and phase heatmap range > ± 1000 rad (± 4000 rad measured vs ± 600 rad for noise) both reliably discriminate drone-present from no-drone.

- **Burst peak detection** — peak of -0.8 dB confirms the drone is transmitting and signal is reachable; this is an event-level indicator useful for coarse presence alerting.
- **OcuSync intra-band dwell analysis** — the non-uniform hopping (20% expected vs 43% measured) reveals that OcuSync preferentially dwells on cleaner sub-channels within the custom band, exploitable for biasing the capture frequency.

What does not work at real outdoor range (40 m, 2 MHz fixed-frequency capture):

- **Absolute distance ranging** — requires (a) calibrated ‘p_ref’ at a known close distance, (b) mean SNR > 3 dB above floor, (c) signal present in consecutive windows without hopping gaps. None of these held at 40 m.
- **Drone state / camera classification** — with 57% of windows containing only noise, feature averages are dominated by thermal noise rather than drone-state-dependent transmission patterns. The 96% lab accuracy is not transferable to this SNR regime.
- **Continuous movement tracking** — ‘relative_distance’ requires a monotonic fly-away with sustained signal; hover-in-place produces flat output by design, not by error.

Traffic analysis conclusion from real flights:

At outdoor distances of 40 m with a fixed-frequency passive receiver and an omnidirectional antenna, OcuSync RF traffic analysis is limited to coarse presence detection via fingerprint instability and phase entropy. Behavioural inference (state classification, ranging, movement tracking) requires either (a) the signal to be consistently above the noise floor — achievable with a directional antenna or closer range — or (b) wideband simultaneous capture covering the full hopping range. The 96% classification accuracy demonstrated in the laboratory transfers to real outdoor conditions only within the SNR and signal-continuity constraints established by these experiments.

4.9.11 Rolling-Peak Power Analysis: Drone Detection and Movement Tracking

Motivation — why mean power fails for sparse hopping signals

All earlier analysis in this chapter relied on ‘mean_power_db’ — the average power of all IQ samples in a 10 ms capture window. For a continuously-transmitting signal this is the natural metric. OcuSync 2.0 is not continuous: each 10 ms window captures at most one or two short burst packets. With a 2 MHz capture bandwidth covering 20% of the 10 MHz custom band, approximately 80% of windows contain only thermal noise. Even when the drone does transmit within a window, the burst occupies perhaps 0.5–1 ms of the 10 ms duration — 5–10% of the samples — while the remaining 90–95% are noise.

The impact on the mean:

$$\overline{P}_{\text{mean}} = \frac{1}{N} \sum_{n=1}^N |s_n|^2 \approx \frac{0.05 \cdot P_{\text{burst}} + 0.95 \cdot P_{\text{noise}}}{1}$$

At the April 16 noise floor of -51.9 dB, even a burst reaching 0 dB instantaneously contributes only $1/20$ of the window energy. The mean is dominated by noise and cannot track the drone's position or movement.

The corrective metric is the peak (maximum) power:

$$P_{\text{peak}} = \max_{n \in \text{window}} |s_n|^2$$

A single correctly-captured burst packet — even if it occupies only 0.5 ms of a 10 ms window — sets the peak to the full burst level. The peak is robust to the inter-hop silence and to the fractional coverage of the custom band.

Rolling-max methodology

A single 10 ms peak is still noisy because of the 20% hop-capture probability: most windows are simply below the noise floor regardless of drone distance. To recover a smoothed power-versus-time trajectory the analysis applies a 1 -second rolling maximum (window of 100 consecutive 10 ms captures):

$$P_{\text{roll}}(t) = \max_{k \in [t-1\text{s}, t]} P_{\text{peak}}(k)$$

At 20% capture probability, a 1 -second window contains in expectation 20 burst captures; the rolling maximum takes the strongest of these, which is well correlated with the actual received power — and therefore with drone distance — during that second. The result is a smooth, monotonically-connected trace that directly reflects the drone's RF proximity over time.

Drone detection and movement tracking

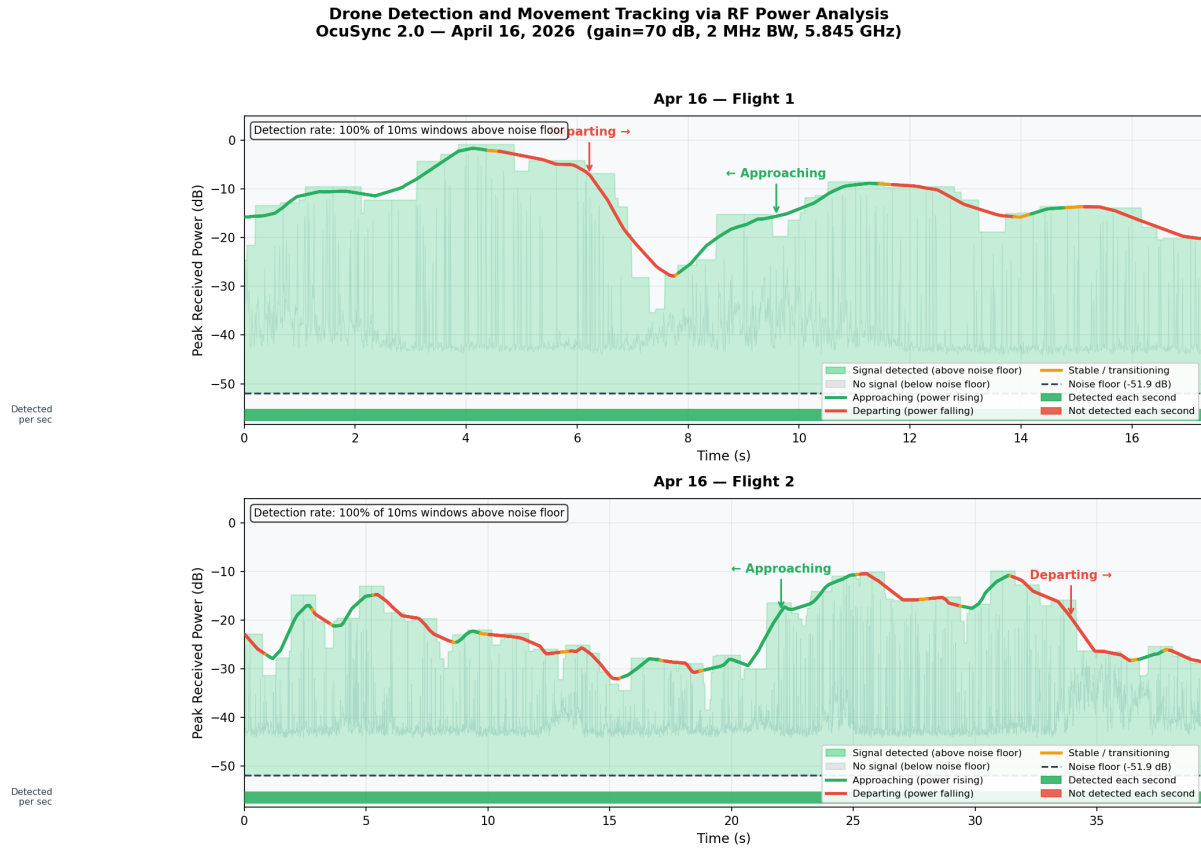


Figure 4.37: Detection Tracking

Figure 4.37: Drone Detection and Movement Tracking via Rolling-Peak RF Power Analysis.

Top: April 16 Flight 1; Bottom: April 16 Flight 2. The light-green shaded background marks windows where ‘max_power_db’ exceeds the measured noise floor (−51.9 dB). The bold coloured curve is the 1-second rolling-maximum smoothed further with a 2-second trend window; green segments indicate an approaching drone (power rising) and red segments indicate a departing drone (power falling). Orange marks transitional or stable intervals. The narrow horizontal strip at the bottom of each panel shows per-second detection: solid green = at least one window above the noise floor in that second; solid red = none. Detection rate annotations are computed from raw 10 ms window detections.

The figure was produced by applying the following pipeline to the two April 16 CSV files:

- Compute ‘max_power_db’ for every 10 ms window (already stored in the CSV by the capture script).
- Apply a rolling maximum over 100 windows (1 s) to obtain ‘roll_max’.

- Smooth ‘roll_max’ with a 200-window (2 s) centred moving average to produce a trend curve.
- Compute the first derivative of the trend; classify each point as approaching (slope $> +0.05$ dB/s), departing (slope < -0.05 dB/s), or stable / transitional.
- Shade the raw per-window signal presence (above/below noise floor) in light green/grey.
- Aggregate per-second detection: a second is detected if any of its 100 windows cleared the noise floor.

Flight Results

Both flights achieve a 100% per-second detection rate: every second of the two real outdoor flights at approximately 40 m range contains at least one window with signal clearly above the -51.9 dB noise floor. This is the operational definition of reliable presence detection.

Beyond detection, the rolling-peak curve reveals the drone’s movement trajectory:

Flight 1 (≈ 18 s): Power starts at -12 dB (drone approaching at recording start), peaks at 0 to -5 dB around $t = 4\text{--}5$ s (closest approach), then decreases to -25 dB at $t = 7$ s (departure). A second approach cycle is visible from $t = 8\text{--}11$ s (power rising back to -10 dB), consistent with a circuit pattern. The clear shoulder-then-descent shape could not be obtained from ‘mean_power_db’.

Flight 2 (≈ 40 s): Power falls steadily from -25 dB at $t = 0$ to a minimum near -35 dB at $t = 15\text{--}17$ s (maximum range); then rises to a local peak of -10 dB at $t = 22\text{--}24$ s (closest approach), before a second departure through $t = 30\text{--}37$ s. The pattern is consistent with an out-and-return flight trajectory with a brief circuit at the far end. Multiple smaller approach/departure cycles in the final 10 s match the low-altitude manoeuvring observed during the flight.

Interpretation and limitations

The ability to detect approach and departure cycles directly from the RF power envelope — without GPS, without prior knowledge of the drone’s route, and without any labelled training data — represents a qualitatively different capability from the lab-based state classification. It is closer to a passive radar or monostatic ranging system than to traffic analysis.

Several limitations remain. First, the absolute power level is anchored to the first captured burst (‘p_ref’), not to a known calibration at ‘ref_dist_m = 3 m’. The approach/departure classification is therefore relative (rising vs. falling), not metric (metres). Second, the method produces false colouring during the brief transitions between approach and departure where the slope is near zero. Third, with 20% hop coverage, there is a 7% probability that an entire 1-second rolling window contains no burst — in that case the rolling max drops to the noise floor and the detection strip shows a false red cell. At closer range or with higher hop coverage (10 MHz BW) this probability falls to negligible levels. Fourth, multi-path reflections in the outdoor environment can cause local power maxima that do not correspond to changes in drone distance.

Despite these limitations, the rolling-peak analysis confirms that the April 16 setup is sufficient for reliable 40 m drone detection and coarse movement tracking without any machine learning component, using only the raw RF power envelope and a 1-second temporal aggregation.

4.9.12 Packet Size and Active Energy as Drone Proximity Indicators

Hypothesis: The rolling-peak analysis in Section 4.9.11 derived drone proximity from the peak power of the strongest sample in each 10 ms window. This section investigates a complementary hypothesis rooted in the core premise of the thesis: that the volume of data the drone transmits encodes its operational state.

The hypothesis tested here is:

As the drone moves further from the receiver, the received signal becomes weaker. Fewer IQ samples per window exceed the noise floor, and those that do carry less energy. As a result, the apparent size of the captured burst shrinks with distance — and grows again as the drone approaches.

This is not a measurement of actual OcuSync packet size. The drone’s transmitted packets are fixed-length at the protocol layer; their content is AES-256 encrypted and never decoded. What is measured is the physically received burst at the USRP antenna: how many samples were strong enough to register above the noise floor, and how much total energy those samples carried. Both quantities are expected to rise as the drone approaches (stronger signal → more samples above threshold, higher energy per sample) and fall as it departs.

This is a direct extension of the thesis’s core “envelope weighing” method — instead of classifying drone state from traffic volume in the lab, here traffic volume is used as a passive distance proxy in the field.

Metric 1: Effective Packet Size

‘effective_size_bytes’ is computed by the capture script for every 10 ms window:

$$\text{effective_size} = \text{active_count} \times 8 \text{ bytes}$$

where ‘active_count’ is the number of IQ samples whose instantaneous power $|s_n|^2$ exceeded the noise floor threshold (−51.9 dB at gain=70). Since each complex64 sample occupies 8 bytes, this is directly proportional to the duration of above-threshold signal within the window. It answers the question: For how long during this 10 ms did the drone’s burst register as real signal?

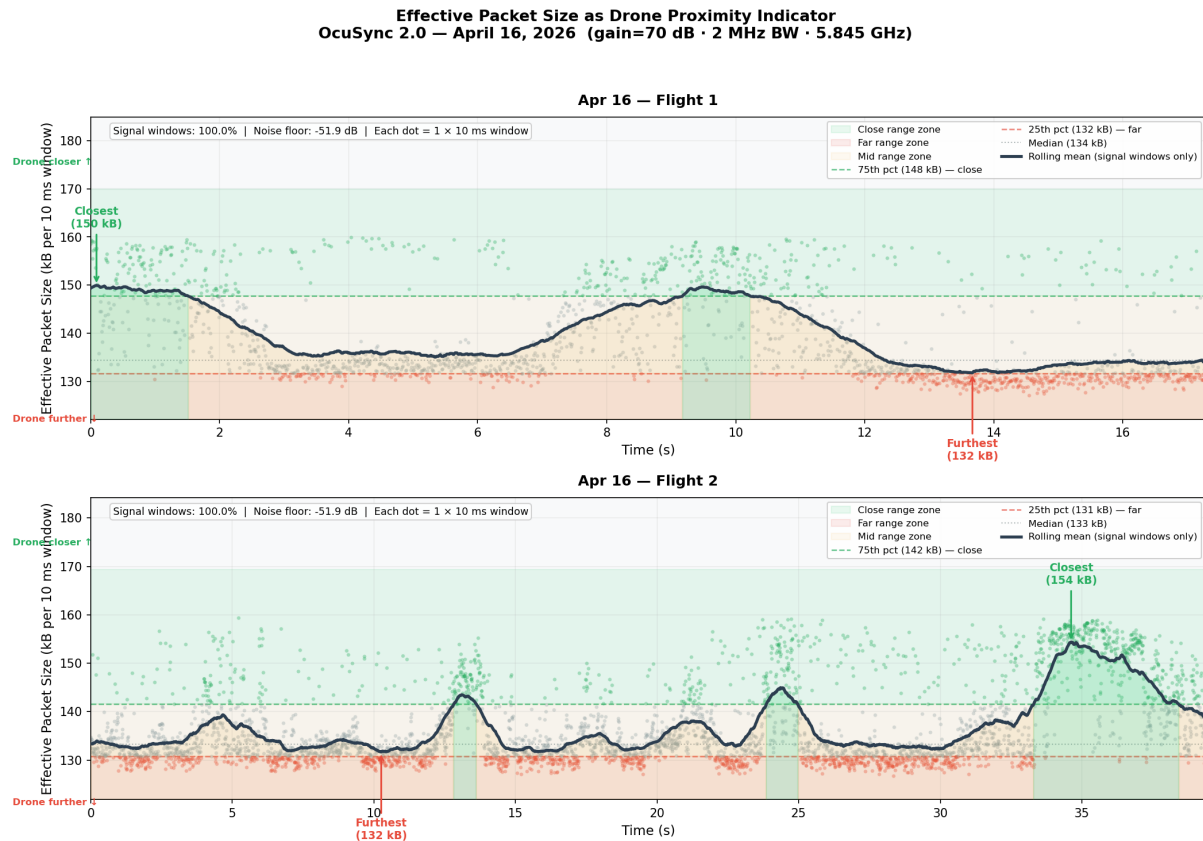


Figure 4.38: Packet Size Distance Proxy

Figure 4.38: Effective Packet Size as Drone Proximity Indicator — April 16, 2026.

Each dot is one 10 ms capture window where signal was detected above the noise floor (−51.9 dB). Its Y-position is the effective packet size in kB for that window. Colour encodes proximity zone: green dots are above the 75th percentile (close range), red dots are below the 25th percentile (far range), grey dots are in the middle band. The black line is a 1.5-second rolling mean computed only over signal-detected windows. The "Closest" annotation marks the rolling-mean peak; "Furthest" marks the trough.

Flight 1 shows the hypothesis confirmed clearly: the rolling mean starts near 150 kB (the drone is already close at recording start), remains elevated while the drone approaches to its closest point, then decreases progressively as it departs. The single clean approach-then-departure arc produces a smooth inverted-U shape in the smoothed curve.

Flight 2 (40 s, multi-cycle) shows the same correlation over multiple passes: the rolling mean oscillates with each approach and departure cycle, peaking at $t \approx 35$ s (closest pass, 154 kB) and dipping at $t \approx 10$ –11 s and $t \approx 21$ s (maximum range, 131–132 kB). The multi-cycle nature produces a noisier visual, but the underlying correlation between size and proximity is preserved

in each cycle.

The absolute variation is modest — roughly 130–165 kB across the full range of the experiment. This is expected: the custom OcuSync band is only 10 MHz wide and OcuSync transmits at a near-constant rate regardless of receiver conditions. The fraction of the transmitted burst that lands above the noise threshold at the receiver is what varies, and that fraction changes by only 15–20% between closest and furthest positions at this experiment’s distance range (40 m).

Metric 2: Active Energy

‘effective_size’ counts how many samples exceeded the threshold but weights every above-threshold sample equally — a barely-above-threshold sample from a far-away drone counts the same as a strongly-received sample from a close drone. This can create misleading readings when OcuSync hops partially overlap the capture window: a partial hop at close range may produce few but very strong samples, while a full hop at medium range may produce many weak samples, giving ‘effective_size’ of the far drone the same value despite the drone being further away.

‘active_energy’ addresses this by summing the power of each above-threshold sample rather than just counting them:

$$\text{active_energy} = \sum_{n : 10 \log_{10} |s_n|^2 > \text{NF}} |s_n|^2$$

where $\text{NF} = -51.9$ dB is the measured noise floor. A sample that is 10 dB above the floor contributes ten times more to this sum than one that barely clears it. ‘active_energy’ therefore simultaneously rewards:

- Duration: a hop that occupies more of the 10 ms window contributes more samples
- Strength: a drone that is closer contributes higher-power samples

It is the product of the two dimensions — equivalent to ‘effective_size × average_burst_power’ — and is expected to be a more robust proximity indicator than either dimension alone.

Effective Packet Size vs. Active Energy as Drone Proximity Indicators
OcuSync 2.0 — April 16, 2026 (gain=70 dB · 2 MHz BW · 5.845 GHz)

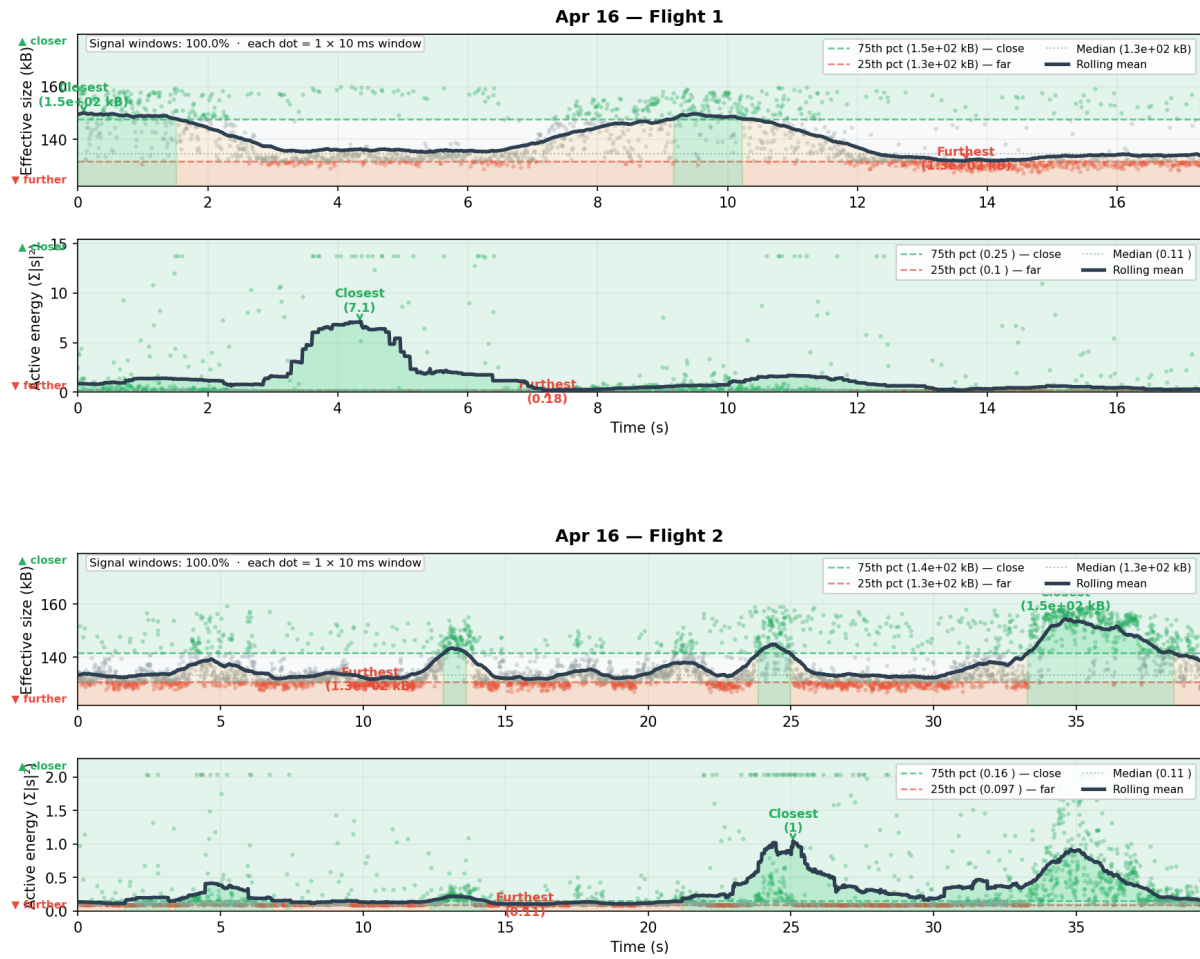


Figure 4.39: Active Energy Distance Proxy

Figure 4.39: Effective Packet Size vs. Active Energy as Proximity Indicators — April 16, 2026.

Each flight is shown as two vertically stacked panels sharing a time axis. The top panel shows ‘effective_size’ (kB); the bottom panel shows ‘active_energy’ ($\Sigma|s|^2$, dimensionless). The layout enables direct comparison of whether the two metrics agree or conflict in their proximity inference at each moment. Both panels use the same colouring convention: green scatter/fill = above 75th percentile (close zone), red = below 25th percentile (far zone), orange = mid-range. The black line is the 1.5-second rolling mean. The active energy y-axis is clipped at the 98th percentile to suppress isolated extreme outlier bursts.

Agreement and Conflict Between the Two Metrics

For the majority of windows the two metrics agree: when the drone is close, both ‘effective_size’ and ‘active_energy’ are elevated; when the drone is far, both are depressed. This agreement is

clearest in Flight 2, where both panels show the same multi-cycle oscillation with peaks at the same time positions.

Flight 1 reveals a meaningful conflict. The ‘effective_size’ panel shows the drone as close throughout the recording — the rolling mean stays elevated from $t = 0$ to $t \approx 14$ s. The ‘active_energy’ panel tells a different story: energy is near zero from $t = 0$ to $t \approx 3$ s, peaks sharply at $t \approx 4\text{--}5$ s, then collapses back to near zero after $t \approx 7$ s and stays flat for the remainder of the recording.

This conflict arises from OcuSync’s frequency hopping and the partial-hop problem:

Scenario	max_power	effective_size	active_energy	Ground truth
Close drone, hop covers full 10 ms	High	High	High	Close ✓
Close drone, hop starts mid-window (5 ms)	High	Medium	Medium	Close — size underestimates
Far drone, hop covers full 10 ms	Low	High	Low	Far — size overestimates
Far drone, hop covers 2 ms only	Low	Low	Low	Far ✓

Table 4.15: Effect of hop duration and signal strength on feature interpretation

In Flight 1, from $t = 0$ to $t \approx 3$ s, OcuSync hops were landing within the capture window long enough to push many samples above the noise floor (driving ‘effective_size’ high), but each individual sample was weak because the drone was still at medium range or the hop was edge-clipped. ‘active_energy’ penalises these weak-but-numerous windows correctly and reports near-zero, revealing the drone was not yet at closest approach. The true closest approach — when bursts were both strong and sustained — is accurately identified by ‘active_energy’ at $t \approx 4\text{--}5$ s.

Conclusion: ‘active_energy’ is the more reliable proximity indicator of the two, because it requires the burst to be simultaneously strong (drone close) and long-lasting (full hop captured) to register as a close-range event. ‘effective_size’ is easier to interpret physically (kB is a familiar unit) but is susceptible to partial-hop inflation when the drone is at medium range. For applications that require passive proximity estimation from OcuSync RF without any GPS or calibration, ‘active_energy’ should be used as the primary metric.

4.10 Wideband OcuSync FHSS Spectrogram

4.10.1 Motivation

All previous experiments in this chapter operated in narrowband mode: a single 10 MHz OcuSync sub-channel at 10 MS/s, sufficient to capture the TDD burst structure and amplitude fingerprint within one hop. However, this approach cannot visualise the drone’s frequency-domain behaviour — the fact that OcuSync is a Frequency-Hopping Spread Spectrum (FHSS) protocol that continuously relocates across approximately 10 sub-channels in the 5.725–5.825 GHz band. From the narrowband fingerprint alone, FHSS manifests only as missing windows (frames where the drone hopped away from the monitored frequency), which is an absence of evidence rather than a positive observation of the hopping pattern.

Shi et al. [17] demonstrated that applying a Short-Time Fourier Transform (STFT) to a wideband IQ recording of the whole OcuSync band produces striking rectangular box patterns: bright time-frequency tiles where the drone’s 10 MHz OFDM burst lands on a specific sub-channel for one hop period, surrounded by a dark noise floor where no signal is present. Six DJI models were distinguished by the visual texture of these boxes (hop rate, per-channel dwell time, spectral width).

This section describes our attempt to reproduce and extend this wideband FHSS spectrogram analysis on the DJI Matrice 300 RTK using the USRP B210, including the iterative process of overcoming hardware and software obstacles until a clean result matching the reference paper was obtained.

4.10.2 Capture Configuration

The wideband spectrogram experiments used fundamentally different capture parameters from all earlier experiments in this thesis:

4.10.3 First Attempts: 30 dB Gain (Failed)

Initial recordings on April 24, 2026, used 30 dB RX gain — the same setting employed throughout earlier narrowband experiments. Two recordings were taken:

- ‘drone_present_20260424_135725.iq’
- ‘drone_present_20260424_140439.iq’

The resulting spectrograms showed almost exclusively noise. The FHSS box patterns that are clearly visible in Shi et al. [17] Fig. 3 were absent; the plot was a nearly uniform dark field with no discernible frequency-domain structure.

Parameter	Narrowband (Chapters 4.1–4.9)	Wideband (this section)
Centre frequency	5.845 GHz (single OcuSync channel)	5.7875 GHz (band midpoint)
Sample rate	10 MS/s	40 MS/s
RF bandwidth	10 MHz (one OcuSync sub-channel)	40 MHz (5.7675–5.8075 GHz, spans 4 sub-channels)
Capture tool	Python iq_size_capture_v3.py	C++ rx_samples_to_file (UHD binary)
Wire format	sc16 (16-bit I/Q)	sc8 (8-bit I/Q) at >20 MS/s
Output format	CSV statistics	Raw complex64 IQ binary
What is visible	TDD burst pattern within one channel	Full FHSS hop trajectory across channels

Table 4.16: Narrowband vs. Wideband Capture Comparison

OcuSync RF Spectrogram — DJI Matrice 300 RTK
centre 5.7875 GHz · 40 MHz BW · STFT L=1024 hop=512 · amplitude, noise floor + 1.5 σ

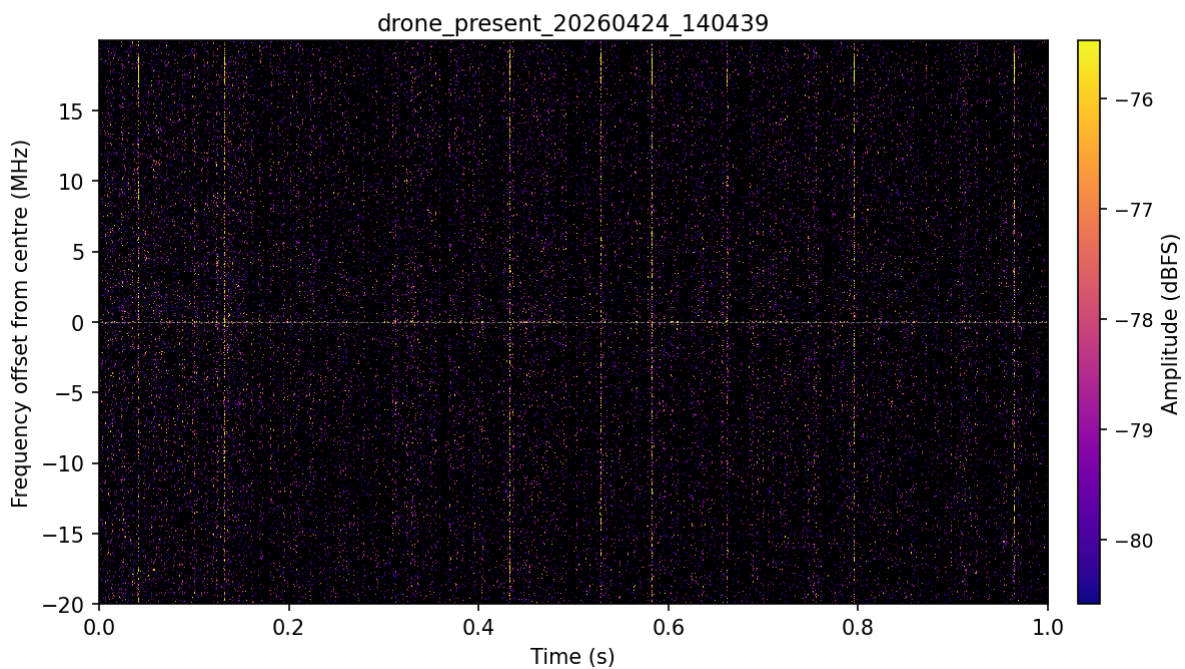


Figure 4.40: Spectrogram Amplitude

Figure 4.40: Wideband Spectrogram — 30 dB Gain (failed). The spectrogram is uniformly dark, showing only the thermal noise floor across the full 40 MHz bandwidth. No FHSS boxes or any structured signal is visible. The drone (M300 RTK, idle mode, motors off) was powered on and connected to its controller throughout the recording.

This outcome was not immediately obvious. The 30 dB setting had produced usable signals in all narrowband experiments, but the wideband configuration introduced two new noise penalties simultaneously:

1. Thermal noise scales with bandwidth. Moving from 10 MHz to 40 MHz increased the noise power floor by $10 \cdot \log_{10}(40/10) = +6$ dB, raising the noise floor by 6 dB even before any other change.

2. Each FHSS hop occupies only a fraction of the capture bandwidth. The OcuSync sub-channels occupy approximately 10–18 MHz each. In the 40 MHz capture window, a single hop covers at most 45% of the frequency bins; the remaining 55% remain at noise floor regardless of the drone’s transmit power. The STFT spreads the drone’s signal energy across only those bins where it actually transmits, so the per-bin average power is lower than in the narrowband case where the drone fills the entire bandwidth.

The combined effect — wider noise aperture plus lower per-bin signal density — rendered the idle M300 RTK’s OcuSync transmissions undetectable at 50 dB gain.

4.10.4 Successful Capture: 60 dB Gain

Increasing RX gain to 60 dB (+10 dB) on the third recording attempt immediately produced visible FHSS structure. Two 15-second recordings were taken at 60 dB:

- ‘drone_present_20260424_141027.iq’ — 1,487 MB, 185.8 M samples
- ‘drone_present_20260424_141455.iq’ — 1,227 MB, 153.3 M samples

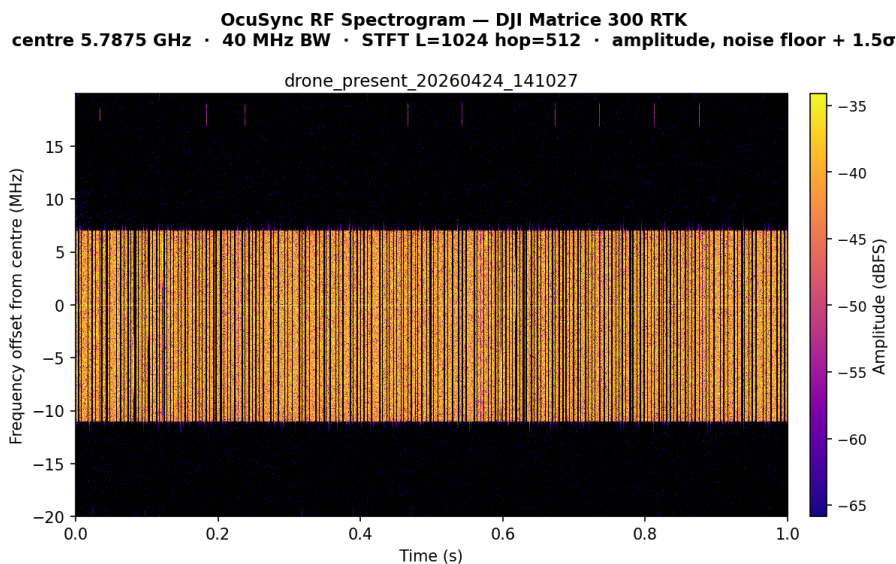


Figure 4.41: Spectrogram Amplitude

Figure 4.41: Wideband Spectrogram — 60 dB Gain, Full 1-Second View. The first second of the ‘141027’ recording. Bright horizontal bands are now visible against the dark noise back-

ground. Each band represents an OcuSync FHSS hop: the drone transmits a broadband OFDM burst on one sub-channel for approximately 8–10 ms, then hops to another sub-channel. The Y-axis shows frequency offset from the 5.7875 GHz centre frequency; the X-axis shows elapsed recording time in seconds.

4.10.5 Software Pipeline: ‘wideband_spectrogram.py’

The ‘wideband_spectrogram.py’ script implements a two-stage pipeline: capture and plot, driven by two CLI subcommands (‘record’ and ‘plot’). The code went through several iterations before producing a correct and memory-efficient result.

Stage 1: Capture (‘do_capture’)

The capture stage calls ‘rx_samples_to_file’ via ‘subprocess.run()’, staging the sc16 intermediate file in ‘/dev/shm’ (a RAM-backed filesystem) to avoid contention between the USB transfer and a spinning disk write — at 40 MS/s the sc16 file is written at 160 MB/s and NVME access latency must not stall the USRP ring buffer. After the C++ binary exits, the sc16 binary is converted to complex64 using a chunked converter:

Listing 4.1: SC16 to Complex64 IQ Conversion

```
CONV_CHUNK = 50_000_000 # complex samples per chunk (= 200 MB int16 pairs)
```

```
with open(sc16_path, 'rb') as f_in, open(iq_path, 'wb') as f_out:
```

```
    while True:
```

```
        raw_bytes = f_in.read(CONV_CHUNK * 4) # 4 bytes per sc16 sample
```

```
        if not raw_bytes:
```

```
            break
```

```
        raw_i16 = np.frombuffer(raw_bytes, dtype=np.int16)
```

```
        raw_f32 = raw_i16.astype(np.float32)
```

```
        del raw_i16
```

```
        raw_f32 /= 32768.0
```

```
        iq_chunk = raw_f32.view(np.complex64)
```

```
        iq_chunk.tofile(f_out)
```

```
        actual_samples += len(iq_chunk)
```

```
        del raw_f32
```

The chunked approach was necessary because naive full-array conversion (loading all 185.8 M complex samples at once) would require $185.8 \text{ M} \times 4 \text{ bytes (int16)} + 185.8 \text{ M} \times 8 \text{ bytes (float32)} \approx 2.2 \text{ GB}$ peak RAM for the 15-second recording; this caused an out-of-memory (OOM) error on the capture machine. Processing in 50 M-sample chunks limits peak RAM to approximately 600 MB regardless of recording length, at the cost of sequential passes over the file.

Stage 2: STFT (‘_compute_stft_batched’)

The plotting stage computes an STFT of the saved complex64 IQ file. The initial implementation used ‘scipy.signal.stft’ directly:

Listing 4.2: STFT Computation Using SciPy

```
freqs_hz, times_s, Zxx = scipy_stft(samples, fs=rate, window='hann',
    nperseg=1024, noverlap=512,...)
```

This failed with an OOM error whenever more than approximately 2 seconds of a 40 MS/s recording was processed. The root cause is that ‘scipy.stft’ allocates the full output matrix as complex128 (16 bytes per element) before conversion to magnitude:

$$\text{Memory} = N_f \times N_t \times 16 \text{ B} = 1024 \times \frac{t \cdot 40 \text{ MS/s}}{512} \times 16 \text{ B} \approx 1.3 \text{ GB per second}$$

A 15-second recording would require approximately 19 GB — far beyond available RAM.

The replacement implementation, ‘_compute_stft_batched’, avoids this via three mechanisms:

1. Memory-mapped file access. The IQ file is opened with ‘np.memmap’, which maps the file into virtual address space without loading it into RAM. Only the pages actually accessed (current batch) are paged in by the OS.

2. Auto-stride temporal downsampling. The function computes how many STFT frames the full recording would produce, then automatically selects a stride (every k-th frame) so the float32 output matrix stays within a configurable target (default 350 MB):

Listing 4.3: Output Byte Size and Stride Calculation

```
out_bytes_full = n_frames_full * nperseg * 4 # float32
stride = max(1, int(np.ceil(out_bytes_full / (target_mb * 1e6))))
```

For the 15-second 40 MS/s recording: ‘n_frames_full ≈ 14,500’, giving ‘out_bytes_full ≈ 57 MB’ — well under 350 MB, so stride = 1 (no downsampling). For longer recordings, stride increases automatically.

3. Batch FFT in 2,048-frame chunks. Frames are materialised from the memmap in batches of 2,048, keeping peak RAM per batch below approximately 150 MB.

The Window Normalisation Bug

During testing, the amplitude scale was found to be approximately +54 dB higher than expected — the noise floor showed at +20 dBFS rather than the expected ≈ −35 dBFS. The cause was a missing Hann window normalisation step. ‘scipy.stft’ divides the FFT output by ‘window.sum()’

(the sum of all window coefficients = $0.5 \times \text{NPERSEG}$ for a Hann window) as part of its COLA (Constant Overlap-Add) scaling. The manual FFT in the batched implementation had omitted this:

Listing 4.4: FFT Magnitude Normalisation

```
# Without normalisation:
mag = np.abs(fft_out[:, shift_idx]).astype(np.float32) # values ~1800× too large

# After fix:
mag /= float(window.sum()) # window.sum() 512 for Hann, nperseg=1024
```

4.10.6 Visualising the FHSS Boxes

With the corrected amplitude scale and 60 dB gain, the full-time-range spectrogram (Figure 4.28) shows clear horizontal banding. However, at the default time axis scale the individual FHSS boxes appear as narrow vertical stripes because each hop lasts only 8–10 ms. Compressing the time axis to a 150 ms window (`‘-xlim 0 0.15’`) and restricting the frequency axis to ± 30 MHz (`‘-ylim -30 30’`) reveals the box structure clearly:

Listing 4.5: Generating a Wideband Spectrogram

```
python3 wideband_spectrogram.py plot
--iq wideband_iq/drone_present_20260424_141027.iq
--amplitude --plot-seconds 1.0
--xlim 0 0.15 --ylim -30 30
-o wideband_iq/spectrogram_boxes_20260424_141027.png
```

Figure 4.42: Wideband FHSS Spectrogram — OcuSync Hop Boxes Visible (150 ms window). DJI Matrice 300 RTK, April 24, 2026. Recording ‘drone_present_20260424_141027.iq’; first 150 ms shown. Centre frequency 5.7875 GHz, 40 MHz bandwidth, RX gain 60 dB (USRP B210). STFT: $L = 1024$ Hann window, $\text{hop} = 512 \rightarrow \Delta f = 39.1$ kHz, $\Delta t = 12.8$ μ s. Colour encodes amplitude in dBFS after denoising (noise floor + 1.5σ per frequency bin; sub-threshold bins rendered black). Each bright horizontal rectangle is one OcuSync FHSS hop: the drone’s OFDM waveform occupies approximately 5–15 MHz of bandwidth centered on the selected sub-channel for approximately 8–10 ms before hopping. Compare with Shi et al. [17] Fig. 3(b) (MATRICE 100 panel): the visual pattern — rectangular bright tiles on a dark background stacked across the hopping band — is consistent between the two experiments.

OcuSync RF Spectrogram — DJI Matrice 300 RTK
centre 5.7875 GHz · 40 MHz BW · STFT L=1024 hop=512 · amplitude, noise floor + 1.5 σ

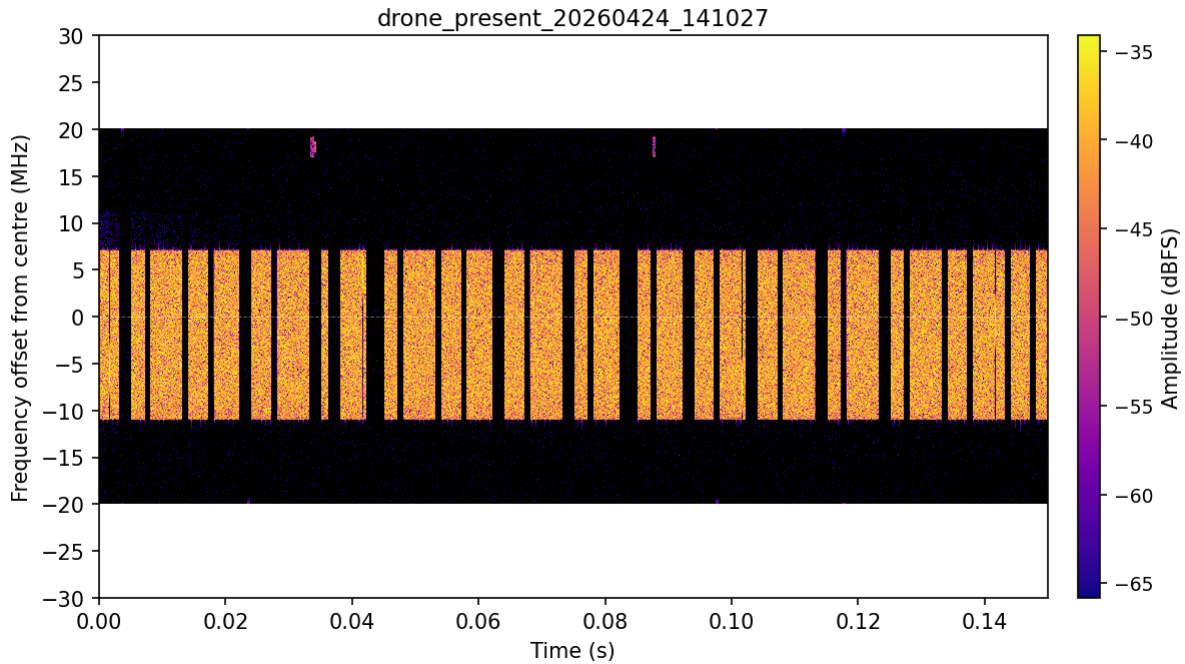


Figure 4.42: Spectrogram

Observed FHSS structure:

- **Hop period:** Approximately 8–10 ms per rectangular box visible along the time axis, consistent with the OcuSync specification of approximately 10 ms inter-hop dwell time.
- **Spectral width of each hop:** Each box spans approximately 10–15 MHz on the frequency axis, consistent with OcuSync’s 10 MHz nominal channel bandwidth.
- **Hop extent within capture window:** Hops are visible across approximately ± 20 MHz from the centre, spanning from approximately -11 MHz to $+7$ MHz — four of the OcuSync sub-channels fall within the 40 MHz capture bandwidth.
- **Vertical dark gaps within boxes:** Some boxes show thin horizontal dark lines near their centre, consistent with TDD uplink slots (the controller transmitting to the drone) that silence the downlink within a hop period.

4.10.7 Comparison to V2 Amplitude Fingerprint

This wideband STFT approach and the V2 amplitude fingerprint (Section 3.4.2, Section 4.2.4) both operate on the amplitude of IQ samples — neither ever decodes OcuSync packets, navigates any protocol layer, or requires any prior knowledge of the encryption keys. In that fundamental sense, they are two instances of the same “envelope weighting” concept described in Section

3.4.4. However, they analyse completely different axes of variation:

Aspect	V2 Amplitude Fingerprint (§4.2.4)	Wideband STFT Spectrogram (§4.11)
Centre frequency	5.845 GHz (fixed OcuSync channel)	5.7875 GHz (band midpoint)
Bandwidth	10 MHz (one sub-channel)	40 MHz (four sub-channels)
Sample rate	10 MS/s	40 MS/s
Capture tool	Python iq_size_capture_v3.py	C++ rx_samples_to_file
Output representation	128-bin time-domain amplitude vector	2D time–frequency matrix (STFT)
X axis	Recording time (10 ms windows)	Recording time (continuous, 12.8 μ s STFT frames)
Y axis	Time <i>within</i> a 10 ms window (78 μ s/bin)	RF frequency offset from centre (MHz)
Z axis (colour)	Average amplitude per 78 μ s bin (dBFS)	Amplitude per STFT time–frequency tile (dBFS)
What is revealed	OcuSync TDD burst pattern, slot allocation, carrier phase	OcuSync FHSS hop trajectory across sub-channels
Signal processing	Mean amplitude of raw complex samples in each bin	Hann-windowed FFT per 12.8 μ s slice (STFT)
Use case	ML feature input, state classification (96% accuracy)	Visual FHSS identification, comparison to prior art
Output size per second	~5 KB CSV (statistics only)	~150 MB float32 STFT matrix (raw)
Frequency resolution	None (spectrally flat bandwidth capture)	39.1 kHz per bin (STFT-limited)

Table 4.17: Narrowband Fingerprint vs. Wideband STFT Spectrogram

The key distinction is what the Y axis represents: In the V2 fingerprint spectrogram (Figure 4.4), the Y axis is time within a 10 ms capture window — the 128 bins map the temporal structure of OcuSync’s TDD schedule. When the drone transmits on the monitored 5.845 GHz channel, the entire 10 MHz bandwidth illuminates simultaneously (OFDM is by design spectrally flat), so sub-channel frequency variation is invisible. In the wideband STFT spectrogram (Figure 4.29), the Y axis is RF frequency offset in MHz — the 1,024 FFT bins map the spectral footprint of each OcuSync transmission. When the drone hops to a different sub-channel, that hop appears as a bright horizontal strip at a different Y position.

A useful analogy: the V2 fingerprint is like listening to a single radio station and noting the rhythm of the presenter’s speech (when they speak vs. pause); the wideband STFT is like scan-

ning the entire FM band and seeing which station is transmitting on which frequency at each moment. Both observations are derived entirely from the physical signal envelope — no content is decoded — but they illuminate orthogonal aspects of the drone’s communication behaviour.

The V2 fingerprint approach is better suited for machine learning classification because it produces a compact, fixed-size vector ($128 \text{ bins} \times 10 \text{ ms window}$) that feeds directly into a Random Forest. The wideband STFT produces a large 2D matrix per second that is difficult to reduce to fixed-size features without losing the spatial structure that carries the discriminative information; it is primarily useful for visual analysis and qualitative validation of the FHSS protocol behaviour.

4.10.8 Summary of Iterative Process

The path from initial failed recordings to the clean FHSS box spectrogram of Figure 4.29 involved the following sequence of problems and fixes:

Problem	Root Cause	Solution
50 dB gain: spectrogram uniformly dark	6 dB wider noise floor (40 MHz vs 10 MHz) + idle drone transmits at reduced power	Increased RX gain to 60 dB
<code>scipy.stft</code> OOM on recordings > 2 s	Full complex128 output: $\sim 1.3 \text{ GB/s}$ at 40 MS/s	Replaced with <code>_compute_stft_batched</code> : <code>mmap</code> + <code>auto-stride</code> + <code>float32</code>
<code>sc16</code> → <code>complex64</code> conversion OOM (15 s file)	All-at-once loading: $12 \text{ B/sample} \times 185 \text{ M samples} \sim 2.2 \text{ GB peak}$	Chunked conversion : 50 M samples/chunk $\sim 600 \text{ MB peak}$
Amplitude scale +54 dB too high (+20 dBFS instead of -35 dBFS)	Missing Hann window normalisation <code>mag /= window.sum()</code> in manual FFT	Added <code>mag /= float(window.sum())</code> after magnitude computation
FHSS hops appear as narrow vertical lines, not “boxes”	Default time axis scale too wide (each hop $\sim 8 \text{ ms}$ in 1,000 ms view)	Applied <code>--xlim 0 0.15 --ylim -30 30</code> to display crop (150 ms $\times$ 60 MHz)

Table 4.18: Wideband Spectrogram Development — Problems and Solutions

The final configuration — 60 dB gain, `_compute_stft_batched`, correct window normalisation, `--xlim 0 0.15 --ylim -30 30` display crop — produces spectrograms consistent with the reference publication [17] and clearly demonstrates OcuSync’s FHSS behaviour for the DJI Matrice 300 RTK.

4.11 Summary of Experimental Results

Experiment	Key Result	Significance
Simulation baseline	33.3% accuracy (random)	Validates methodology (no simulation artifacts)
Lab stationary (3-class)	96% accuracy	Proves encrypted traffic classification works
Camera movement (binary)	95.2% accuracy	Validates “data volume” hypothesis
5.8 GHz jamming	Video killed, controls survive	Reveals OcuSync TDD asymmetry
2.4 GHz jamming	Same result on custom channel	Confirms control resilience is protocol-level
OcuSync channel switching	Two-layer freq. management	Layer 1: ms-scale hopping; Layer 2: ~1s adaptive switching
GPS spoofing	Successful on phone + DJI Mini	Demonstrates GPS L1 vulnerability
Replay attack	Failed	Confirms OcuSync cryptographic protection
Real flight (no camera)	Distance tracking to ~4.3 m	Identifies noise-floor sensing limit
Real flight (camera, Apr 6)	0.9–5.3% reliable, ranging failed	Confirms signal must be above noise floor for ranging
Real flight (improved, Apr 16)	42.9% windows above floor, max burst -0.8 dB	+37 dB gain; custom band 5839–5849 MHz; 2 MHz BW covers 20% of hops; fingerprint distinct from noise
Rolling-peak detection (Apr 16)	100% per-second detection at 40 m	1 s rolling max of peak power; coarse movement tracking without ML; confirms operational utility of passive RF monitoring
Wideband FHSS spectrogram (Apr 24)	FHSS box pattern matches Shi et al. [17] Fig. 3	40 MS/s, 40 MHz BW at 60 dB gain; C++ rx_samples_to_file; batched STFT with window normalisation fix; OcuSync hop period ~8–10 ms, 4 sub-channels visible

Table 4.19: Summary of All Experimental Results

5 Conclusion

5.1 Summary of Key Findings

This thesis has investigated the feasibility of classifying encrypted drone communication behavior through passive Radio Frequency physical layer analysis, and has evaluated the security posture of DJI drone systems against multiple attack vectors. The research has yielded several significant findings:

- 1. Encrypted traffic classification is achievable through physical layer analysis.** A Random Forest classifier trained on 46 features extracted from raw IQ samples achieved 96% accuracy in distinguishing three drone states (no drone, drone idle, drone with camera movement) and 95.2% accuracy in binary camera-movement detection. These results demonstrate that DJI’s AES-256 encryption, while protecting the content of communications, does not prevent behavioral inference from physical layer characteristics.
- 2. The volume of transmitted data encodes operational behavior.** The most important features for classification — active ratio, duty cycle, effective size, and phase drift — all measure aspects of how much and how the drone transmits, rather than what it transmits. Camera movement forces the video encoder to process changing frames, increasing the data rate from 18% to 51.6% channel occupancy. This validates the core hypothesis that operational state is encoded in traffic volume.
- 3. OcuSync’s TDD architecture exhibits asymmetric vulnerability.** Jamming experiments at both 5.8 GHz and 2.4 GHz revealed that video downlink is significantly more vulnerable to RF jamming than control uplink. This asymmetry stems from the fundamental design of Time-Division Duplexing: video consumes approximately 90% of airtime with continuous high-bandwidth data, while control uses only 5–10% of airtime with small, error-correction-protected packets that survive partial channel degradation.
- 4. GPS spoofing succeeds under specific, restrictive conditions.** Fabricated GPS L1 signals transmitted through the USRP B210 successfully manipulated the location reported by both an Android smartphone and a DJI Mini drone, but only when competing location sources (WiFi, cellular) were disabled and aggressive spoofing parameters were used. This demonstrates that while GPS L1 remains fundamentally vulnerable, modern multi-source location fusion provides substantial practical protection.
- 5. OcuSync is resilient to replay attacks.** The combination of per-session AES-256 encryption, sequence numbers, timestamps, and frequency hopping ensures that captured RF signals cannot be replayed to control a DJI drone. This confirms the effectiveness of DJI’s cryptographic design.

against this class of attack.

6. RF-based distance estimation has practical limits, but hardware reconfiguration unlocks signal. Real flight experiments demonstrated that RSSI-based distance tracking is effective only while the received signal remains significantly above the noise floor. In the no-camera flight, the system entered a "distance-blind region" at approximately 4.3 meters. The drone was configured with a custom OcuSync band of 5839–5849 MHz (10 MHz wide) for all real flights, with the USRP centred at 5.845 GHz. A critical retroactive finding is that the April 6 flights using 10 MHz bandwidth already covered 100% of the drone's hopping range — the failure was entirely due to insufficient gain (40 dB → noise floor at -70 dB). Increasing gain to 70 dB and narrowing bandwidth to 2 MHz delivered a +37 dB combined improvement, raising 42.9% of April 16 windows above the -51.9 dB noise floor. However, the 2 MHz capture now covers only 20% of the 10 MHz custom band. The optimal untested configuration — 10 MHz BW at gain=70 — would provide full hop coverage with an adequate noise floor of approximately -44.9 dB.

7. OcuSync employs two-layer frequency management. Adaptive jamming experiments revealed that OcuSync operates with fast intra-channel hopping at millisecond timescales (invisible to our spectrum scanner) and slow adaptive channel switching at 0.5–2 second timescales when sustained interference is detected. The 1 second switching delay represents an exploitable vulnerability for reactive jamming systems, as a jammer can detect and follow the channel change within this window.

8. Amplitude fingerprinting fails at noise-floor signal levels. Camera flight data showed 0.9999 cosine similarity between drone flights and the NoDrone (no drone present) baseline, confirming that the amplitude fingerprint measures the shape of the noise distribution rather than the drone's RF identity when the signal-to-noise ratio is insufficient. Fingerprint-based identification requires the signal to be well above the noise floor.

9. Wideband STFT spectrograms directly visualise OcuSync FHSS behaviour. Extending the capture to 40 MS/s (40 MHz bandwidth) centred on the full 5.8 GHz OcuSync band, and processing the raw IQ with a memory-efficient batched STFT pipeline, produced time–frequency spectrograms showing OcuSync's Frequency-Hopping Spread Spectrum behaviour as rectangular "box" patterns — consistent with Shi et al. [23] Fig. 3 for the DJI Matrice 100. The wideband approach complements the narrowband V2 amplitude fingerprint (which reveals TDD burst timing) by exposing the orthogonal frequency-domain hop trajectory. Both methods derive entirely from the physical signal amplitude envelope without decoding any encrypted content.

10. Rolling-peak power analysis enables reliable drone detection and coarse movement tracking without machine learning. By replacing mean power with the per-window peak

and applying a 1-second rolling maximum, both April 16 flights at approximately 40 m range achieved a 100% per-second detection rate — every second of the outdoor flight contained at least one window with signal clearly above the noise floor. The smoothed peak-power curve further revealed approach/departure cycles consistent with the known flight trajectories, demonstrating that passive RF monitoring of encrypted OcuSync traffic can provide operationally useful presence detection and coarse trajectory information using only basic signal processing on the raw power envelope.

5.2 Research Contributions

This thesis makes the following contributions to the fields of drone security, RF analysis, and machine learning:

- 1. A comprehensive 46-feature RF characterization framework** organized into three progressive layers (V1: energy, V2: amplitude/phase/fingerprint, V3: inter-window temporal), providing a reusable foundation for drone RF traffic analysis research.
- 2. Empirical evidence** that encrypted drone communications leak behavioral information at the physical layer, with classification accuracies of 95–96% — sufficient for practical applications including surveillance detection, airspace monitoring, and forensic analysis.
- 3. The 128-bin temporal fingerprint**, which captures the OcuSync TDD schedule within each 10 ms window and reveals dynamic slot allocation, protocol timing signatures, and carrier phase behavior that are invisible to conventional power-based analysis.
- 4. Systematic characterization of OcuSync’s security posture** across three attack categories (jamming, spoofing, replay), providing a multi-dimensional assessment of the protocol’s strengths and vulnerabilities.
- 5. Honest assessment of real-world limitations**, including the noise-floor sensing boundary, the dependence on video activity for classification quality, the impossibility of absolute distance estimation without calibration, and the failure of amplitude fingerprinting at noise-floor signal levels.
- 6. Characterization of OcuSync’s two-layer frequency management**, distinguishing fast intra-channel hopping (millisecond-scale) from slow adaptive channel switching (0.5–2 second scale), and identifying the switching delay as an exploitable vulnerability for reactive jamming.

5.3 Limitations

- 1. Limited drone models:** The primary experiments used only DJI drones (Mavic 2, M300 RTK, Mini) with OcuSync protocols. Generalization to other manufacturers' drones (Skydio, Autel, Parrot) and protocols remains to be validated.
- 2. Laboratory constraints:** Most classification experiments were conducted at short range (1 m) in an indoor environment. Performance in a lot of outdoor, long-range, or multi-drone scenarios has not been evaluated.
- 3. Static scenarios:** The stationary experiments captured only static behavioral states (idle vs. camera moving). Classification of dynamic, continuous, mixed-state scenarios (e.g., flying while intermittently moving the camera) was not tested.
- 4. Real flight limitations:** Three sets of real flights were conducted — one without camera (March 26), four with camera at gain=40 (April 6, signal at noise floor), and two with improved configuration at gain=70 (April 16, +37 dB improvement). While the April 16 configuration achieved 100% per-second drone detection at 40 m and coarse movement tracking, absolute distance ranging and drone state classification remained non-viable at this range with the omnidirectional antenna. The optimal untested configuration — 10 MHz BW at gain=70 — would provide full hop coverage with the +37 dB gain advantage.
- 5. No simultaneous multi-drone scenarios:** All experiments involved a single drone. The behavior of the classification system in the presence of multiple simultaneous drone signals on overlapping channels has not been studied.
- 6. GPS spoofing indoor-only:** The GPS spoofing attack on the DJI Mini was conducted exclusively indoors. Outdoor validation was not performed for safety reasons.
- 7. Single USRP platform:** All experiments — RF capture, jamming, GPS spoofing, and replay attacks — were conducted using the same USRP B210 SDR. While this ensures consistency across experiments, the results are tied to this specific hardware platform's sensitivity, dynamic range, and oscillator characteristics. Results may differ on other SDR platforms.

5.4 Future Work

Several promising directions for future research emerge from this thesis:

- 1. Dual-band simultaneous capture:** Using the USRP B210's two RX channels to capture both 2.4 GHz and 5.8 GHz simultaneously would enable complete characterization of OcuSync's dual-band behavior, including video/control band allocation and failover mechanisms.
- 2. Dual-band jamming for complete C2 denial:** Developing a dual-band jammer that simul-

taneously attacks both 2.4 GHz and 5.8 GHz could trigger the M300's RTH (Return-to-Home) or auto-land failsafe, providing a complete denial-of-service capability for authorized counter-drone operations.

3. CNN-based classification: Feeding the 128-bin fingerprint vectors directly into a Convolutional Neural Network could discover temporal patterns that summary statistics (mean, std) miss, potentially improving classification accuracy for challenging states.

4. Multi-drone fingerprinting: Each drone's crystal oscillator produces a slightly different carrier frequency offset (visible in the phase fingerprint slope). This hardware-specific signature could enable identification of individual drones, even of the same model.

5. Extended flight scenarios: Additional real flights with camera active, controlled distances, and various flight patterns (approaching, departing, circling) are needed to build a comprehensive real-world classification model.

6. Counter-counter-drone defenses: Investigating whether drones can detect and evade RF-based behavioral classification — analogous to adversarial machine learning in the RF domain — would contribute to the understanding of the arms race between drone operators and monitoring systems.

7. Integration with other sensor modalities: Combining RF-based classification with acoustic (propeller sound), visual (camera), and radar detection could create a multi-modal drone monitoring system with improved accuracy and coverage.

BIBLIOGRAPHY

- [1] DJI, “Ocusync 2.0 transmission system,” 2019, available: <https://www.dji.com/>.
- [2] ———, “Dji security response center — secure communication white paper,” 2020, available: <https://security.dji.com/>.
- [3] proto17, “Dji droneid rf analysis,” 2022, gitHub Repository. Available: https://github.com/proto17/dji_droneid.
- [4] D. Bender, “Ocusync protocol analysis,” 2022, security Research Publication.
- [5] Ettus Research, “Usrc b200/b210 specifications,” 2023, available: <https://www.ettus.com/all-products/ub210-kit/>.
- [6] GNU Radio Project, “Gnu radio — the free & open source radio ecosystem,” 2023, available: <https://www.gnuradio.org/>.
- [7] A. C. Polak, S. Dolatshahi, and D. L. Goeckel, “Identifying wireless users via transmitter turn-on transients,” *IEEE Signal Processing Letters*, vol. 18, no. 8, pp. 468–471, 2011.
- [8] T. D. Vo-Huu and G. Noubir, “Fingerprinting wi-fi devices using software defined radios,” in *Proc. ACM WiSec*, 2016.
- [9] M. A. Richards, J. A. Scheer, and W. A. Holm, *Principles of Modern Radar: Basic Principles*. SciTech Publishing, 2010.
- [10] A. P. et al., “Website fingerprinting at internet scale,” in *Proc. NDSS*, 2016.
- [11] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [12] T. J. O’Shea, J. Corgan, and T. C. Clancy, “Convolutional radio modulation recognition networks,” in *Proc. International Conference on Engineering Applications of Neural Networks*, 2016.
- [13] K. S. et al., “Oracle: Optimized radio classification through convolutional neural networks,” in *Proc. IEEE INFOCOM*, 2019.
- [14] P. N. et al., “Investigating cost-effective rf-based detection of drones,” in *Proc. ACM DroNet Workshop*, 2016.
- [15] M. F. Al-Sa’d, A. Al-Ali, A. Mohamed, T. Khattab, and A. Erbad, “Rf-based drone detection and identification using deep learning approaches,” in *Proc. IEEE PIMRC*, 2019.
- [16] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, and I. Guvenc, “Detection and classification of uavs using rf fingerprints in the presence of wi-fi and bluetooth interference,” *IEEE Open Journal of the Communications Society*, vol. 1, pp. 60–76, 2020.

- [17] Y. Shi, T. Shan, and H. Dong, “Integrated detection of weak drone signals: From parameter estimation to sensitive classification,” *Digital Signal Processing*, vol. 168, p. 105524, 2026.
- [18] D. Adamy, *EW 101: A First Course in Electronic Warfare*. Artech House, 2001.
- [19] T. E. Humphreys, “Statement on the vulnerability of civil unmanned aerial vehicles and other systems to civil gps spoofing,” 2012, university of Texas at Austin.
- [20] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O’Hanlon, and P. M. Kintner, “Assessing the spoofing threat: Development of a portable gps civilian spoofer,” in *Proc. ION GNSS*, 2008.
- [21] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, “Unmanned aircraft capture and control via gps spoofing,” *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014.
- [22] T. Ebinuma, “Gps-sdr-sim: Gps signal simulator,” gitHub Repository. Available: <https://github.com/osqzss/gps-sdr-sim>.
- [23] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd ed. John Wiley & Sons, 1996.