2021-09

5G INSPIRED METHOD FOR RANGING OF UAVS IN SWARMING COMPOSITION

de Andrade Martins, Madjer

Monterey, CA; Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

5G INSPIRED METHOD FOR RANGING OF UAVS IN SWARMING COMPOSITION

by

Madjer de Andrade Martins

September 2021

Thesis Advisor: David A. Garren
Co-Advisor: Clark Robertson

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2021	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE 5G INSPIRED METHOD FOR RANGING OF UAVS IN SWARMING COMPOSITION	**5. FUNDING NUMBERS**
6. AUTHOR(S) Madjer de Andrade Martins	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

The employment of unmanned aerial vehicles is currently a fact in modern warfare. The benefit of using drones as a swarm, working together to accomplish a task, will help save lives; however, the communication among drones within a swarm is a challenge with the available technology due mainly to the power requirements to operate in a small device. Inspired by the massive machine-type communication 5th generation mobile networks, this work offers a novel method of identification and ranging for drones in swarm. The 5G communication channel's preamble with a Zadoff-Chu (ZC) sequence is expected to provide low power and less interference between devices and yet yield good mean-square error results when a matched filter is applied. Simulations considering different numbers of drones within a swarm embedded in noisy and Doppler-affected environments demonstrate promising results even in poor scenarios with small signal-to-noise ratio and high Doppler frequency shift, especially when the batch of ZC sequences' root indexes are selected into a special group.

14. SUBJECT TERMS 5G network, swarm drones, swarming drones, drones, drones swarm, UAV, unmanned aerial vehicle, Zadoff-Chu Sequence, ZC sequence, matched filter, orthogonal frequency-division multiplexing, OFDM, frequency offset, Doppler effect, Doppler, ranging, distance, mean square error, MSE, bit error rate, signal-to-noise ratio	**15. NUMBER OF PAGES** 87
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

5G INSPIRED METHOD FOR RANGING OF UAVS IN SWARMING COMPOSITION

Madjer de Andrade Martins
Lieutenant, Brazilian Navy
BS, Universidade Federal do Rio de Janeiro (UFRJ), 2008

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2021

Approved by: David A. Garren
Advisor

Clark Robertson
Co-Advisor

Douglas J. Fouts
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The employment of unmanned aerial vehicles is currently a fact in modern warfare. The benefit of using drones as a swarm, working together to accomplish a task, will help save lives; however, the communication among drones within a swarm is a challenge with the available technology due mainly to the power requirements to operate in a small device. Inspired by the massive machine-type communication 5th generation mobile networks, this work offers a novel method of identification and ranging for drones in swarm. The 5G communication channel's preamble with a Zadoff-Chu (ZC) sequence is expected to provide low power and less interference between devices and yet yield good mean-square error results when a matched filter is applied. Simulations considering different numbers of drones within a swarm embedded in noisy and Doppler-affected environments demonstrate promising results even in poor scenarios with small signal-to-noise ratio and high Doppler frequency shift, especially when the batch of ZC sequences' root indexes are selected into a special group.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

5G	5th Generation Mobile Network
CA	Constant-Amplitude
CAZAC	Constant-Amplitude, Zero Autocorrelation
DFT	Discrete Fourier Transform
DOD	Department of Defense
GCD	Greatest Common Divisor
ksps	KSymbols per Second
lag	the distance from zero to a series of numbers
MANET	Mobile Ad Hoc Network
MF	Matched Filter
mMTC	Massive Machine-Type Communication
MSE	Mean-Square Error
NPS	Naval Postgraduate School
OFDM	Orthogonal, Frequency-Division Multiplexing
PRACH	Physical Random-Access Channel
R	Root Index of a ZC Sequence
SNR	Signal-to-Noise Ratio
SRS	Sounding Reference Signals
UAV	Unmanned Aerial Vehicle

USN U.S. Navy

VANET Vehicular Ad Hoc Network

ZAC Zero Autocorrelation

ZC Zadoff-Chu

Acknowledgments

I would like to acknowledge God, who listens to my complaints—without complaining back—guides me in tough decisions, and makes me strong enough to survive for 8 quarters, 109 laboratory reports and projects, and 86 exams.

I would like to thank my advisor Dr. John D. Roth, who supported all of my research with patience, wisdom, and knowledge and weekly meetings to answer my countless doubts. Unfortunately, Professor Roth has taken a leave of absence from NPS, but I must make it clear that without his help, this work would not have been completed.

I would like to thank my advisor Dr. David A. Garren, who was fundamental in the completion of my thesis by evaluating my work and providing support as I worked to bring my thesis to completion.

I would like to thank my co-advisor Dr. R. Clark Robertson, who reviewed my thesis, making important suggestions and advising about significant theoretical elements.

I would like to thank my education technician, Julie Samples, my academic associate, Preetha Thulasiraman, my program officers, CDR Clay Herring and LCDR Brannon Chapman, and the chair of the Department of Electrical and Computer Engineering, Professor Douglas Fouts, for guiding me in every decision I made during my MSEE Program at NPS.

I would like to thank my writing coaches and professors, Michael Thomas, Michele D'Ambrosio, Meg Beresik and Martha Brumback, for all their support and hours dedicated to reviewing my thesis text, providing corrections of my non-perfect English.

I wish to show my gratitude to the Washington Brazilian Naval Commission and Brazilian Navy Directorate of Communication, who provided support during my stay in the United States, as well as acknowledge the Brazilian Navy for the opportunity to live a new experience in a new country, studying beside people from all over the world, which made it possible to increase my knowledge about this challenging Electrical Engineering topic.

I wish to show my gratitude to my Brazilian friends Antonio João C. R. Coutinho, Canisio Barth, Fábio Leal da Silva, Heitor Albuquerque B. Q. Gonçalves, José Francisco de Andrade

Jr, and Leonardo Guimarães Ribeiro, who counseled me and gave me recommendations throughout my time at NPS.

I wish to show my gratitude to the friends that I made at NPS: Diego Rangel, Dimitrios Mastrantonis, Elias Nadaf, Evan Shorter, Francesco Cochella, Ken Fletcher, Marcea Ascencio, Mariano Negron, Mike Wakeland, and Konstantinos Paschalidis, friends who made me feel better during difficulties. Working together made it possible to achieve success in my master's degree.

Finally, I wish to dedicate this work and my degree to my wife, Juliane Alves da Silva Martins, and my parents, Alcimar de Andrade Martins and Jorge T. Azeredo Martins, for all the patience and support they gave to me even before my coursework started, listening to my anxious feelings, helping in my difficulties and making me feel sane during a turbulent time in the world.

CHAPTER 1:
Introduction

The employment of unmanned aerial vehicles (UAV) is prevalent in modern warfare. Furthermore, multiple drones working together to accomplish a task, also known as a swarm, can help save lives; however, the communication among drones within a swarm is a challenge with the available technology. This challenge is due mainly to the limited electrical power available to operate all sensors and electronics in a small device.

One possible solution to this problem is to use the communication channel for exchanging data containing relevant information about the drones within a swarm.

This work details computer simulations to evaluate the practical applicability of a Zadoff-Chu (ZC) sequence, extensively used in 5th generation mobile network (5G) communications, to carry information about the identification and position of drones in a swarm.

1.1 Background

Modern warfare took a major step forward in the protection of lives when drones were used in operations for the first time. In a context where the main purpose is to protect a nation's interest and lives, the future of war will be defined by the level of technology applied in UAV in service.

The operation of a swarm of drones, instead of just one drone, would increase the range of missions that could be accomplished, such as by increasing payload carried into and out of battle. In such missions, drones would most likely be operating in environments embedded in noise, and drones are susceptible to a shift in frequency caused by the Doppler effect.

This thesis work analyses the use of the preamble, located in 5G communication packets, that in this case would carry a ZC sequence with identification and position information regarding a reference drone. Most specifically, this research establishes a method to get that information by applying a matched filter (MF).

Matched filters are widely used in radar systems, employed primarily for detection. Here,

the well-known cross correlation, when applied to a ZC sequence and samples containing this same ZC sequence, acts as a matched filter. The results obtained for this filter exhibit a peak exactly at the point where the ZC sequence starts at the sample vector. Given the expected results for the matched filter, it is possible to evaluate the influence of the noise and the Doppler effect by changing these parameters when the drones are operating in scenarios with these adversities.

1.2 Related Work

The employment of drones in combat zones is relatively novel, starting during the Cold War [1], and modern studies have investigated the applications of many drones working together and how efficient they can be as a swarm [2].

Qiannan Cui, Peizhi Liu, Jinhua Wang, et al., in their 2017 study [3], provided an approach for trying to determine the best network to manage mesh swarms: mobile ad hoc networks (MANET) or vehicular ad hoc networks (VANET). Their approach separated the swarm into small groups, with each group having a mother drone that manages the communication with other drones' groups.

The research of Luji Cui, Hao Zhang, et al. [4], in 2015, showed promising results of estimating range using a 60 GHz orthogonal, frequency-division multiplexing (OFDM) system with the Guard Interval as a communication channel. Also in 2015, Vincent Savaux and Faouzi Bader [5] implemented a mean-square error (MSE) based method to analyze the performance of the OFDM channel. Finally, Min Hua, Mao Wang, et al. [6] analyzed the Doppler effect in the timing performance of a ZC sequence in their 2014 work.

The work presented in this thesis is a combination of all the research just mentioned, using MSE plots to analyze the performance of ZC sequences as a way to identify and range drones operating in a swarm, within a noisy and Doppler-affected environment.

1.3 Objective

In this thesis, we consider ZC sequences with a length of 839 symbols. A simulation algorithm was created to replicate real-world scenarios for the swarm. The parameters changed during the performance of the algorithm included the size of the swarms, the

position of the drones in the layout, the packet preamble, the load of noise, and the amplitude of the Doppler effect. The main purpose of this work was to manipulate those parameters, analyze the results, and select sets of the root index of a ZC sequence (R) as best or worst to mitigate errors in ranging unmanned aerial vehicles operating in a swarm.

1.4 Thesis Organization

In Chapter 2, we present the theoretical background of the mathematics and methods applied in the simulations. This math and methods background includes the basics of 5G, and the particularities of ZC sequences, as well as the applicability of the cross-correlation as an MF to identify and range UAVs in a swarm deployed in a noisy and Doppler affected environment. In Chapter 3 we discuss the scenarios that were simulated and their algorithms, showing and explaining the reasons why the parameters that were changed were changed, concluding with the performance results and whether they showed what was expected. In the conclusion, Chapter 4, we provide a brief summary and discussion of the results of simulations and propose some ideas for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
UAV Communication System

The communication among drones is the core of this research because ultimately, poor communications will prevent a swarm of drones from being effective. In this chapter we explain the features of a 5G-inspired communication system for drones and, at the same time, explore the effect of the different factors in this channel.

The math behind the methods used in this thesis, and some assumptions made for drone distributions and their positioning, are also provided in this chapter.

2.1 Basics of 5G Communication

A possible solution for drone communication problem is the new 5G technology, which provides communication between a large number of devices using the feature known as massive machine-type communication (mMTC), which boasts low cost and low energy consumption [7], necessary features in a dense environment of drones. A swarm of communication drones is shown in Figure 2.1.

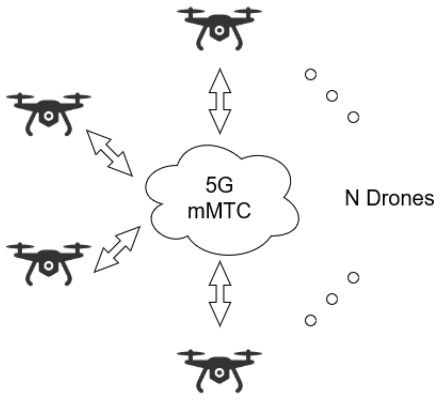


Figure 2.1. Drone communication using 5G mMTC.

Considering that 5G uses OFDM, the information is divided into several slots, called symbols, in both the time domain and frequency domain, as shown in Figure 2.2.

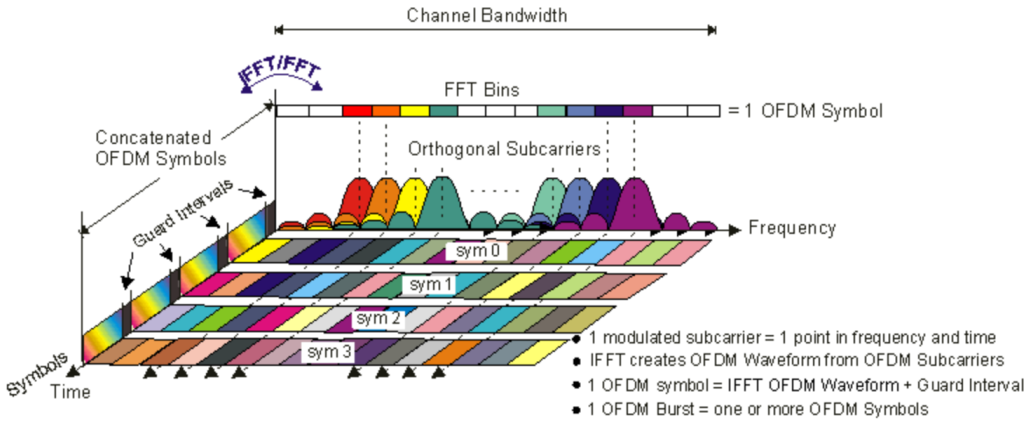


Figure 2.2. Time-frequency representation of OFDM. Source: [8].

The concatenation of symbols in the time domain for 5G is given by a sequence of Preamble, Symbol (or symbols), and Guard Interval. The Preamble is also known as the physical random-access channel (PRACH), and it can be located in Figure 2.2 as the sym 0 sequence. Depending on its format, the 5G preamble may have two supported sequence lengths: a short length $L_{RA} = 139$ and a long length $L_{RA} = 839$ [9]. Drones sending preamble information to a 5G mMTC cloud are shown in Figure 2.3. In a real situation, the preambles are not synchronized, and there is noise and Doppler effect associated with this signal.

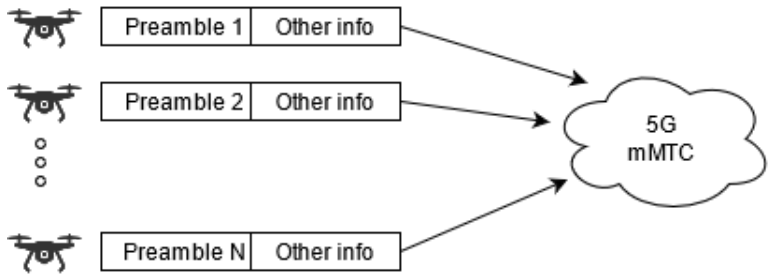


Figure 2.3. Drones sending the preamble information to a cloud network.

The preamble in the 5G communication protocol provides synchronization information to the uplink [7]. Since the drones do not need synchronization, the preamble can be used for another purpose – for instance, containing a sequence with the identification information of each drone. Nonetheless, it needs to be determined which kind of number sequence sent as the preamble would provide a good signal-to-noise ratio (SNR) feature and simultaneously provide ranging information for the drone sending this information. The answer to this question is the ZC polyphase sequence, a good option since it has ideal properties for this application.

The ZC sequence was implemented in the preamble field in this research, and how the information for range and identification is included is explained in the next sections.

2.2 Polyphase Sequences

The core of the development of this thesis is a very specific family of sequences, the Polyphase Sequences. These kinds of sequences have some specific correlation properties that make them very useful in several places in 5G communications [7].

A given sequence $S = b_0, b_1, ..., b_{n-1}$ is a polyphase sequence if every number in the sequence is a complex N^{th} *root of unity*, that is, if there exists a positive integer N such that each $|a_k^N| = 1$, where $x \in \mathbb{C}$ [10].

$$b_n = e^{\frac{-jn2\pi}{N}} \tag{2.1}$$

where:

- n is the sequence position for $0 \leq n < N$, and
- N is the sequence length.

2.2.1 The Zadoff-Chu Sequence

Initially studied in 1972 by David C. Chu [11], the ZC sequence is a family of polyphase sequences commonly used as the sounding reference signals (SRS) in 5G communications. The ZC sequences are particular cases of polyphase sequences, where the factor n is

separated into two factors, m and p_n, where m is any integer co-prime with an odd number N, and p_n is the n^{th} pronic number: $n(n+1)$.

A ZC sequence is given by [4], [7], [11]:

$$a_n = \begin{cases} e^{\frac{j\pi p_n n^2}{N}} = e^{\frac{-j\pi R n^2}{N_{ZC}}}; 0 \leq n < N_{ZC}, N_{ZC} \text{ even} \\ e^{\frac{j\pi p_n m(n+1)}{N}} = e^{\frac{-j\pi R m(n+1)}{N_{ZC}}}; 0 \leq n < N_{ZC}, N_{ZC} \text{ odd} \end{cases} \tag{2.2}$$

where:

- m is some positive integer that is co-prime with N;
- N is odd; and
- p_n is n^{th} pronic number (defined as $n(n+1)$) for $0 \leq n < N_{ZC}$.

Since p_n, or, as referred to starting here as R, defines the generation of R^{th} root ZC sequence with length N_{ZC} [12], R can provide an identification for the drones. The preamble length upper limit, $L_{RA} = 839$, allows the identification of 839 individual drones and represents the maximum swarm size.

These sequences were initially introduced by Robert Frank, in 1962 [13]. The construction of the sequence was initially described as follows:

$$\mathbf{M} = \begin{bmatrix} 1 & 2 & 3 & \cdots & N \\ 2 & 4 & 6 & \cdots & 2N \\ 3 & 6 & 9 & \cdots & 3N \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ N & 2N & 3N & \cdots & N^2 \end{bmatrix}. \tag{2.3}$$

The numerals in Equation 2.3 work as basic phase angles and can be defined as: $2\pi m/N$ (m is co-prime with N). The sequence is then created by:

$$a_n = e^{j\frac{2\pi m}{N}} \times \mathbf{M}_{\lceil n/N \rceil, mod(n,N)}. \tag{2.4}$$

The drawback of Frank's sequence is that it is restricted to a length that must be a perfect square. David Chu extended this idea to a code of any length N, demonstrating mathematically a valuable property of a ZC sequence, the constant-amplitude, zero autocorrelation (CAZAC) property [11], [13], [14]. (Note that although the preceding requirement is odd, Chu also provided a separate formula if N is even.) The sequence is said to be a constant-amplitude (CA) sequence if for any R it has a constant amplitude $|a_n| = 1$ [13], as shown in Figure 2.4.

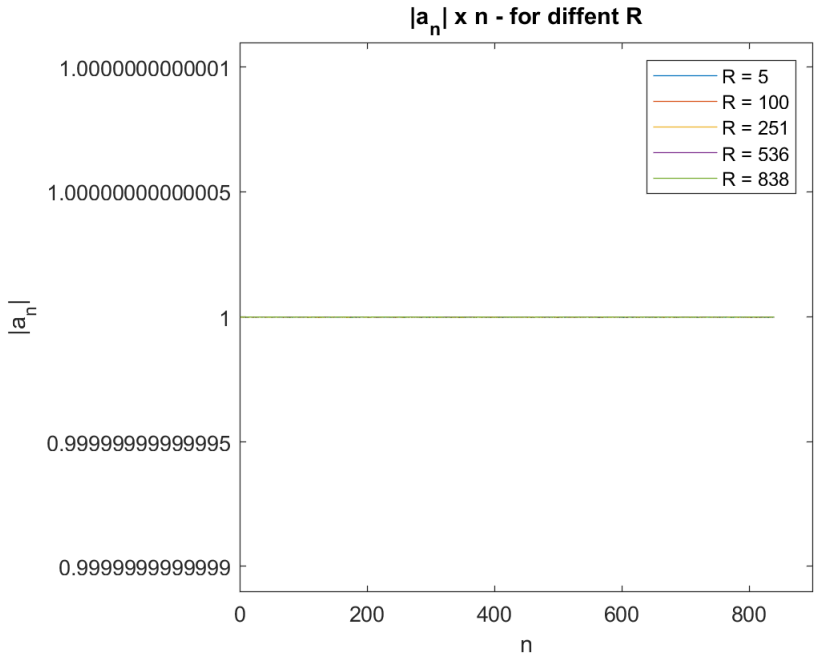


Figure 2.4. $|a_n|$ x n for various values of R.

A good reason to use CA sequences is that they provide accurate channel estimation [15] and have a power-saving feature very useful in a swarm of drones.

On the other hand, a given sequence is said to be zero autocorrelation (ZAC) if it has an ideal periodic autocorrelation [14]. A large autocorrelation in the distance from zero to a series of numbers (lag) $k = 0$, which is a demonstration for the ZAC property of the ZC sequence is shown in Figure 2.5. In the next section, we explain this property of ZC sequences as well

as explain why it is important in this work.

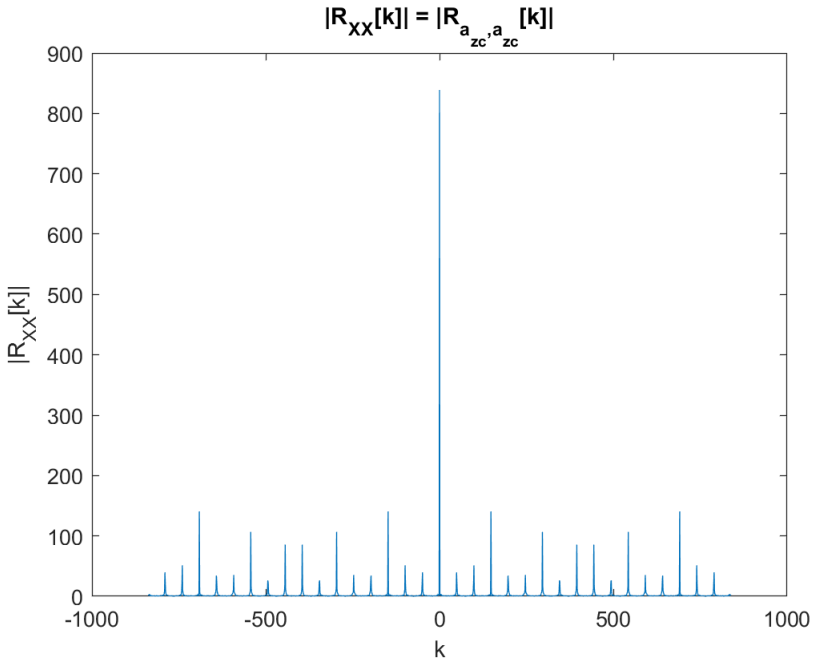


Figure 2.5. Auto-correlation R_{XX} of a ZC sequence.

2.3 The Matched Filter Application

"The necessary and sufficient condition for a sequence to be CAZAC is that its discrete Fourier transform (DFT) have constant amplitude" [14]. As is known, the DFT of a constant is a delta function [16]:

$$\mathcal{F}\,(1) = \delta(n). \tag{2.5}$$

Since the ZC sequences have a constant amplitude equal to one, the DFT of the sequence is a delta function located at a lag equal to zero. This shows an important feature of a ZC sequence, which is its zero circular autocorrelation, since

$$\mathcal{F}\left(\mathcal{F}^{-1}\left(a_n\right) \times \mathcal{F}^{-1}\left(a_n\right)^*\right) = a_n \circledast a_n \tag{2.6}$$

is true.

It is also known that the linear cross-correlation is given by [17]:

$$R_{XY}\left[k\right] = E\{x\left[n\right] y^*\left[n-k\right]\} = \sum_{n=-\infty}^{\infty} x\left[n\right] y^*\left[n-k\right] \tag{2.7}$$

where, in this research:

- $x[n]$ is a ZC sequence;
- $y[n-k]$ is a larger vector with $x[n]$ inserted in it; and
- k is the position, or lag, where $x[n]$ is located in $y[n]$.

When linear cross-correlation is applied to a ZC sequence and a larger vector containing this sequence, the cross correlation works as a matched filter [18], matching the position where the ZC sequence was inserted, placing a delta with amplitude $N_{ZC} = 839$ (Figure 2.6).

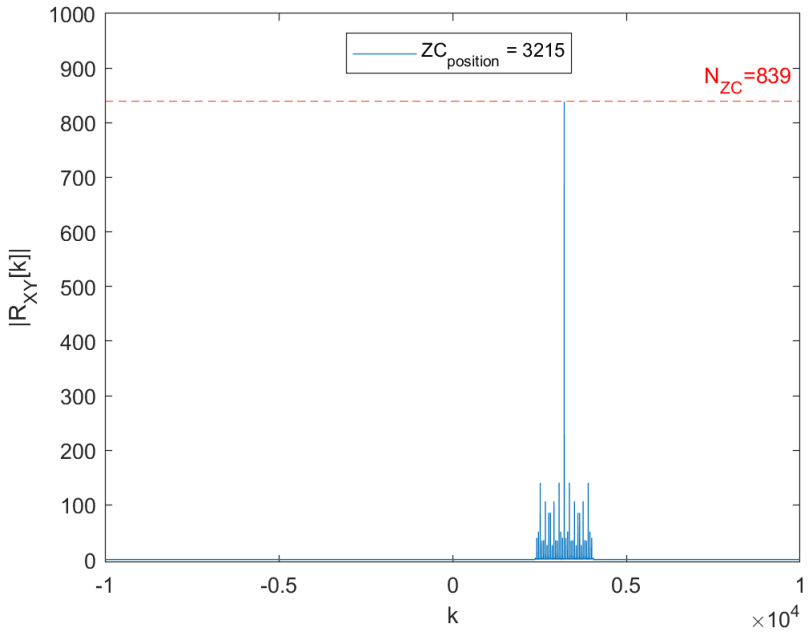


Figure 2.6. R_{XY} of a ZC sequence and a vector of zeros containing this same sequence.

This peak of amplitude 839, in an environment without noise or Doppler effect (elements added to our environment in the next sections), will always be the greatest peak identified in the resultant linear cross correlation.

2.4 UAV Swarms Communication

Knowing that the maximum peak of the cross correlation R_{XY} is located at the lag representing the position of a drone, an algorithm can be applied to calculate this position in terms of distance, for example.

2.4.1 Ranging the Drones

Suppose a drone A receives the preamble transmitted by another drone B. Suppose also that drone B has an embedded algorithm able to apply the cross correlation between B's known ZC sequence of length $N_{ZC} = 839$ and the sample received; therefore, we expect that the

resultant R_{XY} contains a maximum value exactly at the distance between the drones A and B in terms of the lag k. Therefore, the distance between A and B can be determined by:

$$Distance_{A,B} = f\left(Max\left(R_{XY}\left[k\right]\right)\right) = f\left(Max\left(R_{a_{ZC},S}\left[k\right]\right)\right). \tag{2.8}$$

It is important to note that this work starts from here to identify $x[n]$ as the ZC sequence $a_{ZC}[n]$ and the larger sample containing a_{ZC}, $y\left[n\right]$, as $s\left[n\right]$.

2.4.2 Identifying the Drones

As said previously, the ZC sequence can be used to identify the drones in the swarm. Given the factors N_{ZC} and R, the sequence of numbers is always the same. Considering the length of the sequences fixed in $N_{ZC} = 839$, the only variable factor in the sequence is the root index R. Since R should be in the interval $0 < R < N_{ZC}$, there are just 838 possible ZC sequences of length $N_{ZC} = 839$. Given that fact, the drones can have a memory bank with all sequences used in the swarm, where the root index R identifies each drone. After a sample sequence is received, the drone would apply an internal algorithm for the linear cross correlation $R_{a_{ZC},S}[k]$ and identify the drone by applying the correct a_{ZC} for the specified R since the maximum peak occurs only for the specific ZC sequence.

In Figure 2.7, we see a signal containing two hypothetical drones, identified by the root indices $R = 17$ and $R = 19$, located at samples 3,215, and 4,702. Applying the cross correlation between the samples $s[n]$ and the ZC sequence a_{CZ} with $R = 17$, we can identify the position of the drone at sample $3,215$. The same method also works for another drone, identified by $R = 19$ at sample $4,702$.

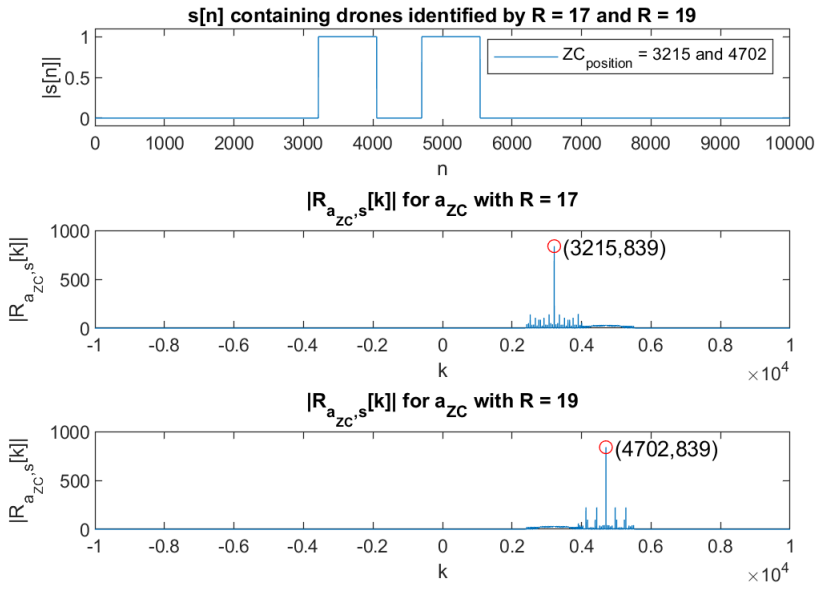


Figure 2.7. Drone identification.

2.4.3 ZC Sequences Orthogonality

Another important characteristic of ZC sequences is their inherent orthogonality. This characteristic, if it exists, implies a low level of interference between the devices using it, and this is one more favorable reason to use these sequences in this research. A sufficient condition to prove the orthogonality of two sequences is that the cyclic cross correlation between these two sequences has a constant value equal to $a_{ZC1} \circledast a_{ZC2} = 1/\sqrt{N_{ZC}}$ [19]. In this work $N_{ZC} = 839$; therefore, the constant value expected for the cyclic cross correlation is 0.0345.

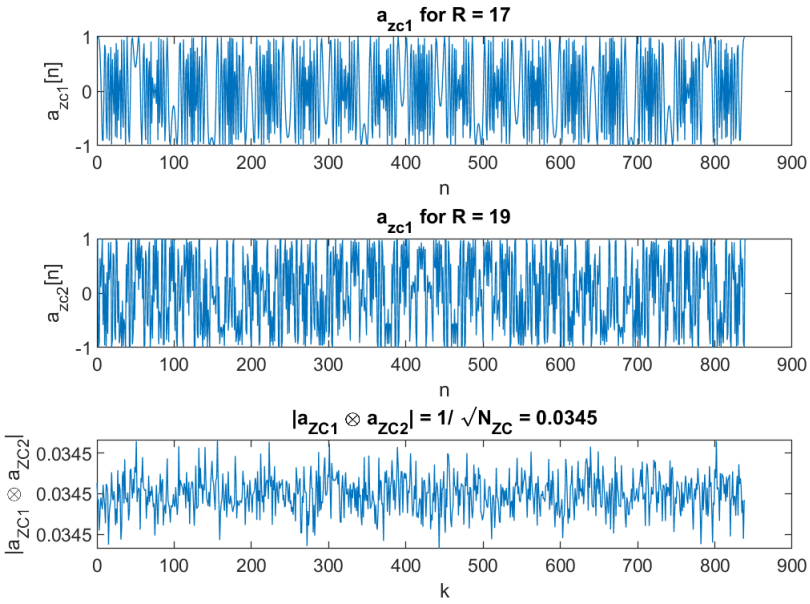


Figure 2.8. Orthogonality in ZC sequences.

The result in Figure 2.8 shows that ZC sequences are orthogonal to each other. Since the number of drones will be large, the orthogonality property is desired, considering the low interference level between the drones. Employing ZC sequences in the communication between the drones will provide low interference among them.

Given all the positive features of the ZC sequences – that is, they are CAZAC and orthogonal – these sequences are ideal for the application studied in this work. In view of the preamble as a possible identification candidate for each drone, this identification parameter can be used to find the location of the drones using a ZC sequence containing this information.

2.5 Noise and Doppler Effect in UAV Swarms

A perfect environment was defined for this research as one without noise interference or Doppler effect, but in a real situation, the swarm of drones would be exposed to many kinds of undesirable elements. In the next sections, we show some important aspects of these elements and explore their influence on the swarms analyzed here.

2.5.1 Noise in the Swarm

To account for the noise that would be present in a real environment, for this research we changed the SNR of the sample $s[n]$ by adding white Gaussian noise (noise $\in \mathbb{C}$). Initially, the noise was added directly to the sample using the MATLAB function $awgn()$. Unfortunately, this approach to add the noise does not account for the imaginary part of the signal, $\Im(s[n]) = b$, and the function $AWGN()$ adds pure real noise to $s[n]$, $\Re(s[n]) = a$. The SNR in this research assumes a signal power $P_{signal} = 1$ dB and the addition of the correct noise for both parts, real and imaginary, was made as follows:

$$\text{SNR}_{linear} = 10^{\text{SNR}/10} \tag{2.9a}$$

$$P_{noise} = \frac{P_{signal}}{\text{SNR}_{linear}} \tag{2.9b}$$

$$P_{noise-real} = P_{noise-complex} = \frac{P_{noise}}{2} \tag{2.9c}$$

$$\sigma = \sqrt{P_{noise-real}} \tag{2.9d}$$

$$Noise_{total} = \sigma a + j\sigma b \tag{2.9e}$$

where:

- SNR is a;
- a is a random vector of numbers distributed over the function $f(x) = \frac{1}{\sigma\sqrt{(2\pi)}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ of the length of $s[n]$; and
- b is another random vector of numbers distributed over the function $f(x) = \frac{1}{\sigma\sqrt{(2\pi)}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ of the length of $s[n]$.

Adding the value of the total noise to the signal $s[n]$, this sample vector has the correct noise distributed between real and complex parts.

$$s_{noisy}[n] = s[n] + Noise_{total} \tag{2.10}$$

2.5.2 Doppler Effect in the Swarm

Another undesired factor present in mobile communications environment is the Doppler effect. "It is well known that this effect occurs due to the relative speed between the elements in the communication system. The Doppler effect is directly proportional to the magnitude of the relative speed" [20]. In this work, the elements involved in the communication system are the drones, and since they are moving, and enclosed in the swarm, the Doppler effect, though small, affects the information sent between the drones.

The Doppler effect here was applied to the frequency, where a shift in frequency of the observed signal can be represented in the frequency domain as:

$$a_{ZC}(n) = r_n \cdot cos((w + w_d)n) = r_n \cdot cos(2\pi(f + f_d)n). \tag{2.11}$$

Since the signal is orthogonal, the difference between the frequencies cannot be greater than $\Delta f = f - f_d = 1/T_S$ [20], where T_S is the sampling period for the drones; therefore, it is possible to set some conditions for the Doppler effect applied to the signal $s[n]$:

$$f_{Doppler} = f_0 \cdot \left(\frac{v + v_0}{v - v_f}\right). \tag{2.12}$$

Setting $1/T_S = f_S = 839$ ksymbols per second (ksps), for a sequence of length $N_{ZC} = 839$, will be $T = 1$ ms. From Fourier theory, we see that the first null for a rectangular acquisition window is located at 1,000 Hz. This frequency is the worst case for the Doppler shift $f_{Doppler}$ that can occur for a $f_S = 839$ ksps.

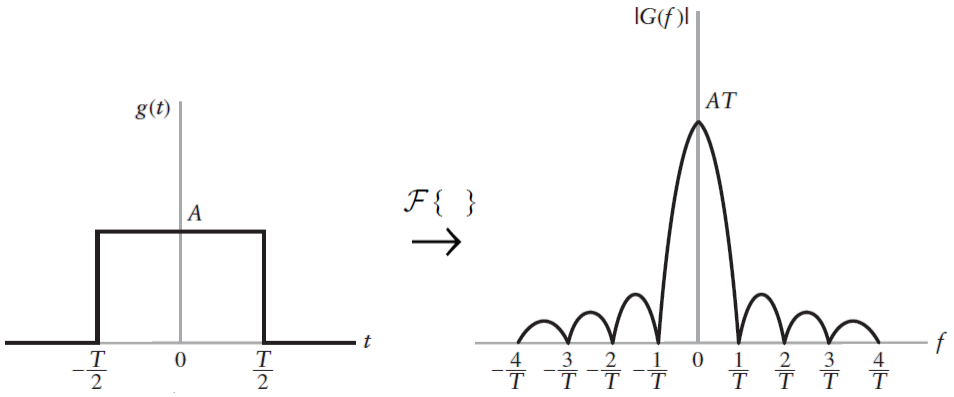


Figure 2.9. Fourier transform of a gate signal. Adapted from [21].

The frequency-shifting property for the Fourier transform is well known and is given by [16]:

$$e^{j\omega_0 n} \cdot a_{ZC}[n] \longleftrightarrow A_{ZC} \cdot (e^{j(\omega-\omega_0)}). \tag{2.13}$$

Combining the Doppler frequency $f_{Doppler}$ and the sample frequency f_S into Equation 2.2, using Equation 2.13 as support, we get:

$$a_{ZC_{Doppler}}[n] = a_{ZC}[n] \cdot e^{j2\pi \frac{f_{Doppler}}{f_S} n}. \tag{2.14}$$

Since the Doppler frequency is a function of the relative speed of the drone, as shown in Equation 2.12, and lamentably this speed will not be known, the worst case for the Doppler shift employed was $f_{Doppler} = 1,000$ Hz. Once this work adopted a $f_S = 839$ ksps, it became apparent that Equation 2.14 would be changed as in Equation 2.15. A factor α varying from 0 to 1 was assigned to Doppler shift. The best case of Doppler shift, 0 Hz, was assigned as $\alpha = 0$ and the worst case of Doppler shift, 1,000 Hz as $\alpha = 1$. A digital frequency α was used in this thesis as a Doppler factor:

$$a_{ZC_{Doppler}}[n] = a_{ZC}[n] \cdot e^{j2\pi \frac{1000}{839000} n} = a_{ZC}[n] \cdot e^{j2\pi \frac{\alpha}{N_{ZC}} n}. \tag{2.15}$$

The addition of Doppler effect to a ZC sequence a_{ZC} directly disturbs the result of the matched filter $R_{a_{ZC},s}[k]$. Depending on how large α is, side peaks appear next to the correct largest peak pointing to the correct distance ZC_{pos}. Now, the previous power for the matched filter is shared between the correct maximum and side maximums, which have amplitudes almost as large as the correct peak. In Figure 2.10, we see the presence of side peaks for matched results of two ZC sequences with $R = 19$, one of them with Doppler effect and another without Doppler effect.

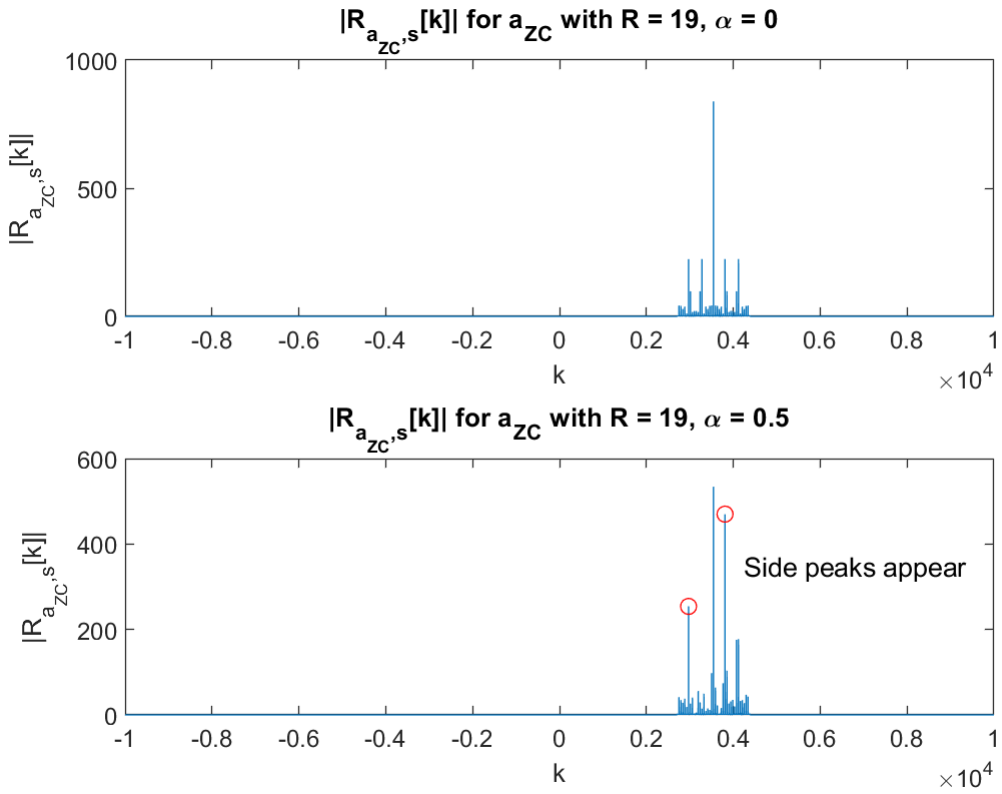


Figure 2.10. Comparison between matched filter results for ZC sequence without Doppler effect and ZC sequence with Doppler effect.

In Section 3.1.4, we explain how this power is shared as well as explain how the choice of the root index R can influence the MSE results.

2.6 The Monte Carlo Method

The Monte Carlo method is widely employed to solve mathematics problems of a probabilistic nature and evaluate the statistics of a given data base [22]. According to [23], "the analytical evaluation of a statistic is based on the theoretical development of its sampling distribution. Monte Carlo simulation offers an alternative to analytical mathematics for understanding a statistics sampling distribution and evaluating its behavior in random samples. Monte Carlo simulation does this empirically using random samples from known populations of simulated data to track a statistics behavior." Since this thesis is using random variables for some scenarios, as will be explained in Section 3.1, when some parameters will be changed over statistical distributions, it is a reasonable approach to use the Monte Carlo method, where for each iteration, a new set of random data is created.

A simplified algorithm of the Monte Carlo method applied in this work, where K represents the number of trials running in the simulation, is shown in Figure 2.11.

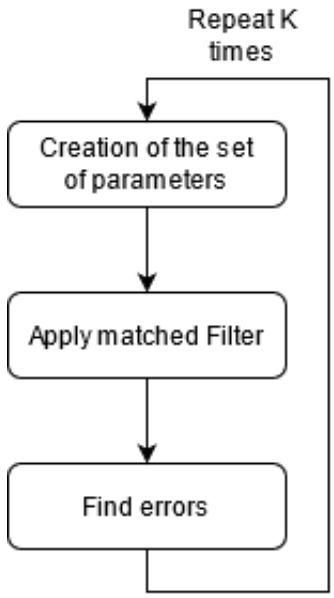


Figure 2.11. Monte Carlo algorithm.

2.7 Ranging Accuracy

As to possible errors in the communication between the drones, because this work considers $s[n]$ embedded in a noisy and Doppler affected environment, a valuable method to estimate the ranging accuracy is to verify the number of errors that occur after employing the matched filter. A satisfactory approach to evaluate the performance of the ranging estimator is applying an MSE algorithm for every K trials.

The procedure of acquisition of MSE via the differences between correct distances $\hat{k}_n$ and estimated distances k_n, used in this research, is given by:

$$\text{MSE} = \frac{\left(\hat{k}_1 - k_1\right)^2 + \left(\hat{k}_2 - k_2\right)^2 + \ldots + \left(\hat{k}_D - k_D\right)^2}{K} \tag{2.16}$$

where:

- D is the number of drones in the simulation;
- $\hat{k}_n$ is the distance estimated by the matched filter between the reference drone and drone $R\#$;
- k_n is the correct distance between the reference drone and drone $R\#$; and
- K is the total number of trials.

2.7.1 Ranging for Real Applications

In a real application, the range obtained from the method studied in this work would be a bi-dimensional distance. This distance could be obtained considering the propagation speed as the speed of light, $c = 3 \times 10^8$ m/s, and a maximum symbol rate of $SR = 16,800,000$ symbols/s for 5G applications [24]. With these parameters for speed and symbol rate, the minimum distance resolution for this method is:

$$d_{min} = \frac{c}{SR} = \frac{3 \times 10^8}{16,800,000} = 17.85 \text{ m} \tag{2.17}$$

It should be noted that the distance resolution of Equation 2.17 may be insufficient for some applications. For such cases, it may be necessary to utilize a different communications

standard (i.e., not 5G) in order to meet requirements.

CHAPTER 3:
Computer Simulations

In this chapter, particular aspects of the simulations performed for this thesis are described, including which parameters out of the possible set of parameters were used, and how they were changed. At the end of this chapter, we show the results obtained and provide a brief discussion.

3.1 Scenarios of Unmanned Aerial Vehicle Swarms

The universe of possibilities is very extensive when designing applications for unmanned aerial vehicles. In this work, scenarios with swarms of drones were planned using the 5G OFDM preamble to calculate the distance information [9], which is used to calculate the distance to a reference drone. The proposal here was to find a range estimation where every drone in the swarm could compute its distance from other drones. This research focuses on the number of errors, given by the MSE, in the distance between the drones, evaluating the distance information after changes in the parameters of the scenarios and ZC sequences.

To aid in the understanding of the scenarios, it should be noted that one drone, receiving the preambles of all other drones in the scenario, is considered as the point-of-view of a hypothetical reference drone. The other drones, which are sending the preambles, are referred to here as *R*#, indicating that they have a different root index, not necessarily the number indicated by #; therefore, a robust vector of samples $s[n]$ containing $10,000$ samples, initially a vector of all zeros, and all the ZC sequences for each drone *R*# were inserted into $s[n]$. In Figure 3.1, we see the aspect of a ZC sequence with $R = 10$ generated in MATLAB, and by randomly inserting this sequence into a vector $s[n]$ of 10,000 samples, it is possible to verify the ZAC property evident in Figure 3.1.

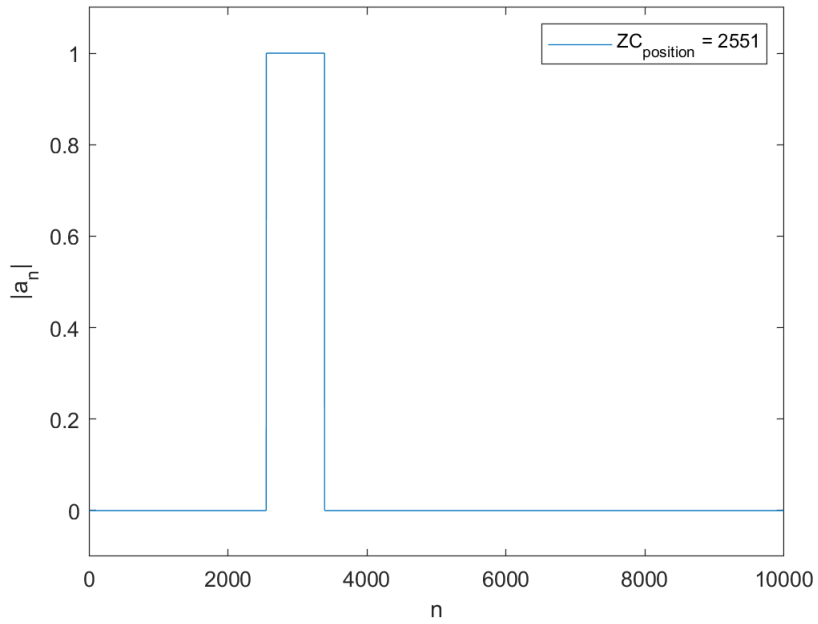


Figure 3.1. MATLAB created absolute values of a randomly positioned ZC sequence.

The information acquired from the preamble, given as a position in $s[n]$, is treated as the distance between the reference drone and the *R#* drone. With this distance data, the Monte Carlo iterative scenario is applied considering one hypothetical reference drone and D other drones (the existence of the reference drone is only hypothetical in this work). This means that the distance from the other drones to the reference drone, named ZC_{pos} in this work, is computer generated. Furthermore, the information is not actually being transmitted between drones. In a real scenario, all drones would receive information about all other drones in the swarm. A hypothetical example of how the ZC sequences of five *R#* drones would be randomly distributed over $s[n]$ is shown in Figure 3.2.

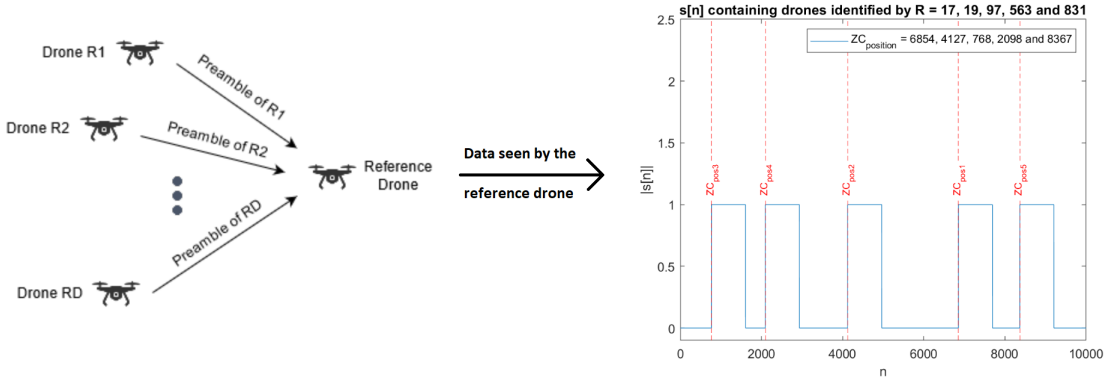


Figure 3.2. Reference drone receiving information from other drones ($D = 5$, without noise and Doppler effect).

In this work, four parameters were changed for the various scenarios of samples $s[n]$:

- Distance between the reference drone and other *R*# drones;
- Number of drones in the scenario;
- The noise and Doppler effect to which the channel is exposed; and
- Root index R for the ZC sequences.

The number of trials K were executed for each simulation performing a Monte Carlo experiment, as described in Section 2.6.

In addition, the length of the ZC sequence was established as $N_{ZC} = 839$, since the 3GPP TS 38.211 standard [9] allows PRACH preambles of prime length 139 and 839, as referenced in Section 2.1.

3.1.1 Parameter Changed: Distance between Reference and Other Drones

This work treats the distance between the reference drone and other drones as randomly uniform distributed values. For graphing purposes, this distance is identified as ZC_{pos}. The other drones, identified by *R*#, were distributed between the positions 1 and $N - N_{ZC} = 9,161$, since 839 free spaces should exist to write the ZC sequence. The inclusion of the ZC sequence of length N_{ZC} was done by inserting it in a random position of a data vector

with length N, as described in Figure 3.3. It is important to observe that the ZC_{pos} is the starting point of the sequence, and there are 838 points after it.

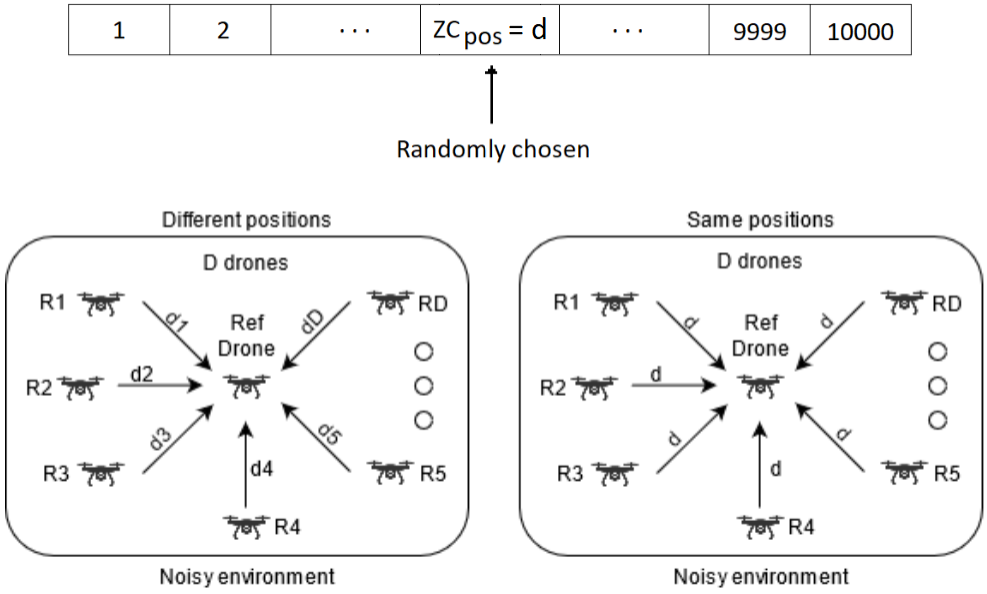


Figure 3.3. Drones in different ZC_{pos} or all drones in the same ZC_{pos}.

As studied in Section 2.7, the change in the environment could cause mismatches in ranging for the drones. Given these possible mismatches, two different cases for the positions were considered:

- D Drones in different positions; and
- All D drones in the same random position.

A characteristic $s[n]$ for $D = 20$, without noise or Doppler effect, with drones positioned in different random positions and positioned at a same random position, is shown in Figure 3.4.

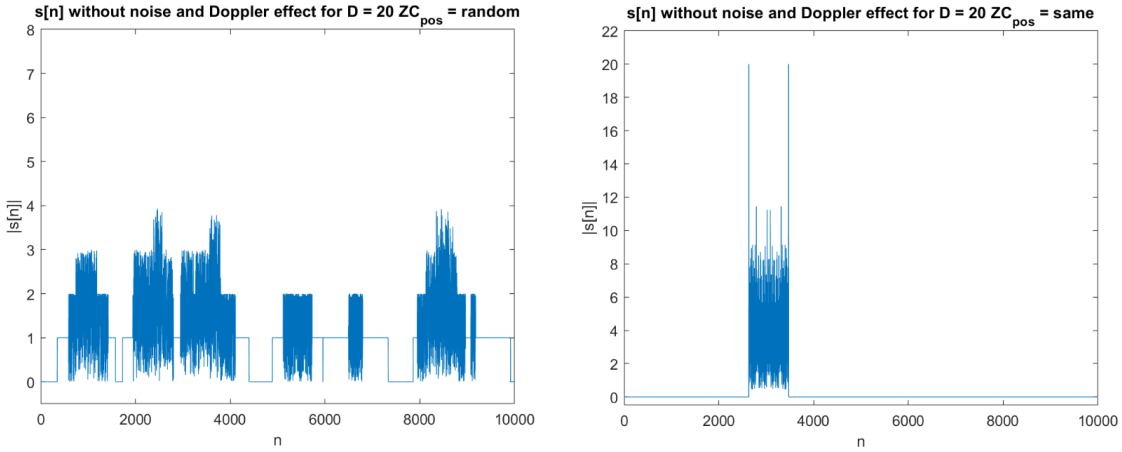


Figure 3.4. $|s[n]|$ within drones positioned in different ZC_{pos} and positioned at the same random ZC_{pos}.

3.1.2 Parameter Changed: Number of Drones D

The number of drones D in the simulation environment was changed to the following quantities: 1, 2, 4, 5, 10, 20, 50 and 100. It is expected that increasing the number of drones in the scenario affects the error count and, consequently, the MSE given by Equation 2.16. How the UAVs were arranged for simulations performed in this research is shown in Figure 3.5

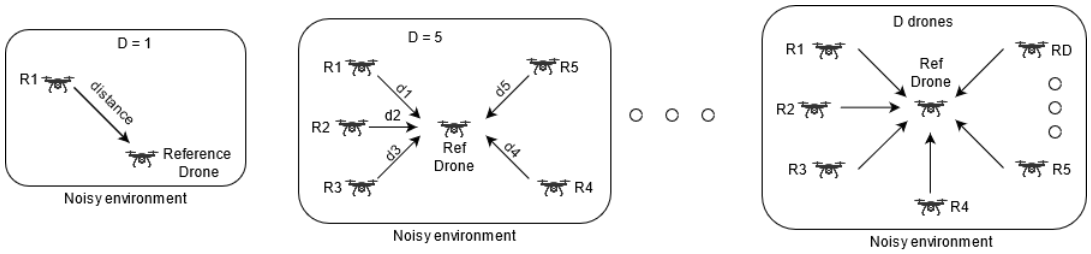


Figure 3.5. Graph representation of swarms of UAVs.

This research does not exclude the possibility that the same ZC_{pos} could be allocated to more than one drone $R\#$ or would overlap samples of others drones ZC sequence. This

method allows some drones to have the same distance from the reference drone.

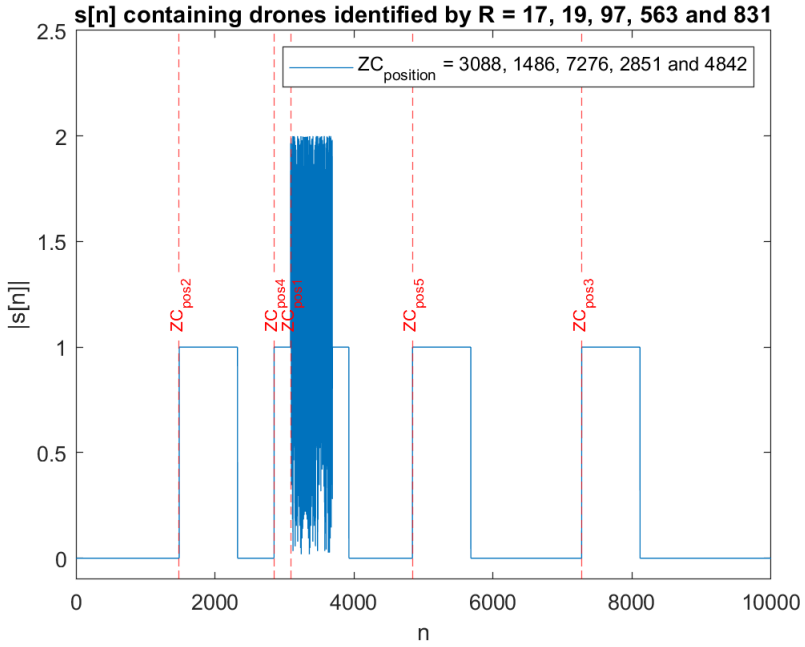


Figure 3.6. Overlap of ZC sequences.

A scenario with $D = 5$, where drones $R1$ and $R4$ are seen by the reference drone almost at the same distance, caused an overlap of the two sequences, as shown in Figure 3.6.

The number of drones in the scenario was considered for the MSE Equation 2.16 calculations since the greater the number of drones in the scenario, the greater the multiplication factor for the errors. To mitigate this multiplication factor, this work considered a division by D for every MSE calculation, as shown in the revised equation for MSE.

$$\text{MSE} = \frac{\left(\hat{k}_1 - k_1\right)^2 + \left(\hat{k}_2 - k_2\right)^2 + \ldots + \left(\hat{k}_D - k_D\right)^2}{K \cdot D}. \tag{3.1}$$

3.1.3 Adding Noise and Doppler Effect to *s[n]*

In addition, in trying to make the simulations close to the real environment observed for the swarms, noise and Doppler effect were added to $s[n]$. For all scenarios and simulations, $s[n]$ was initially generated as a vector of zeros, and then the ZC sequences and noise were added.

The method of addition of the Doppler effect to a ZC sequence $a_{ZC}[n]$ is shown in Section 2.5.2. Since similar speeds are expected for all drones in the swarm, the factor α (Equation 2.15) was randomly chosen from $[0, 0.05, 0.1, ..., 0.45, 0.5]$, considering a maximum Doppler frequency of $0.5 \cdot f_{Doppler}$. The Doppler effect was added individually to each drone, and then each a_{ZC} was added to $s[n]$ in a randomly chosen ZC_{pos} position.

On the other hand, and different from the procedure to add the Doppler effect, the entire $s[n]$ were embedded in noise. For each number D of drones, an SNR was chosen from the following: -10, -5, 0, 5, 10, 15 and 20. For each Monte Carlo trial, one SNR was selected from this set, and we expected that changes in SNR would affect the number of errors and the MSE.

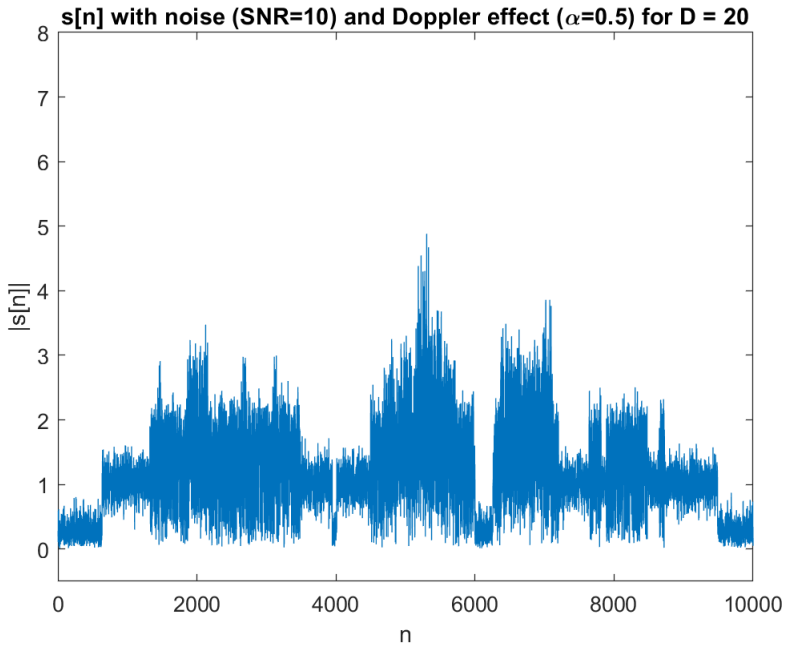


Figure 3.7. $s[n]$ with noise and Doppler effect for $D = 20$.

3.1.4 Parameter Changed: R

As seen in Equation 2.2 and in Sections 2.3 and 2.5.2, the parameter R, the root index of the ZC sequence, can be changed. This change, especially when a_{ZC} is affected by Doppler, directly affects the output of the matched filter. Notice that each drone has a tag R; the matched filter output of the noisy and Doppler affected data, with the expected ZC sequence, changes depending on the R selected.

After simulations where SNR is zero and Doppler effect $\alpha = 0.5$, it was possible to see that the amplitude and distance between the correct peak and side peaks is deterministic, always being the same for a given R. As a matter of fact, this distance between the correct maximum peak and the second maximum peak is given by the greatest common divisor (GCD) between the investigated R and the length N_{ZC}:

$$Dist_{2\_peaks} = |GCD(R, N_{ZC})|. \tag{3.2}$$

In Figure 3.8, we see how the distance between two consecutive peaks of the matched filter change for a specific R. An evaluation of the errors from three perspectives enables us to define the best set of Rs or worst set of Rs possible:

- Distance error between two peaks;
- Difference between the amplitudes of two peaks; or
- Vector distance between the amplitudes of two peaks.

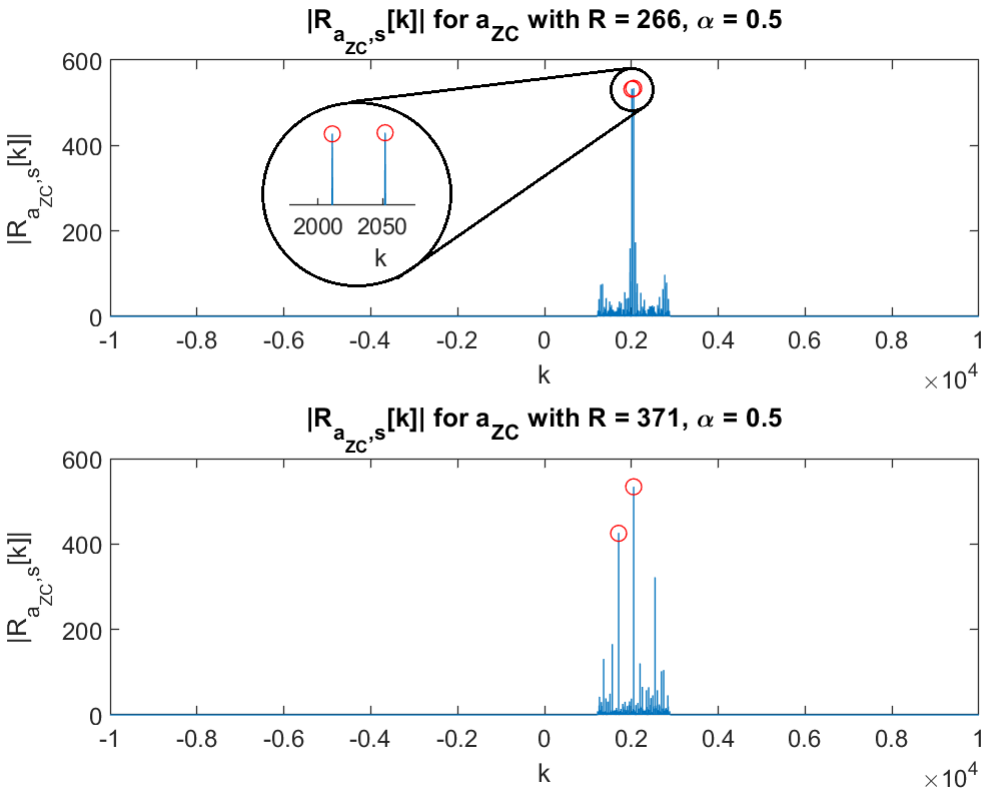


Figure 3.8. Changes in the distance between two maximum peaks due to changes in R, for $\alpha = 0.5$.

We also observe in Figure 3.8 that the greater the distance error, the smaller the amplitude of the side peak and the greater the amplitude error. On the other hand, the smaller the distance error, the greater the amplitude of the side peak and the smaller the amplitude error. The

various perspectives on the errors are shown in detail in Figure 3.9.

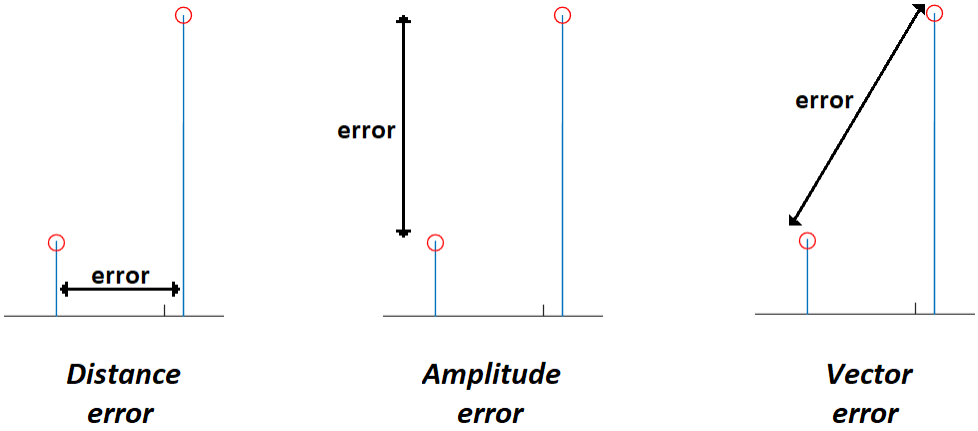


Figure 3.9. Error perspectives for selection of best and worst Rs.

Since the aim of this work is to obtain errors in the distance between drones, the error used here was the one from the perspective of the distance between two maximum peaks. Given the length of the ZC sequences is $N_{ZC} = 839$, the possible number of Rs is 838. Therefore, establishing a fixed ZC_{pos}, it was possible to calculate all the distance errors between the two maximum peaks of the matched filter results for all R, and Figure 3.10 was obtained.

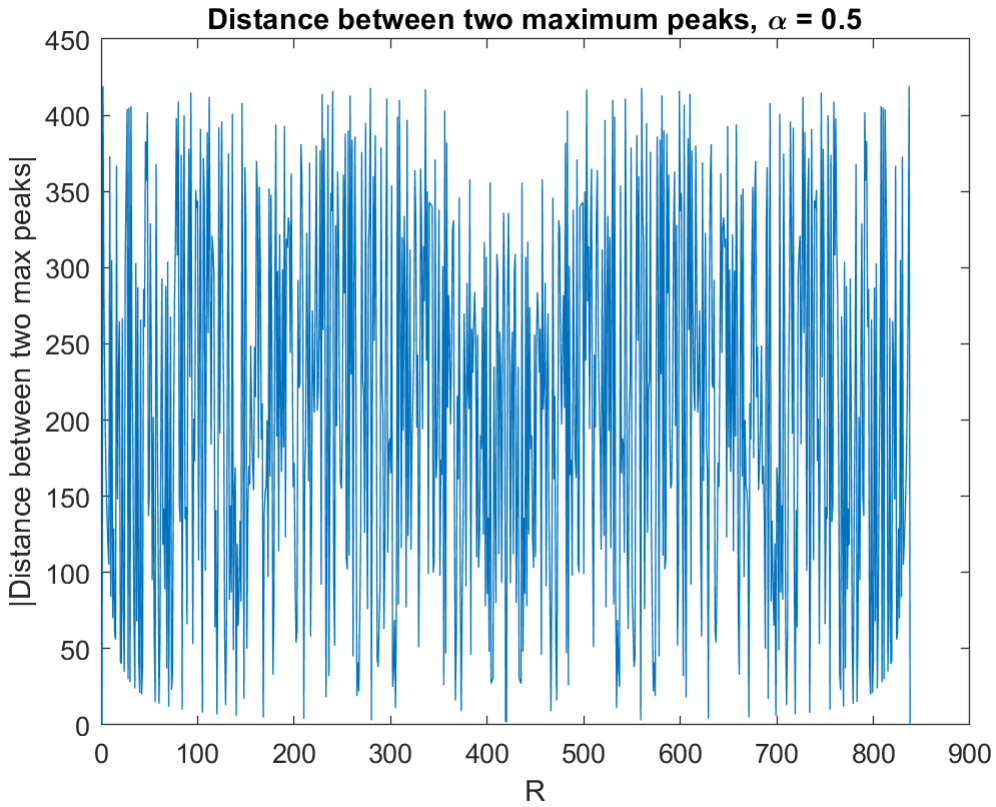


Figure 3.10. Error perspectives for selection of best and worst Rs.

The Rs were then sorted by the growth of the distance errors into a vector, and the best set of Rs was selected from the beginning of this vector, while the worst set of Rs was selected from the end of this vector.

Considering the selection just mentioned, three cases for the simulations were created:

- Simulations for random R;
- Simulations for best R; and
- Simulations for worst R.

These simulations consider the random R, bad R and good R selected from the range $[1, ..., 838]$, as $R = 839$ is not a valid root for $N_{ZC} = 839$, the case studied in this research.

3.2 Algorithm for Computer Simulations

Following the change in parameters as described in Section 3.1, six scenarios were created for this thesis project. These scenarios included changes for R and the distance between the reference drones and other drones, ZC_{pos}. Parameters Rs were selected in different sets as random, best or worst. The distance ZC_{pos} were selected in different sets as random or same position. The algorithm applied in this work selects the types of parameters to be used, and then creates the sample vector $s[n]$. Then, the algorithm performs the Monte Carlo trial, as shown in Figure 3.11.

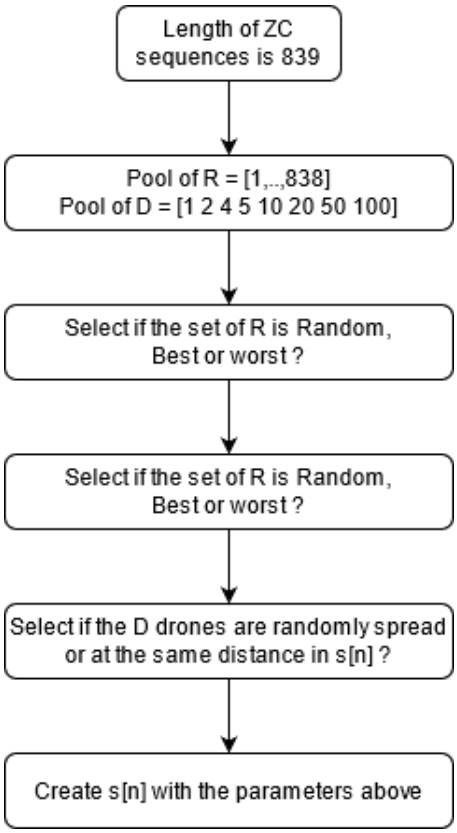


Figure 3.11. Flowchart with the preset algorithm.

Given these preset parameters, the result produced by this study was the MSE. Since this research has many random variables in the process of evaluation of the MSE, it was

necessary to use a Monte Carlo model for the simulations. In this case, every time a new set of variables was randomly created (for instance, if the previously defined parameters were changed) a new iteration was performed and the data for errors saved. Based on the data saved, the number of iterations defined here for the Monte Carlo method was $K = 1,000$ and the MSE was calculated using Equation 2.16. The flow chart of the Monte Carlo method applied to the preset data is shown in Figure 3.12.

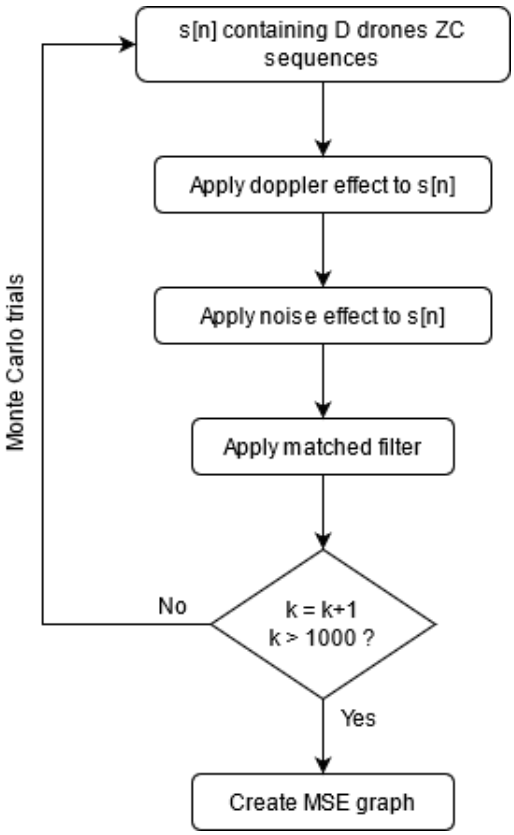


Figure 3.12. Flowchart with the Monte Carlo algorithm used in simulations.

The simulations in this thesis examine six different scenarios, changing the parameters R and ZC_{pos}:

- Simulations for random ZC_{pos} and R;
- Simulations for the same ZC_{pos} and random R;

- Simulations for random ZC_{pos} and best Rs;
- Simulations for the same ZC_{pos} and best Rs;
- Simulations for random ZC_{pos} and worst Rs; and
- Simulations for the same ZC_{pos} and worst Rs.

After all MSE calculations were complete for each scenario with D drones, a combined MSE plot was generated, combining all the SNRs and numbers of drones D.

3.3 Performance Results

The results obtained in this research employed $K = 1,000$ iterations for the Monte Carlo method and $s[n]$ with $N = 10,000$ samples. Three assumptions about the error behavior were created from the changed parameters for the number of drones D, Root index R, and positions ZC_{pos}:

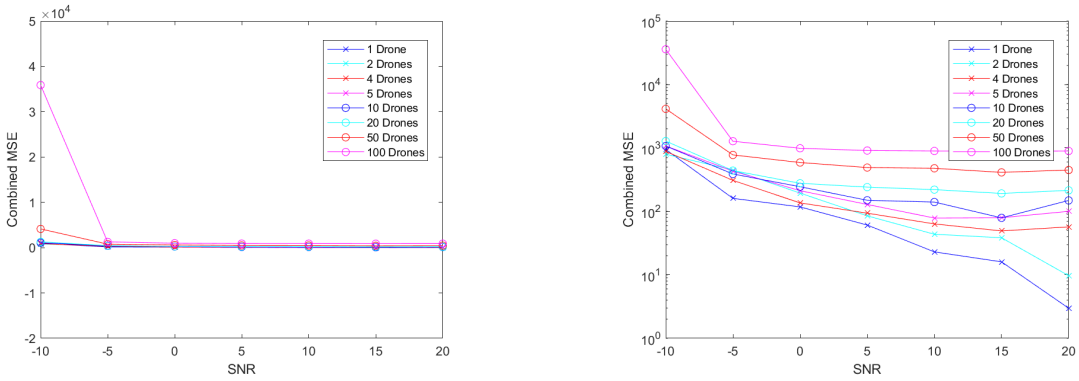
- Number of drones D: It was expected that since the number of samples N is fixed, the increase in the number of drones would imply more mismatches between the correct position and the received position. These mismatches would happen more often because of the addition of recurring noise, which raises the number of errors, and consequently increases the MSE;
- Root index R: The difference between the magnitude resulting from the matched filter for a different R when there is a large Doppler effect produces mismatches for the correct peak located where this is expected to find the position of the ZC sequence. Essentially, for a bad R, more errors are expected, which increases the MSE; and
- Positions ZC_{pos}: In this work, two positioning methods were considered: one case where every R# drone has a different random distance from the reference drone; and another case where every R# drone has the same random distance from the reference drone. A large number of errors is expected for the second case, since the noise will be applied in the same position and, thus, multiplied by D.

Note that a logarithmic function was applied to the resultant MSE when α is random. Since the MSE results in these cases have a large difference between a smaller number of drones, when compared with a larger number of drones, it seem more appropriate show the results in semi-log. Observe that the discontinuities are in fact MSE = 0: no errors occur; and as

long as $log\,(0) = \nexists$, semi-log plots ignore them.

3.3.1 Random ZC_{pos} and Rs

For the case of random ZC_{pos} and random R, the analysis of Figure 3.13 shows that higher levels of noise and low SNR produce many errors, even for a small number of drones.



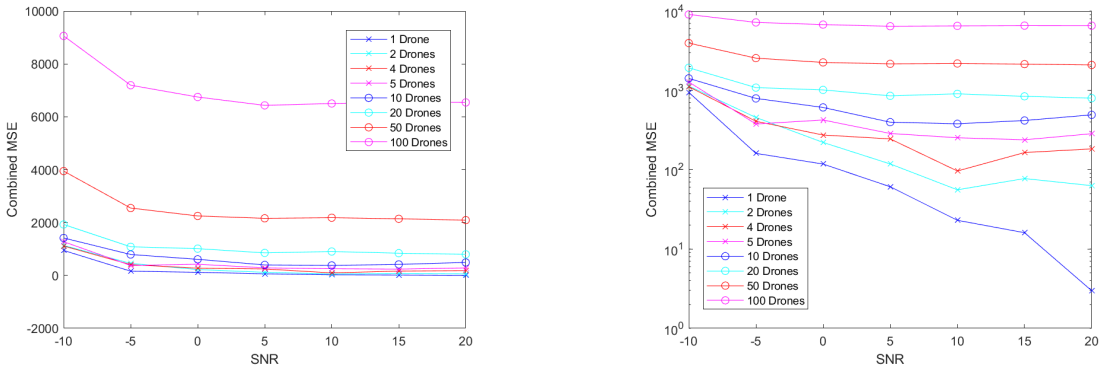
(a) Linear Combined MSE

(b) Logarithmic Combined MSE

Figure 3.13. Combined MSE for different D drones. In this case, each drone has a different R, and the drones are randomly positioned in a large vector.

3.3.2 Same ZC_{pos} and Random Rs

For the case of all ZC_{pos} located at the same position and random R, the analysis of Figure 3.14 again shows that higher levels of noise and low SNR, produce many errors, even for a small number of drones. Since all the drones are in the same position, and all the noise that affects the sequences a_{ZC} are added, it is possible to observe the increase of errors even for lower SNR.



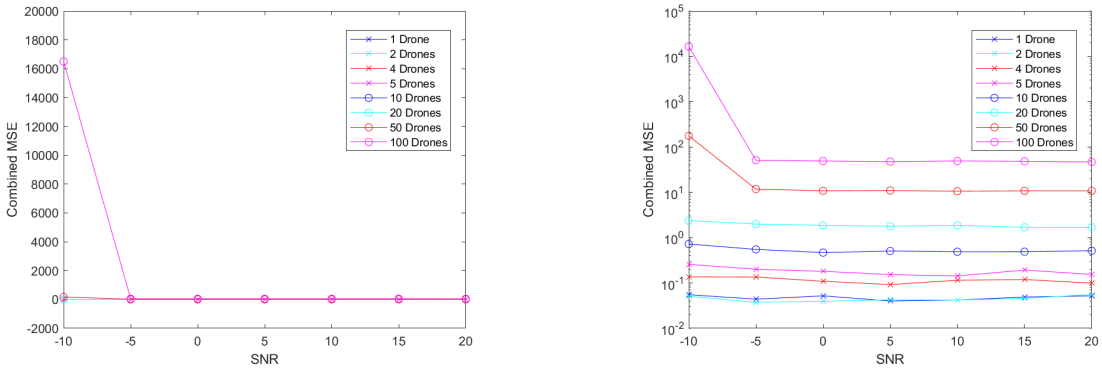
(a) Linear Combined MSE

(b) Logarithmic Combined MSE

Figure 3.14. Combined MSE for different D drones. In this case, each drone has a different R, and the drones are located in the same position in a large vector.

3.3.3 Random ZC_{pos} and Best Rs

For the case of random ZC_{pos} and best R, the analysis of Figure 3.15 shows again that higher levels of noise and low SNR produce many errors but now the count of errors is lower when the number of drones is small. For a large swarm of drones, $D = 50$ and $D = 100$, the number of errors is still large for lower SNR but decreases for higher SNR when compared with the scenario with random R. The errors obtained for small swarms, even those of small order, are due to the random Doppler-effect worst cases associated with errors count.



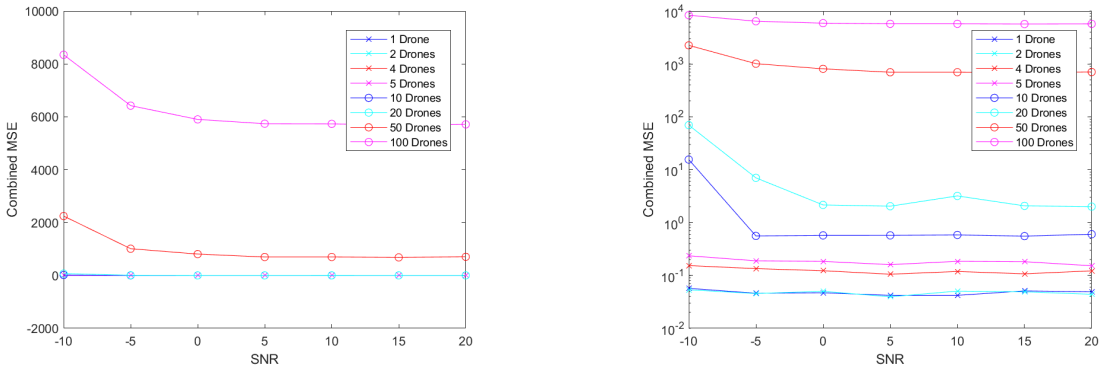
(a) Linear Combined MSE

(b) Logarithmic Combined MSE

Figure 3.15. Combined MSE for different D drones. In this case, the set of R selected is the best and the drones are located in different positions in a large vector.

3.3.4 Same ZC_{pos} and Best Rs

For the case of all ZC_{pos} located at the same position and best R, the analysis of Figure 3.16 shows a decrease in performance since all the drones are in the same position, especially for large SNR when compared with the scenario with ZC_{pos} randomly distributed. For a large swarm of drones, $D = 50$ and $D = 100$, the number of errors is still large for higher SNR.



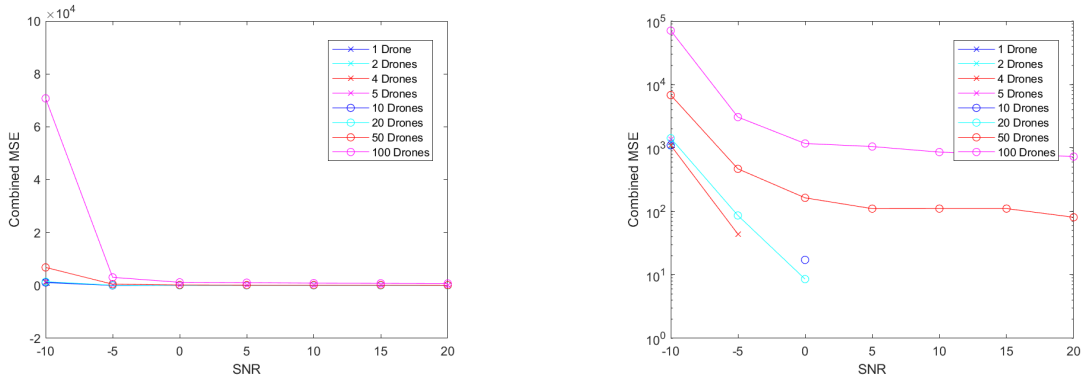
(a) Linear Combined MSE

(b) Logarithmic Combined MSE

Figure 3.16. Combined MSE for different D drones. In this case the set of R selected is the best and the drones are located in same position in a large vector.

3.3.5 Random ZC_{pos} and Worst *R*s

For the case of random ZC_{pos} and worst R, the analysis of Figure 3.17 shows an increase in performance for small swarms, even without the count of errors. For a large swarm of drones, $D = 50$ and $D = 100$, the number of errors is still large for higher SNR, although the errors simply disappear for small swarms. This is expected since the approach for this work considers the distance between peaks of matched filter results to find the best and worst *R*s. As far as the distance between these two maximums, the smaller is the secondary peak, which improves the performance for lower SNR.



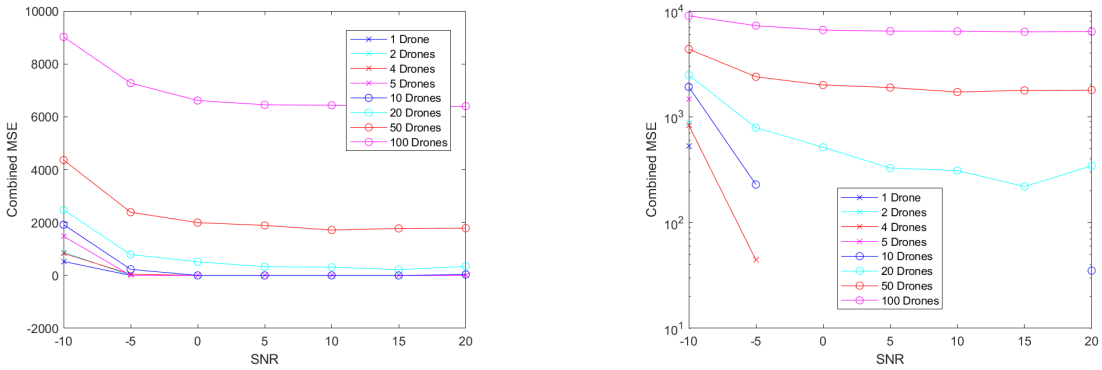
(a) Linear Combined MSE

(b) Logarithmic Combined MSE

Figure 3.17. Combined MSE for different D drones. In this case, the set of R selected is the worst and the drones are located in different positions in a large vector.

3.3.6 Same ZC_{pos} and Worst *R*s

For the case of all ZC_{pos} located at the same position and worst R, the analysis of Figure 3.18 shows an increase in performance for small swarms, even without counting errors. For a large swarm of drones, $D = 100$ and $D = 50$, the number of errors is still large for higher SNR, including now $D = 20$ swarms with a large count of errors, although the errors simply disappear for small swarms. Again, this is expected since the approach for this work considers the distance between peaks of matched filter results to find the best and worst Rs. As far as the distance between these two maximums, the smaller is the secondary peak, which improves the performance for lower SNR.



(a) Linear Combined MSE

(b) Logarithmic Combined MSE

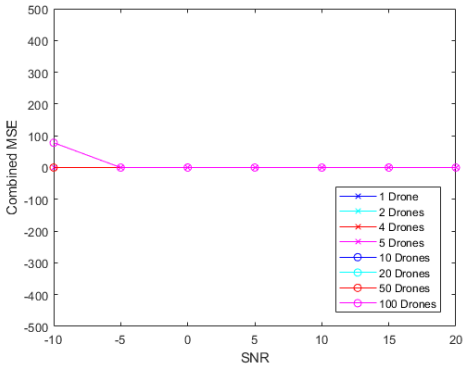
Figure 3.18. Combined MSE for different D drones. In this case, the set of R selected is the worst and the drones are located in the same position in a large vector.

3.3.7 Simulations with $\alpha = 0$ and $\alpha = 0.5$

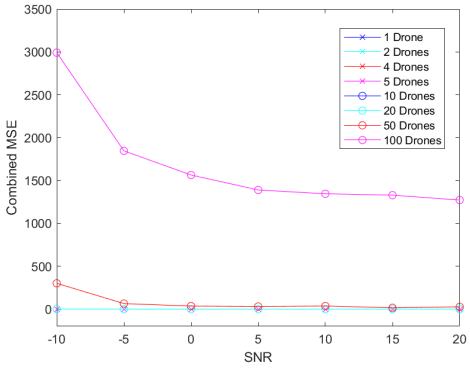
This section provide results from simulations that were performed for $\alpha = 0$ and $\alpha = 0.5$, intending to reproduce an environment without Doppler shift and with the maximum Doppler shift treated in this work, respectively.

- $\alpha = 0$

  From Figures 3.19, 3.20, and 3.21, we see that the MSE results for all ZC sequences containing a Doppler shift $\alpha = 0$ have an outstanding performance of near-zero errors for almost any scenario simulated. These results shows that the Doppler shift is the major source of errors in the scenarios.

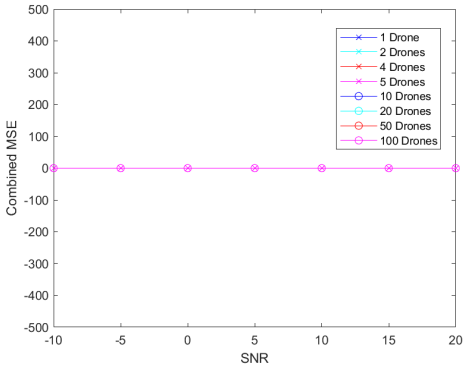


(a) Drones in Different ZC_{pos}

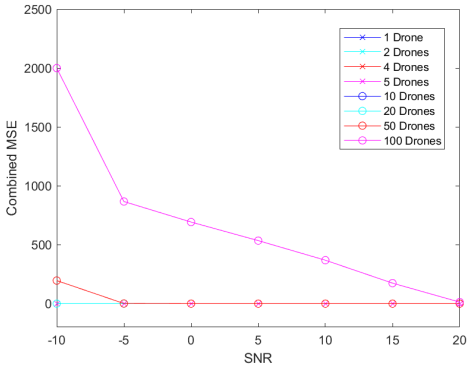


(b) Drones in the same ZC_{pos}

Figure 3.19. Linear combined MSE for random R and $\alpha = 0$.

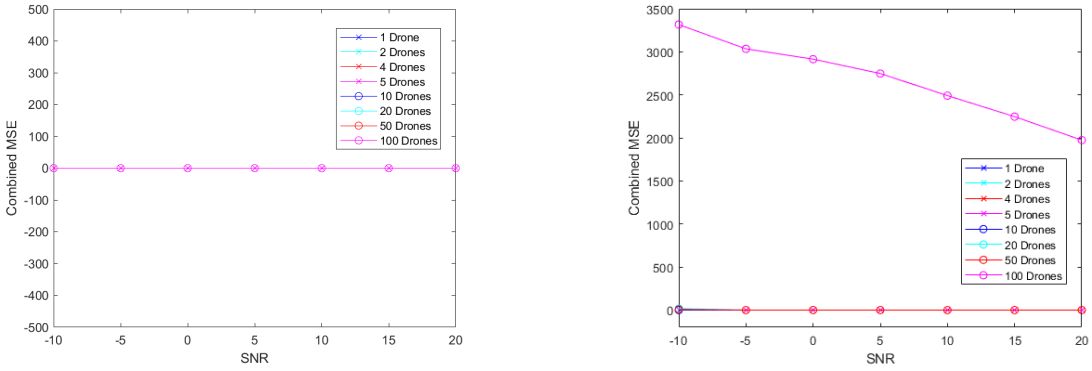


(a) Drones in Different ZC_{pos}



(b) Drones in the same ZC_{pos}

Figure 3.20. Linear combined MSE for best set of Rs and $\alpha = 0$.

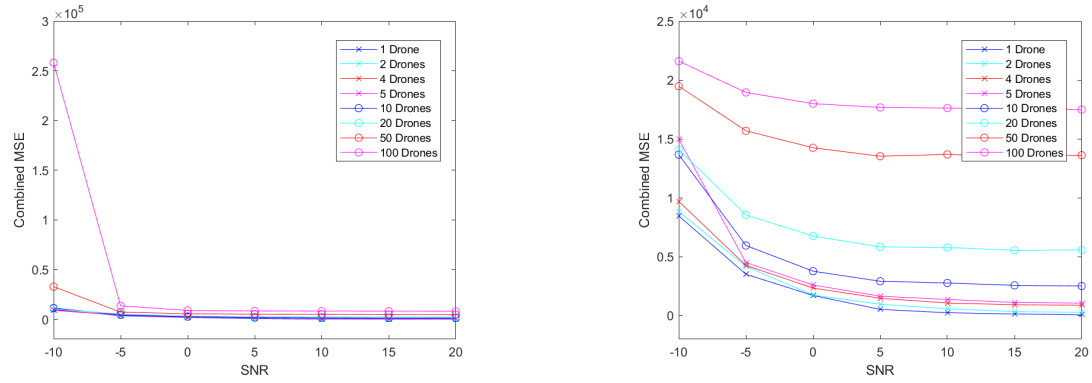


(a) Drones in Different ZC_{pos} (b) Drones in the same ZC_{pos}

Figure 3.21. Linear combined MSE for worst set of Rs and $\alpha = 0$.

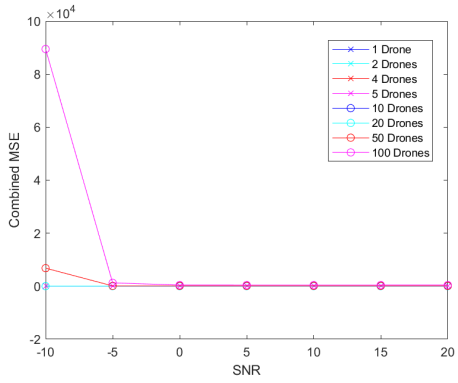
- $\alpha = 0.5$

  From Figures 3.22, 3.23, and 3.24, we see that the MSE results for all ZC sequences containing a Doppler shift $\alpha = 0.5$ have a slightly worse performance, especially for poor SNR and large swarms. These results are expected since the Doppler shift makes a large negative contribution to the errors.

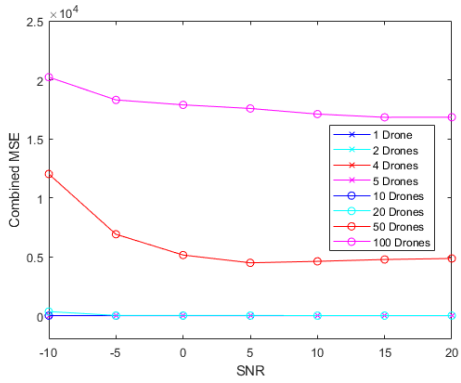


(a) Drones in Different ZC_{pos} (b) Drones in the same ZC_{pos}

Figure 3.22. Linear combined MSE for random R and $\alpha = 0.5$.

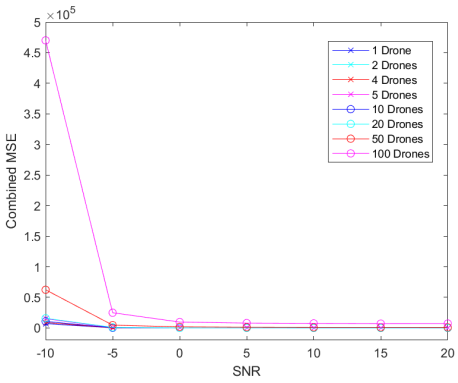


(a) Drones in Different ZC_{pos}

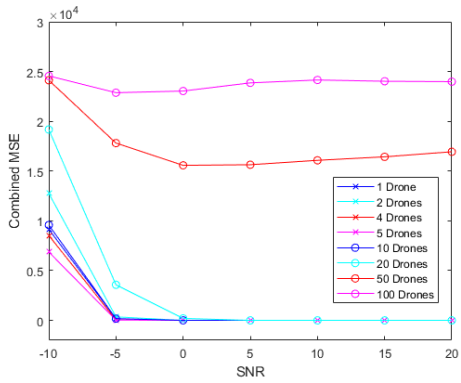


(b) Drones in the same ZC_{pos}

Figure 3.23. Linear combined MSE for best set of Rs and $\alpha = 0.5$.



(a) Drones in Different ZC_{pos}



(b) Drones in the same ZC_{pos}

Figure 3.24. Linear combined MSE for worst set of Rs and $\alpha = 0.5$

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Conclusion

This chapter provides a summary of the work done in this thesis and proposes areas for future work.

4.1 Summary

This work investigated how some variables, such as the number of drones, the structure of the packet preamble, the distribution of the drones in a swarm, or even the degree of environmental noise, affect the ability of UAVs to communicate in a swarm.

By using the MSE as a quality measure, we verified that, for a Doppler shift with a random α, the number of errors in communication increases in direct proportion to the increase in the number of drones in the swarm, and the number of errors increases when the scenario is embedded in low SNR. This is due to the confusion that is sometimes created when the matched filter is applied, generating an incorrectly measured position.

It is also possible to observe that when the drones are in the same position, obviously in a hypothetical scenario, the number of errors increases by a factor of ten for any R, be it random, worst, or best. This issue most likely occurs due to the noise behavior given by the repetitive addition of noisy signals, hence scaling the errors.

One more verification we accomplished was that both the set of best Rs and the set of worst Rs show better results than the set of random Rs. This can be explained by the reduction in the amplitude of side-peak occurrences generated for the worst set. The larger the difference between the amplitudes of the main and side peaks, the smaller the probability of incorrect detection, especially in good SNR environments.

Overall, the prime source of errors identified in this research was the noise in the environment, especially for high density swarms.

These results improve our knowledge about the factors that influence communication in a swarm of UAVs and provide useful data for future studies on methods for mitigating errors

in similar scenarios.

4.2 Further Research

This work shows that the number of errors increases as the number of drones in a swarm increases or when drones are placed hypothetically at the same distance from the reference drone. One question that might be researched is: How can these factors be eliminated?

The answer to that question may be present in machine learning techniques, which can provide ways to mitigate these errors. The input of a neural network, for instance, would include the random data of position and the desired output, and the correct position of the drone, which could mitigate the problems associated with the SNR and the number of drones.

Another aspect not verified in this work is the sub-sampling effect, which can cause a power division of the peak identified in the matched filter. This approach could be considered in future work, since this scenario is certainly plausible.

As explained in Section 2.7.1, real applications using the method proposed in this thesis would have issues if using a 5G network due to its limited symbol rate. Another proposal of future work would be to address this limitation, going deeper into 5G technology and looking for some alternative with a faster symbol rate.

The last, but not least, desirable application for this study would be a real implementation with small UAVs provided with 5G communication, flying together in small swarms.

APPENDIX:
MATLAB Code

A.1 MSECombBestRDiffPosFdRand.m Scenario Code

This code is an example of the simulations performed in this work, generating a plot of combined MSE for the selection of best *R*s, located in different ZC_{pos} for a random Doppler effect.

```
% Thesis work
% Professor: John Roth
% Student: LT Madjer Martins
%-----------------------------------------%
% Multi drone simulation by changing a R for D drones
% Creating each N samples ZC sequence separately and then
    applying doppler
% separately also. Adding all clean N samples sequences and
    then applying
% the noise (complex).
% Combined MSE
% Executing the algorithm for the D R's in 1000 (trials) and
     get the
% results of MSE for each SNR (-10 -5 0 5 10 15 20)
% Here I'm choosing the D best Rs
%-----------------------------------------%

clear all;
close all;
tStart = tic
%-----------------------------------------%
% Parameters inicializations
```

```
20 Drones = [1 2 4 5 10 20 50 100]; % drone# d in the space d =
       [1,D] D = # of drones
21 %u = 1;
22 % Nth root. R and Nzc must be prionic (only common divisor
       is 1)
23 % Nzc = length of sequence
24 Nzc = 839; % length of sequence
25 colors = ['r' 'b' 'y' 'g' 'c' 'm'];
26 % Creating R for D drones
27 % Nth root. R and Nzc must be prionic (only common divisor
       is 1)
28 R_primes = 1:Nzc-1; % I exclude 839
29 % R = randperm(length(R_primes),D); % Provide R without
       repetitions
30
31 N = 10000; % the entire data will have 10000 samples
32 trials = 1000;
33 SNR = [-10 -5 0 5 10 15 20];
34
35 for Dk = 1:length(Drones)
36     tic
37     D = Drones(Dk);
38
39     % [G,U,V] = gcd(A,B) returns A.*U + B.*V = G.
40     %[~,Rinv] = gcd(R,Nzc(4))
41     % U = Rinv, U gives the number of points between the
           position of begining
42     % of ZC seq without doppler effect and the position of
           begining of ZC seq
43     % with doppler effect.
44 %     [G,U,V] = gcd(R_primes,Nzc);
45 %     mod_U = mod(U,Nzc);
46 %     [B,I] = sort(mod_U,'descend');
```

```
47 %       I = I(1:D);
48 %       R = R_primes(I);
49
50       % Best case of Rs from the vector Sorted_R_Best_to_Worst
             .mat
51       R_temp = load('Sorted_R_best_to_Worst.mat');
52       R = R_temp.R_sorted(1:D);
53
54       f_doppler = 0:0.05:0.5;%0:0.1:1; % % 11 elements
55
56       % Reseting some variables
57       a_zc_doppler = [];
58       MSE_combined = zeros(1,length(SNR));
59       signal = [];
60       doppler = [];
61       ZC_positions = [];
62
63       %----------------------------------------%
64       % k = trials, N = # of samples, kk = SNRs
65       % Data structure
66       % signal(k,kk)
67       % For-loop SNRs
68       % signal = struct with fields:
69       %   SNR:
70       %   -trial:
71       %   ZCSeq_pos: vector with dimension D (# of drones)
72       %   doppler:
73       %   data: (samples with both drones zc sequence added
             with noise and applied
74       %   doppler)
75       %   -Rx(d):
76       %   -lags(d):
77       %   -max_val(d):
```

```
78      %    -I_max(d):
79      %    error:
80
81      %-----------------------------------------%
82      % For-loop SNRs
83      % parpool(length(SNR))
84      for k = 1:length(SNR)
85
86          error_sum = zeros(1,D);
87
88          %-----------------------------------------%
89          % For-loop 1000 trials
90          for kk = 1:trials
91
92              %------ Creating the ZC sequences separately
93              a_zc_N = zeros(D,N);
94              % Finding a vector with D positions of ZC
                    sequences
95              ZCSeq_pos = randi(N-Nzc,1,D);
96              signal(k,kk).ZCSeq_pos = ZCSeq_pos;
97              % creating the zc sequences separetely
98              for d=1:D
99                  a_zc(d,1:Nzc) = zadoffChuSeq(R(d),Nzc);
100             end
101
102             % Putting the zc sequence at ZCSeq_pos
103             %for m = 1:Nzc
104             for d = 1:D
105                 %a_zc_N(d,ZCSeq_pos(d)+m) = a_zc_N(d,
                        ZCSeq_pos(d)+m) + a_zc(d,m);
106                 a_zc_N(d,(ZCSeq_pos(d):ZCSeq_pos(d)+Nzc-1))
                        = a_zc(d,1:Nzc);
107             end
```

```
108             %end
109             %-----------------------------------------%
110             %           figure
111             %           plot(abs(a_zc_N(1,:)))
112
113             %------ Applying doppler to each pure zc
                    sequence and then, add
114             % them
115             l = [0:N-1];
116             a_zc_noisy = zeros(1,N);
117             for d = 1:D
118                 % choosing a random doppler effect using
                        randi() for each drone
119                 signal(k,kk).doppler(d) = f_doppler(randi(
                        length(f_doppler)));
120                 a_zc_doppler(d,:) = a_zc_N(d,:).*exp(1i*2*pi
                        .*(signal(k,kk).doppler(d)/Nzc).*l);
121                 a_zc_noisy = a_zc_noisy + a_zc_doppler(d,:);
                        % preparing to apply noise
122             end
123
124             %-----------------------------------------%
125             %           figure
126             %            plot(abs(a_zc_noisy))
127
128             %------ Creating the complex noisy signal for N
                    samples Zadoff-chu
129             % sequence
130             signal(k,kk).SNR = SNR(k);
131
132             sigPwr = 1; %rms(a_zc_noisy(1,:))^2; %Show that
                    the power of our signal is 1
```

```
133         linearSNR = 10^(SNR(k)/10); %Convert decibel-
                valued SNR to a regular ratio value
134         nsPwr = sigPwr/linearSNR; %Find required complex
                 noise power
135         realNsPwr = nsPwr/2; %remember that this is the
                same value as the complex noise power
136         sig = sqrt(realNsPwr); %std deviation of the
                real and imaginary noise components
137         correctNoise = sig*randn(1,length(a_zc_noisy
                (1,:))) ...
138              + 1i*sig*randn(1,length(a_zc_noisy(1,:)));
139
140         a_zc_noisy(1,:) = a_zc_noisy(1,:) + correctNoise
                ;
141         %----------------------------------------%
142         %           figure
143         %           plot(abs(a_zc_noisy))
144
145         %****************************************%
146         % Matched filter and errors
147         signal(k,kk).data = a_zc_noisy;
148         for d = 1:D
149             [Rx(d,:), lags(d,:)] = xcorr(a_zc_noisy,a_zc
                    (d,:));
150             %signal(k,kk).Rx(d,:) = Rx(d,:);
151             %signal(k,kk).lags(d,:) = lags(d,:);
152             %plot(lags(d,:),abs(Rx(d,:)))
153
154             % Finding the greatest Rx (Not using Rinv)
155             [max_val(d) I_max(d)] = max(abs(Rx(d,:)));
156             %signal(k,kk).max_val(d) = max_val(d);
157             %signal(k,kk).I_max(d) = I_max(d);
158
```

```
159                 % I subtract N and add 1 because the xcorr
                        double the number of samples and
160                 % center in 0 (# of samples = 2N + 1)
161                 signal(k,kk).error(d) = ((I_max(d) - N + 1)
                        - ZCSeq_pos(d))^2;
162                 error_sum(d) = error_sum(d) + signal(k,kk).
                        error(d);
163
164                 %--------------------------------%
165                 % Preparing data to save:
166                 doppler(k,kk,:) = squeeze(signal(k,kk).
                        doppler(:));
167                 ZC_positions(k,kk,:) = ZCSeq_pos;
168                 %--------------------------------%
169
170             end
171             %*****************************************%
172
173         end % For-loop trials
174
175         %-----------------------------------------%
176
177         % Calculating the mean square error for each SNR k
178         for d = 1:D
179             MSE_combined(k) = MSE_combined(k) + error_sum(d)
                    ;
180         end
181
182     end % For-loop SNRs
183
184     %-----------------------------------------%
185
186     MSE = MSE_combined/(trials*D);
```

```
187
188     figure
189     plot(abs(a_zc_noisy))
190
191     str_saveas = 'Noisy_BestR_%d_drones.png';
192     str_saveas = sprintf(str_saveas, D);
193     saveas(gcf,str_saveas,'png')
194
195     figure
196     plot(SNR,MSE)
197     title(['MSE combined for ' num2str(D) ' drones and f_{
            doppler} = 0.5*U[0,1]*f_0' ])
198     xlabel('SNR')
199     ylabel('Mean Square Error')
200     %legend('Drone 1 R = 17','Drone 2 R = 23','Drone 3 R =
            53','Location','northeast')
201     set(gcf,'name','MSE for a noisy signal under doppler
            effect with D drones','numbertitle','off')
202
203     % I didn't apply the subsamplying division of power
204     % I'm just getting maximums
205     elapsedTime = toc
206
207     str_saveas = 'MSE_BestR_%d_drones.png';
208     str_saveas = sprintf(str_saveas, D);
209     saveas(gcf,str_saveas,'png')
210
211     % Saving the workspace
212     str_wks_save = 'MSE_BestR_%d_drones.mat';
213     str_wks_save = sprintf(str_wks_save, D);
214     save(str_wks_save,'R','ZC_positions','doppler','SNR','
            MSE','elapsedTime')
215
```

```
end % Different # of Drones
%-----------------------------------------%

%-----------------------------------------%
% Script to read every MSE for each drone and plot it
    together
str_load = 'MSE_BestR_%d_drones.mat';
% str_title = 'Combined MSE for different number of drones -
    Best R, Random ZC positions';
str_title = ' ';
str_saveas = 'MSE_Combined_BestR_DiffPos_FdRand.png';
Read_All_MSE(SNR,Drones,str_load,str_title,str_saveas)

tEnd = toc(tStart)
```

A.2 ReadAllMSE.m Code

This code is a function that obtains the results for each D swarm, generated in scenario Code, and combines them together in just one plot.

```
% Thesis work
% Professor: John Roth
% Student: LT Madjer Martins
%-----------------------------------------%
% function to read every MSE for each drone and plot it
    together
%-----------------------------------------%
function save_OK = Read_all_MSE(SNR,d,str_load,str_title,
    str_saveas)

%SNR = [-10 -5 0 5 10 15 20];
%d = [1 2 4 5 10 20 50 100];
%str_load = 'MSE_comb_%d_drones.mat';
num_files = length(d);
```

```
color = ['-xb'; '-xc'; '-xr'; '-xm'; '-ob'; '-oc'; '-or'; '-
    om'];
figure
for k = 1:num_files
    str_d_load = sprintf(str_load, d(k));
    data(k) = load(str_d_load);
    semilogy(SNR, data(k).MSE, color(k,:))
    hold on
end
legend('1 Drone','2 Drones', '4 Drones', '5 Drones', '10
    Drones', ...
    '20 Drones', '50 Drones', '100 Drones')
xlabel('SNR')
ylabel('Combined MSE')
%str_title = 'Combined MSE for different number of drones -
    Random R, Random ZC positions';
title(str_title)

%str_saveas = 'MSE_Combined_All_RandomR_DiffPositions.png';
saveas(gcf,str_saveas,'png')

save_OK = 1;

end
```

A.3 BestWorstR.m Code

This code provides support for understanding the sorting procedure for the *R* following the creation of a vector with sorted *R*.

```
clear all;
close all;

D = 1;
```

```
SNR = 100; % almost No noise
Nzc = 839;
N = 10000;
f_doppler = 0.5; % Worst case of doppler
ZCSeq_pos = randi(N-Nzc,1,D); % D random positions
a_zc_noisy = [];
Rxx = [];
lags = [];
for R = 1:838
    a_zc_noisy = Createa_a_zc_noisy_doppler(D,Nzc,N,
       ZCSeq_pos,SNR,f_doppler,R);
    a_zc = zadoffChuSeq(R,Nzc);
    [Rxx(R,:) , lags(R,:)] = xcorr(a_zc_noisy,a_zc);
    [y,x] = sort(abs(Rxx(R,:)),'descend');
    dist_2peaks(R) = abs(x(1) - x(2)); % Distance difference
    diff_2peaks(R) = abs(y(1) - y(2)); % Amplitude
       difference
    vector_2peaks(R) = sqrt((dist_2peaks(R)^2)+(diff_2peaks(
       R)^2));
    dist_correct_peak(R) = abs(x(1)-ZCSeq_pos-N+1);
end

f_doppler = 0.25;
for R = 1:838
    a_zc_noisy = Createa_a_zc_noisy_doppler(D,Nzc,N,
       ZCSeq_pos,SNR,f_doppler,R);
    a_zc = zadoffChuSeq(R,Nzc);
    [Rxx(R,:) , lags(R,:)] = xcorr(a_zc_noisy,a_zc);
    [y,x] = sort(abs(Rxx(R,:)),'descend');
    dist_2peaks_025(R) = abs(x(1) - x(2));
    diff_2peaks_025(R) = abs(y(1) - y(2));
    vector_2peaks_025(R) = sqrt((dist_2peaks_025(R)^2)+(
       diff_2peaks_025(R)^2));
```

```
33 end
34
35 f_doppler = 0.05;
36 for R = 1:838
37     a_zc_noisy = Createa_a_zc_noisy_doppler(D,Nzc,N,
          ZCSeq_pos,SNR,f_doppler,R);
38     a_zc = zadoffChuSeq(R,Nzc);
39     [Rxx(R,:) , lags(R,:)] = xcorr(a_zc_noisy,a_zc);
40     [y,x] = sort(abs(Rxx(R,:)),'descend');
41     dist_2peaks_005(R) = abs(x(1) - x(2));
42     diff_2peaks_005(R) = abs(y(1) - y(2));
43     vector_2peaks_005(R) = sqrt((dist_2peaks_005(R)^2)+(
          diff_2peaks_005(R)^2));
44 end
45
46 figure
47 subplot(3,1,1)
48 plot(dist_2peaks)
49 title('error between the distance of 2 peaks for each R,
     f_doppler = 0.5')
50 subplot(3,1,2)
51 plot(dist_2peaks_025)
52 title('error between the distance of 2 peaks for each R,
     f_doppler = 0.25')
53 subplot(3,1,3)
54 plot(dist_2peaks_005)
55 title('error between the distance of 2 peaks for each R,
     f_doppler = 0.05')
56
57 figure
58 subplot(3,1,1)
59 plot(diff_2peaks)
```

```
60 title('error between the amplitude of 2 peaks for each R,
       f_doppler = 0.5')
61 subplot(3,1,2)
62 plot(diff_2peaks_025)
63 title('error between the amplitude of 2 peaks for each R,
       f_doppler = 0.25')
64 subplot(3,1,3)
65 plot(diff_2peaks_005)
66 title('error between the amplitude of 2 peaks for each R,
       f_doppler = 0.05')
67
68 figure
69 subplot(3,1,1)
70 plot(vector_2peaks)
71 title('vector error between 2 peaks for each R, f_doppler =
       0.5')
72 subplot(3,1,2)
73 plot(vector_2peaks_025)
74 title('vector error between 2 peaks for each R, f_doppler =
       0.25')
75 subplot(3,1,3)
76 plot(vector_2peaks_005)
77 title('vector error between 2 peaks for each R, f_doppler =
       0.05')
78
79 [dist_2peaks_sorted,R_dist_sorted] = sort(dist_2peaks,'
       ascend');
80 [dist_2peaks_sorted_025,R_dist_sorted_025] = sort(
       dist_2peaks_025,'ascend');
81 [dist_2peaks_sorted_005,R_dist_sorted_005] = sort(
       dist_2peaks_005,'ascend');
82
83 R_dist_sorted(1:10)
```

```
84 R_dist_sorted_025(1:10)
85 R_dist_sorted_005(1:10)
86
87 R_dist_sorted - R_dist_sorted_025
88 R_dist_sorted - R_dist_sorted_005
89
90 figure
91 subplot(3,1,1)
92 plot(dist_2peaks_sorted)
93 subplot(3,1,2)
94 plot(dist_2peaks_sorted_025)
95 subplot(3,1,3)
96 plot(dist_2peaks_sorted_005)
97 title('Distance the 2 largest peaks sorted')
98
99 figure
100 plot(dist_correct_peak)
101 title('error for maximum peak for R_{xy} and correct ZC_{pos
    }')
102
103 % Saving the workspace
104 str_wks_save = 'Sorted_R_Best_to_Worst.mat';
105 save(str_wks_save,'R_dist_sorted');
106
107
108 %-----Function that create s[n]------
109 function a_zc_noisy_doppler = Createa_a_zc_noisy_doppler(D,
    Nzc,N,ZCSeq_pos,SNR,f_doppler,R)
110
111 if R == 0
112     R_primes = 1:Nzc-1; % I exclude 839
113     R = randperm(length(R_primes),D); % Provide R without
        repetitions
```

```
114 end
115
116 %------ Creating the ZC sequences separately
117 a_zc_N = zeros(D,N);
118
119 % creating the zc sequences separetely
120 for d=1:D
121     a_zc(d,1:Nzc) = zadoffChuSeq(R(d),Nzc);
122 end
123
124 % Putting the zc sequence at ZCSeq_pos
125 %for m = 1:Nzc
126 for d = 1:D
127     %a_zc_N(d,ZCSeq_pos(d)+m) = a_zc_N(d,ZCSeq_pos(d)+m) +
        a_zc(d,m);
128     a_zc_N(d,(ZCSeq_pos(d):ZCSeq_pos(d)+Nzc-1)) = a_zc(d,1:
        Nzc);
129 end
130 %end
131 %-----------------------------------------%
132 %         figure
133 %         plot(abs(a_zc_N(1,:)))
134
135 %------ Applying doppler to each pure zc sequence and then,
    add
136 % them
137 l = [0:N-1];
138 a_zc_noisy_doppler = zeros(1,N);
139 for d = 1:D
140     % choosing a random doppler effect using randi() for
        each drone
141     a_zc_doppler(d,:) = a_zc_N(d,:).*exp(1i*2*pi.*(f_doppler
        /Nzc).*l);
```

```
142     a_zc_noisy_doppler = a_zc_noisy_doppler + a_zc_doppler(d
          ,:); % preparing to apply noise
143 end
144
145 %-----------------------------------------%
146 %           figure
147 %            plot(abs(a_zc_noisy))
148
149 %------ Creating the complex noisy signal for N samples
      Zadoff-chu
150 % sequence
151
152 sigPwr = 1; %rms(a_zc_noisy(1,:))^2; %Show that the power of
       our signal is 1
153 linearSNR = 10^(SNR/10); %Convert decibel-valued SNR to a
      regular ratio value
154 nsPwr = sigPwr/linearSNR; %Find required complex noise power
155 realNsPwr = nsPwr/2; %remember that this is the same value
      as the complex noise power
156 sig = sqrt(realNsPwr); %std deviation of the real and
      imaginary noise components
157 correctNoise = sig*randn(1,length(a_zc_noisy_doppler)) ...
158     + 1i*sig*randn(1,length(a_zc_noisy_doppler));
159
160 a_zc_noisy_doppler = a_zc_noisy_doppler + correctNoise;
161
162 end
```

List of References

[1] K. P. Valavanis and G. J. Vachtsevanos, *Handbook of Unmanned Aerial Vehicles*. Dordrecht, Netherlands: Springer Netherlands, 2014.

[2] N. Jia, Z. Yang, T. Liao, Y. Dou, and K. Yang, "A system dynamics model for analyzing swarming UAVs air combat system," in *13th Annual Conference on System of Systems Engineering*, 2018, pp. 74–81.

[3] Q. Cui, P. Liu, J. Wang, and J. Yu, "Brief analysis of drone swarms communication," in *IEEE International Conference on Unmanned Systems*, 2017, pp. 463–466.

[4] L. Cui, H. Zhang, X. Liu, and T. A. Gulliver, "A ranging method for 60GHz OFDM system based on improved preamble sequence," in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 2015, pp. 246–251.

[5] V. Savaux and F. Bader, "Mean square error analysis and linear minimum mean square error application for preamble-based channel estimation in orthogonal frequency division multiplexing/offset quadrature amplitude modulation systems," *IET Communications*, vol. 9, no. 14, pp. 1763–1773, 2015.

[6] M. Hua, M. Wang, K. W. Yang, and K. J. Zou, "Analysis of the frequency offset effect on Zadoff-Chu sequence timing performance," *IEEE Transactions on Communications*, vol. 62, no. 11, pp. 4024–4039, 2014.

[7] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The Next Generation Wireless Access Technology*, 1st ed. London, England: Academic Press, 2018.

[8] K. T. W. Page. "Concepts of Orthogonal Frequency Division Multiplexing (OFDM) and 802.11 WLAN." Accessed March 31, 2021. [Online]. Available: http://rfmw.em.keysight.com/wireless/helpfiles/89600B/WebHelp/Subsystems/wlan-ofdm/Content/ofdm_basicprinciplesoverview.htm

[9] ETSI. "3GPP TS 38.211 v15.7.0, Technical Specification, 3rd Generation Partnership Project". [Online]. Available: https://www.etsi.org/deliver/etsi_ts/138200_138299/138211/15.07.00_60/ts_138211v150700p.pdf

[10] I. Mercer, "Merit factor of Chu sequences and best merit factor of polyphase sequences," *IEEE Transactions on Information Theory*, vol. 59, no. 9, pp. 6083–6086, 2013.

[11] D. Chu, "Polyphase codes with good periodic correlation properties (corresp.)," *IEEE Transactions on Information Theory*, vol. 18, no. 4, pp. 531–532, 1972.

[12] MathWorks, Inc., 2020. "MATLAB, version 9.9.0 (R2020b).". [Online]. Available: https://www.mathworks.com/help/comm/ref/zadoffchuseq.html

[13] R. Frank, S. Zadoff, and R. Heimiller, "Phase shift pulse codes with good periodic correlation properties (corresp.)," *I.R.E. Transactions on Information Theory*, vol. 8, no. 6, pp. 381–382, 1962.

[14] C. Carlet and A. Pott (Eds.), *Sequences and Their Applications - SETA 2010 6th International Conference, Paris, France, September 13-17, 2010. Proceedings*, 1st ed. (Theoretical Computer Science and General Issues; 6338). Berlin, Heidelberg, Germany: Springer Berlin Heidelberg, 2010.

[15] C.-D. Chung, W.-C. Chen, and C.-K. Yang, "Constant-amplitude sequences for spectrally compact OFDM training waveforms," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 974–12 991, 2020.

[16] R. Cristi, *Modern Digital Signal Processing*. Pacific Grove, CA, USA: Thomson/Brooks/Cole, 2004.

[17] C. W. Therrien and M. Tummala, *Probability and Random Processes for Electrical and Computer Engineers*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2012.

[18] D. Manolakis, V. Ingle, and S. Kogon, *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering and Array Processing*. Norwood, MA, USA: Artech House, 2005.

[19] B. Popovic, "Generalized chirp-like polyphase sequences with optimum correlation properties," *IEEE Transactions on Information Theory*, vol. 38, no. 4, pp. 1406–1409, 1992.

[20] A. B. D. Paula. "An introduction to Doppler effect and fading in mobile communication," M.S. thesis, Dept. of Electrical Engineering, Naval Postgraduate School, Monterey, CA, USA, December 1992. [Online]. Available: https://apps.dtic.mil/sti/pdfs/ADA260976.pdf

[21] S. Haykin and M. Moher, *Introduction to Analog and Digital Communications*, 2nd ed. New York, NY, USA: Wiley, 2007.

[22] R. R. D. Paula. "Monte Carlo Method and Applications," B.S. monography, Mathematics Institute, Universidade Federal Fluminense, Volta Redonda, RJ, Brazil, November 2014. [Online]. Available: https://app.uff.br/riuff/bitstream/1/4180/1/RenatoRicardoDePaula%202014-2.PDF

[23] C. Z. Mooney, *Monte Carlo Simulation*. Thousand Oaks, CA, USA: SAGE, 1997.

[24] 5G Networks. "Spectral Efficiency: 5G-NR and 4G-LTE compared." February 8, 2021. [Online]. Available: https://www.5g-networks.net/5g-technology/spectral-efficiency-5g-nr-and-4g-lte-compared/

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California